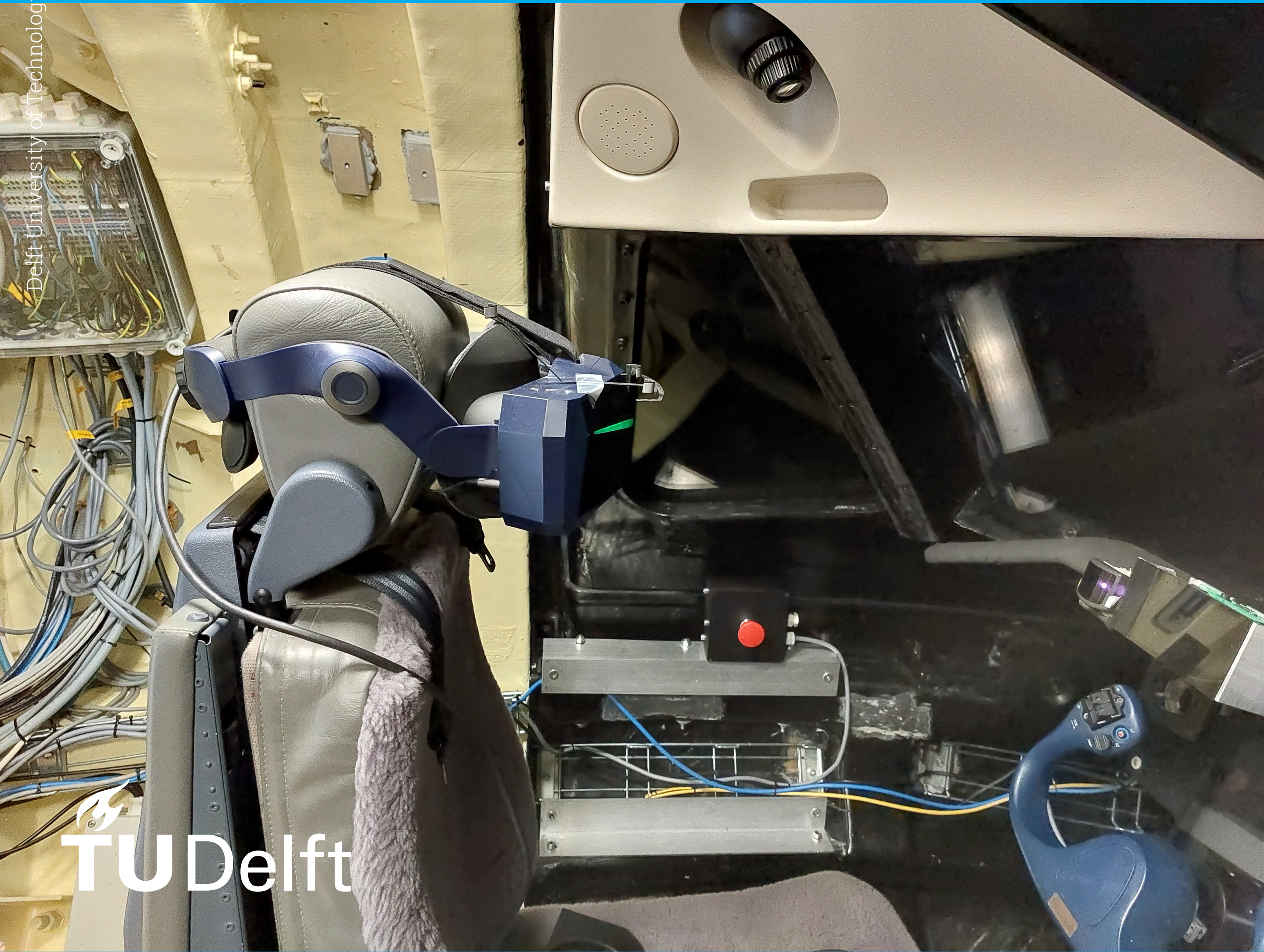
Enabling a Head-Mounted Display in an Enclosed Cabin on a Moving Base Simulator

Multiple Implementations with an Unscented Kalman Filter

Jorne Matton

Delft University of Technology



Enabling a Head-Mounted Display in an Enclosed Cabin on a Moving Base Simulator

Multiple Implementations with an Unscented Kalman Filter

by

Jorne Matton

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday December 12, 2023 at 9:00 AM.

Student number:	4215885	
Project duration:	June 15, 2020 — December 12, 2023	
Thesis committee:	Dr. Ir. M. M. van Paassen,	TU Delft, chair
	Ir. O. Stroosma,	TU Delft, supervisor
	Dr. S. J. Hulshoff,	TU Delft
	Dr. Ir. D. M. Pool,	TU Delft

Cover: SIMONA Research Simulator [62]
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Contents

List of Figures

List of Tables

List of Abbreviations

A

AD Allan Deviation.
AR Augmented Reality.
AV Allan Variation.

C

CF Complementary Filter.

D

DERP Design Eye Reference Point.
DoF Degree-of-Freedom.
DUECA Delft University Environment for Communication and Activation.

E

eCAL enhanced Communication Abstraction Layer.
EKF Extended Kalman Filter.

F

FoR Field-of-Regard.

G

GN Gauss-Newton.

H

HMD Head-Mounted Display.

I

i.e. id est.
IEKF Iterative Extended Kalman Filter.
IMU Inertial Measurement Unit.
IR Infrared.

K

KF Kalman Filter.

L

LGP Lower Gimbal Point.

M

MEMS Micro-Electro-Mechanical Systems.
MMSE Minimum Mean-Square-Error.
MQTT Message Queuing Telemetry Transport.
MSE Mean Squared Error.

N

NEES Normalized Estimated Error Squared.
NR Newton-Raphson.

O

OAD Overlapping Allan Deviation.

P

PC Personal Computer.
PSD Power Spectral Density.

R

ROS Robot Operating System.

S

SD Standard Deviation.
SDSF Smoothed Delayed Sensor Fusion.
SDUKF Smoothed Delayed Unscented Kalman Filter.
SLAM Simultanious Localization And Mapping.
SRS SIMONA Research Simulator.
SUT Scaled Unscented Transformation.

U

UDP User Datagram Protocol.
UGP Upper Gimbal Point.
UKF Unscented Kalman Filter.
USB Universal Standard Bus.

V

VR Virtual Reality.

W

w.r.t. with respect to.

List of Symbols

Scalars

L Dimension of the total augmented state.
L^o Dimension of the observation noise.
L^v Dimension of the process noise.
L^x Dimension of the state.
$\ddot{x}$ Acceleration x-axis component in $\mathbb{E}_I$.
$\ddot{y}$ Acceleration y-axis component in $\mathbb{E}_I$.
$\ddot{z}$ Acceleration z-axis component in $\mathbb{E}_I$.
α Positive sigma-point weighting term and controls the distance between the sigma-points and the current state.
A Amplitude.
β Non-negative sigma-point weighting term and controls the influence of the higher order moments of the probability distribution.
λ_p MEMS angular rate bias x-axis component in $\mathbb{E}_{\square}$.
λ_q MEMS angular rate bias y-axis component in $\mathbb{E}_{\square}$.

λ_r MEMS angular rate bias z-axis component in $\mathbb{E}_{\square}$.

$\lambda_{\ddot{x}}$ MEMS acceleration bias x-axis component in $\mathbb{E}_{\square}$.

$\lambda_{\ddot{y}}$ MEMS acceleration bias y-axis component in $\mathbb{E}_{\square}$.

$\lambda_{\ddot{z}}$ MEMS acceleration bias z-axis component in $\mathbb{E}_{\square}$.

b_{x_i} Each of the Stewart platform joint x-coordinate in $\mathbb{E}_B$.

b_{y_i} Each of the Stewart platform joint y-coordinate in $\mathbb{E}_B$.

b_{z_i} Each of the Stewart platform joint z-coordinate in $\mathbb{E}_B$.

μ_p MEMS angular rate bias stability x-axis component in $\mathbb{E}_{\square}$.

μ_q MEMS angular rate bias stability y-axis component in $\mathbb{E}_{\square}$.

μ_r MEMS angular rate bias stability z-axis component in $\mathbb{E}_{\square}$.

$\mu_{\ddot{x}}$ MEMS acceleration bias stability x-axis component in $\mathbb{E}_{\square}$.

$\mu_{\ddot{y}}$ MEMS acceleration bias stability y-axis component in $\mathbb{E}_{\square}$.

$\mu_{\ddot{z}}$ MEMS acceleration bias stability z-axis component in $\mathbb{E}_{\square}$.

d_B Distance between two joints of a pair on the bottom fixed frame of the Stewart platform.

$\dot{\tau}_{\dot{p}}$ Lag of the angular acceleration around the x-axis.

$\dot{\tau}_{\dot{q}}$ Lag of the anglular acceleration around the y-axis.

$\dot{\tau}_{\dot{r}}$ Lag of the angular acceleration around the z-axis.

$\dot{\tau}_{\ddot{x}}$ Lag of the acceleration on the x-axis.

$\dot{\tau}_{\ddot{y}}$ Lag of the acceleration on the y-axis.

$\dot{\tau}_{\ddot{z}}$ Lag of the acceleration on the z-axis.

d_P Distance between two joints of a pair on the upper moving frame of the Stewart platform.

e_0 Quaternion scalar component.

e_x Quaternion x-axis vector component in $\mathbb{E}_I$.

e_y Quaternion y-axis vector component in $\mathbb{E}_I$.

e_z Quaternion z-axis vector component in $\mathbb{E}_I$.

$f_{\ddot{x}}$ Specific force x-axis component in $\mathbb{E}_{\square}$.

$f_{\ddot{y}}$ Specific force y-axis component in $\mathbb{E}_{\square}$.

$f_{\ddot{z}}$ Specific force z-axis component in $\mathbb{E}_{\square}$.

g Gravitational constant = $9.81\,\mathrm{m/s^2}$.

$\dddot{x}$ Jerk x-axis component in $\mathbb{E}_I$.

$\dddot{y}$ Jerk y-axis component in $\mathbb{E}_I$.

$\dddot{z}$ Jerk z-axis component in $\mathbb{E}_I$.

κ Non-negative sigma-point weighting term and helps to guarantee positive semi-definiteness of the covariance matrix.

ζ Intermediate for sigma-point weights.

l_i ith hydraulic actuators length.

s Norm of a angle-axis rotation.

p Angluar rate $\underline{\omega}$ x-axis component in $\mathbb{E}_{\square}$.

$\ddot{p}$ Angluar jerk $\underline{\ddot{\omega}}$ x-axis component.

$\dot{p}$ Angluar acceleration $\underline{\dot{\omega}}$ x-axis component.

θ y-axis rotation in Euler attitude formulation.

p_{x_i} Each of the Stewart platform joint x-coordinate in $\mathbb{E}_P$.

p_{y_i} Each of the Stewart platform joint y-coordinate in $\mathbb{E}_P$.

p_{z_i} Each of the Stewart platform joint z-coordinate in $\mathbb{E}_P$.

x Position x-axis component in $\mathbb{E}_I$.

y Position y-axis component in $\mathbb{E}_I$.

z Position z-axis component in $\mathbb{E}_I$.

q Angluar rate $\underline{\omega}$ y-axis component in $\mathbb{E}_{\square}$.

$\ddot{q}$ Angluar jerk $\underline{\ddot{\omega}}$ y-axis component.

$\dot{q}$ Angluar acceleration $\underline{\dot{\omega}}$ y-axis component.

r Angluar rate $\underline{\omega}$ z-axis component in $\mathbb{E}_{\square}$.

r_B Constant radius from the LGP to the joints on the lower fixed frame of the Stewart platform.

$\ddot{r}$ Angluar jerk $\underline{\ddot{\omega}}$ z-axis component.

$\dot{r}$ Angluar acceleration $\underline{\dot{\omega}}$ z-axis component.

ϕ x-axis rotation in Euler attitude formulation.

r_P Constant radius from the UGP to the joints on the upper moving frame of the Stewart platform.

T Scaled duration.

$s_{\dot{p}}$ Second derivative of attitude setpoint $\underline{\boldsymbol{s}}_{\ddot{c}}$ x-axis component in upper moving frame $\mathbb{E}_P$.

$s_{\dot{q}}$ Second derivative of attitude setpoint $\underline{\boldsymbol{s}}_{\ddot{c}}$ y-axis component in upper moving frame $\mathbb{E}_P$.

$s_{\dot{r}}$ Second derivative of attitude setpoint $\underline{\boldsymbol{s}}_{\ddot{c}}$ z-axis component in upper moving frame $\mathbb{E}_P$.

$s_{\ddot{x}}$ Second derivative of positon setpoint $\underline{\boldsymbol{s}}_{\ddot{c}}$ x-axis component in inertial frame $\mathbb{E}_I$.

$s_{\ddot{y}}$ Second derivative of positon setpoint $\underline{\boldsymbol{s}}_{\ddot{c}}$ y-axis component in inertial frame $\mathbb{E}_I$.

$s_{\ddot{z}}$ Second derivative of positon setpoint $\underline{\boldsymbol{s}}_{\ddot{c}}$ z-axis component in inertial frame $\mathbb{E}_I$.

σ Standard deviation.

τ Lag of a system to reach 63% of the final step input value.

$\tau_{\dot{p}}$ Lag of the angular acceleration around the x-axis.

$\tau_{\dot{q}}$ Lag of the anglular acceleration around the y-axis.

$\tau_{\dot{r}}$ Lag of the angular acceleration around the z-axis.

$\tau_{\ddot{x}}$ Lag of the acceleration on the x-axis.

$\tau_{\ddot{y}}$ Lag of the acceleration on the y-axis.

$\tau_{\ddot{z}}$ Lag of the acceleration on the z-axis.

θ_B Angle between two joints of a joint pair on the lower fixed frame of the Stewart platform.

θ_P Angle between two joints of a joint pair on the upper moving frame of the Stewart platform.

$\dot{x}$ Velocity x-axis component in $\mathbb{E}_I$.

$\dot{y}$ Velocity y-axis component in $\mathbb{E}_I$.

$\dot{z}$ Velocity z-axis component in $\mathbb{E}_I$.

n_p MEMS white noise angular rate x-axis component in $\mathbb{E}_{\square}$.

n_{θ_H} Sensor's white noise on the pitch attitude component.

n_x Positional Gaussian noise x-axis component.

n_y Positional Gaussian noise y-axis component.

n_z Positional Gaussian noise z-axis component.

n_q MEMS white noise angular rate y-axis component in $\mathbb{E}_{\square}$.

n_r MEMS white noise angular rate z-axis component in $\mathbb{E}_{\square}$.

n_{ϕ_H} Sensor's white noise on the roll attitude component.

n Sensor's white noise on a particular axis.

$n_{\ddot{x}}$ MEMS white noise acceleration x-axis component in $\mathbb{E}_{\square}$.

$n_{\ddot{y}}$ MEMS white noise acceleration y-axis component in $\mathbb{E}_{\square}$.

n_{ψ_H} Sensor's white noise on the yaw attitude component.

$n_{\ddot{z}}$ MEMS white noise acceleration z-axis component in $\mathbb{E}_{\square}$.

ξ_B Angle between two consecutive joints not of the same pair on the lower fixed frame of the Stewart platform.

ξ_P Angle between two consecutive joints not of the same pair on the upper moving frame frame of the Stewart platform.

ψ z-axis rotation in Euler attitude formulation.

Vectors

$\underline{\boldsymbol{\ddot{c}}}$ Acceleration, second positon derivative.

$\underline{\boldsymbol{e}}$ Quaternion attitude representation.

$\underline{\boldsymbol{b}}_i$ Bottom fixed frame ith joint's position.

$\underline{\boldsymbol{\lambda}}$ MEMS bias.

$\underline{\boldsymbol{\mu}}$ MEMS rate/acceleration random walk, modelled as a Gaussian noise.

$\underline{\boldsymbol{f}}$ Specific force.

$\underline{\boldsymbol{g}}$ Gravity vector.

$\underline{\boldsymbol{\epsilon}}$ Innovation.

$\underline{\boldsymbol{\dddot{c}}}$ Jerk, third position derivative.

$\underline{\boldsymbol{l}}_i$ ith hydraulic actuator position of Stewart platform.

$\underline{\boldsymbol{o}}$ Observation noise random variables.

$\underline{\boldsymbol{\omega}}$ Angular rate.

$\underline{\boldsymbol{p}}_i$ Upper motion frame ith joint's position.

$\underline{\boldsymbol{c}}$ Position.

$\underline{\boldsymbol{\nu}}$ Process noise random variables.

$\underline{\boldsymbol{q}}$ Quaternion vector element.

$\underline{\dot{\boldsymbol{\omega}}}$ Angular acceleration.

$\underline{\ddot{\boldsymbol{\omega}}}$ Angular jerk.

$\underline{\boldsymbol{s}}$ Stewart platform setpoints send to Motion Control Computer.

$\underline{\boldsymbol{s}}_{\ddot{c}}$ Acceleration.

$\underline{\boldsymbol{s}}_{\dot{\omega}}$ Angular acceleration.

$\underline{\boldsymbol{\sigma}}$ Standard deviation.

$\underline{\boldsymbol{\tau}}$ Lag of a system, indicating the time the system takes to reach 63 % of the final step input.

$\underline{\dot{\boldsymbol{\tau}}}$ Lag of a system over time.

$\underline{\boldsymbol{u}}$ System input.

$\underline{\dot{\boldsymbol{c}}}$ Velocity, first postion derivative.

$\underline{\boldsymbol{n}}$ MEMS angle/velocity random walk, modelled as a Gaussian noise.

$\underline{\boldsymbol{x}}$ System state.

$\underline{\hat{\boldsymbol{x}}}$ Estimated system state.

$\underline{\boldsymbol{y}}$ System observation.

$\underline{\widehat{\boldsymbol{y}}}$ Estimated system observation.

Matrices

$\boldsymbol{I}$ Identity matrix.

$\boldsymbol{K}$ Kalman gain matrix.

$\boldsymbol{\Omega}$ skew-symmetric matrix composed of the rotational rate.

$\boldsymbol{Q}$ Covariance matrix of the process noise random variables.

$\boldsymbol{R}$ Covariance matrix of the observation noise random variables.

$\boldsymbol{T}$ Transformation matrix.

$\boldsymbol{T}_{HP}$ Transformation matrix transforming vectors from the upper platform reference frame $\mathbb{E}_P$ to the HMD own reference frame $\mathbb{E}_H$.

$\boldsymbol{T}_{PI}$ Transformation matrix transforming vectors from the lower fixed frame $\mathbb{E}_B$ (equal to the inertial frame $\mathbb{E}_I$) to the upper moving frame $\mathbb{E}_P$.

$\boldsymbol{T}_{\square I}$ Transformation matrix transforming vectors from the inertial reference frame $\mathbb{E}_I$ to the placeholder reference frame $\mathbb{E}_\square$.

$\boldsymbol{T}_{ZI}$ Example transformation matrix.

$\boldsymbol{\mathcal{Y}}$ Sigma point samples matrix projected through the observation model.

$\boldsymbol{\mathcal{X}}$ Sigma-points sampled states.

$\boldsymbol{\mathcal{X}}^o$ Observation noise part of the sigma-points sampled states.

$\boldsymbol{\mathcal{X}}^v$ Process noise part of the sigma-points sampled states.

$\boldsymbol{\mathcal{X}}^x$ State part of the sigma-points sampled states.

$\boldsymbol{P}$ Covariance.

$\boldsymbol{P}_{\hat{x}}$ Covariance matrix of the Gaussian approximation of the distribution of the states.

$\boldsymbol{P}_{\hat{x}_k \hat{y}_k}$ Cross-covariance between the state sigma-points samples and the measurement points.

$\boldsymbol{P}_{\hat{y}_k}$ Observation covariance.

General state-space system components.

$\boldsymbol{B}$ Input matrix.

$\boldsymbol{A}$ State matrix.

Reference Frames

$\mathbb{E}_B$ Attached to the lower fixed frame of Stewart platform and centered with the LGP.

$\mathbb{E}_H$ Attached to the HMD and centered between the retinas of the user.

$\mathbb{E}_I$ Inertial reference frame.

$\mathbb{E}_P$ Attached to the upper moving frame of Stewart platform and centered with the UGP.

$\mathbb{E}_\square$ Placeholder non-inertial reference frame.

$\mathbb{E}_Z$ Example reference frames.

Functions

H Function describing the discrete-time observation of the state of a system.

F Function describing the discrete-time change of the state of a system.
diag Diagonal of matrix.
h Function describing the continuous-time observation of the state of a system.
f Function describing the continuous-time change of the state of a system.
$\mathrm{Q2R}$ Logarithm mapping, id est (i.e.) quaternion space to rotational space.
$\mathrm{R2Q}$ Exponential mapping, i.e. rotational space to quaternion space.
rad Convert degrees into radians.
$\mathrm{G}(s)$ Transfer function.
$\mathrm{U}(s)$ Transfer input function.
$\mathrm{Y}(s)$ Transfer output function.

Subscript

$\Box$ Placeholder for reference frames indications.
B Lower fixed frame of Stewart platform indication.
H HMD indication.
HP Upper moving frame to HMD indication.
I Inertial frame indication.
P Upper moving frame of Stewart platform indication.
d Indicate discretized.
Integer indeces.
i
j
m Indicate measured quantity.
δ Offset.
Discrete time step indications.
k Time step.
l Earlier then time step k unless mention otherwise.
0 Initial.
$\rightarrow$ Delayed to.

Superscript

$+$ Corrected delayed status.
$-$ Predicted status.
a Augmented.
e Expanded status.
o Observation noise's selection.
v Process noise's selection.
x State's selection.

Other Symbols

Ω Two-sided .
$\mathbb{S}$ Sigma-points.
σ Allan derivation.
m Averaging factor.
$m\tau_0$ Averaging time.
K Rate/Acceleration random walk coefficient.
s Complex variable.
Ci Cosine-integral function.
f_0 Cutoff frequency.
d Delay expressed in the algorithm discrete time-steps.
Δt Difference between two discrete time steps.
ε error.
$E\left|...\right|$ Expected value.
f Frequency.

$\aleph\,(\ldots,\ldots)$ Gaussian distribution.
$\boldsymbol{l}_m$ Collection of the hydraulic actuators lengths.
ν .
M Maximum amount of past estimated states necessary per delayed sensor.
B Bias instability coefficient.
$\ominus$ Difference between two unit-quaternions resulting in a angle-axis.
$\oplus$ The addition of an angle-axis attitude to unit-quaternion attitude.
$\sim$ Indicates a reverses quaternion.
$\underset{L}{\otimes}$ Preserves the left-to-right quaternion sequence.
$\underset{R}{\otimes}$ Reverses the quaternion sequence to right-to-left.
z A discrete measured sample of a sensor.
M Total amount of measured samples of a sensor.
τ_0 Discrete sample time.
t Continuous time.
τ Continuous time, different then t.
Υ Short for $\pi f_0 m \tau_0$.
$\boldsymbol{w}$ Weights of the sigma-points.

- $\boldsymbol{w}^{(c)}$ Weights to compute the covariance.
- $\boldsymbol{w}^{(m)}$ Weights to compute the mean.

N Angle/Velocity random walk coefficient.

I

Paper

Enabling a Head-Mounted-Display in an Enclosed Cabin on a Moving Base Simulator: Multiple Implementations with an Unscented Kalman Filter

***Abstract*—When placing a Head-Mounted Display (HMD) of a Virtual Reality (VR) or Augmented Reality (AR) system in the cabin of a moving base simulator to provide a larger field-of-regard, the HMD's inertial sensors will interpret the motion of the moving base simulator as if provided by the user, thus distorting the HMD's cabin-fixed pose estimation. In order to estimate a proper cabin-fixed HMD pose, this thesis proposes an Unscented Kalman Filter (UKF) that fuses the information from the simulator sensors, as well as the sensors of the VR/AR system, including its visual position sensor and IMU. For this, three sensor configurations are proposed and tested offline deterministically based on real data. Although an unexpected latency in the HMD's visual pose sensor prevented a full validation of the algorithms, both the configuration with an IMU attached to the motion base, and the configuration without an IMU but with second-order additive noise variables added to the UKF, resulted in a proper, tight innovation sequence, indicating accurate system state estimation. The third configuration, one using the novel approach of using setpoints, provided a tight, but out-of-bounds, innovation sequence indicative of high accuracy, but also diminished tuning efficacy. It is also sensitive to stability issues when estimating the motion system's dynamics when no movement is present on the moving base. The unaltered setpoint method is not recommended as a solution. However, a hybrid solution of the setpoint and second-order additive noise variables methods should be investigated in a follow-up study to combine stability with high accuracy. Another recommendation is to test all configurations interactively, i.e. online.**

I. Introduction

A Virtual Reality (VR)/Augmented Reality (AR) system provides an immersive experience as the user's gaze of the virtual world is rendered onto the Head-Mounted Display (HMD). If a VR/AR system is placed in a moving base simulator, it could provide a larger Field-Of-Regard (FOR) than a conventional out-of-window visual system currently in use. In particular, a larger FOR would improve the fidelity of rotorcraft and fighter jet simulators.

Nowadays, head tracking in VR/AR systems are commonly performed by a combination of inertial and vision-based sensors. Inertial sensors present on a HMD, namely Micro Electro Mechanical Systems (MEMS) based Inertial Measurement Units (IMU), provide high-frequency low-latency position and attitude, hereafter referred to as pose, updates. However, IMU's inherent implementation for pose estimation causes drift over time [1]. This drift is eliminated by vision-based sensors that measure the pose of the HMD, commonly present in the form of lightpoint or image tracking, albeit at a lower frequency than the MEMS based IMU. In turn, the low frequency pose updates of the vision-based sensor are resolved by the IMU high-frequency updates.

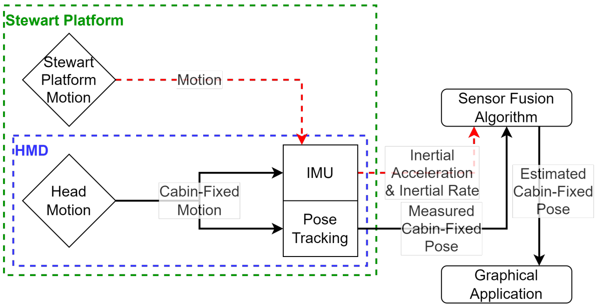


Fig. 1. Highlighting the problem by the red dashed arrow of placing a traditional VR system within an enclosed cabin on a moving base simulator.

The integration of a VR/AR system with the HMD's IMU active has already been done in open cabins on top of motion simulators using off-the-shelf components with open-source software packages, such as as [2], by placing the visual pose tracking onto the walls or ceiling [3], [4], i.e., rigid to the inertial reference frame, and 'simply' deducting the motion simulator's pose from the HMD's pose.

However, when placing such a system within an enclosed cabin on a motion simulator, where the visual pose tracking has to be done within the cabin, i.e., relative to the motion simulator, the HMD cabin-fixed pose, i.e., relative to the cabin, is incorrectly estimated by traditional off-the-shelf VR/AR system. This is due to the fact that the VR/AR system expects all of its sensors to measure relative to the same reference frame, in most cases the inertial reference frame. More specifically, the IMU sensor interprets the combined motion of both the moving base as well as head movements solely as the latter, which inevitably distort the estimated head-pose, as indicated by the red dashed line in Fig. 1. Consequently, the distorted head-pose rendered visual causes a mismatch between the user's visual perception and their expected pose within the cabin [4]–[6], which would have a significant negative impact on the user experience, thus practically rendering the visual system unusable.

The aim of this research is to present a solution on the implementation of a HMD with both an IMU and visual pose tracking inside an enclosed cabin on a moving base simulator. This will be done by developing a sensor fusion algorithm that generates a proper cabin-fixed pose estimate. Previous attempts to find solutions to the aforementioned distorted head-pose issue include using purely visual pose tracking [7]; integrating the cabin-fixed differential (between motion simulator and HMD) IMU measurements [8], [9]; and using the vehicle state fixed HMD's IMU measurements solely for the HMD's attitude, while using the visual pose tracking for attitude drift correction and position estimation [10]. Lessons learned from previous approaches will be taken into account by including the HMD's IMU; using it to estimate the cabin-fixed position alongside its attitude; and estimating the state of the motion simulator concurrent to the HMD's, while keeping both states fully observable. Furthermore, this paper will present multiple configurations to estimate the HMD's cabin-fixed pose, whereby the motion simulator's sensory layout is altered in order to allow for versatility of the implementation of the solution. To this end, the necessary characteristics of the state estimation algorithm are discussed, and the influence of the motion simulator's sensory layout on the HMD's cabin-fixed estimated pose is explored. In particular, the impact of a motion simulator's IMU or its control setpoints is investigated. In addition, the multiple configurations will be compared to solely visual pose tracking.

To attain the goal of this research, first, the systems and their sensors are introduced in Section II. Second, the state estimation algorithm is discussed and briefly presented in Section III. In Section IV, three configurations are presented based on the available sensors and their stochastic system models are presented for use in the proposed algorithm. Next, in Section V, brief descriptions of the data gathering process and the performance metrics are given, followed by a short section on the noise parameters and initial values to be used in the algorithm. Lastly, given the results shown and discussed in Sections VIII and IX respectively, this paper concludes in Section X.

II. Sensors and Systems

Understanding the sensors and their limitations will help with the overall understanding of the design of the stochastic models in subsequent sections. Therefore, this section will identify the sensors used in order to ease the replication or modification of this research in other settings. The noise parameters will be detailed in Section VI.

A. Virtual Reality System

1) HMD: The HMD used in this study, the Pimax 8k-x, shown in Fig. 2, was chosen based on its availability and design coherency with other SteamVR devices. The compatibility with SteamVR devices makes it possible to read out the raw IMU data by making use of the open-source software tools like libsurvive [11]. The HMD's IMU is a MPU-6500 running at 1000 Hz [12], [13] and consist of one accelerometer and one gyroscope per orthogonal axis, thus three of each in total.

The HMD has its own right-handed reference frame $\mathbb{E}_H$, shown in Fig. 2, which is located at the center between the user's retinas [14] and fixed to the HMD.

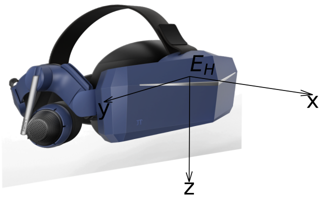


Fig. 2. Pimax 8k-x [15]

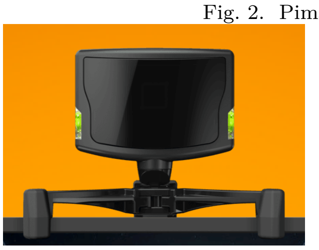


Fig. 3. TrackIR 5 [16]

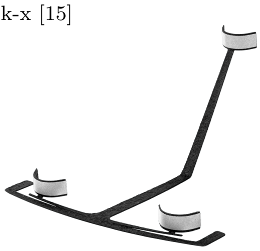


Fig. 4. TrackIR 5 clip [16]

2) Visual Tracking: The visual pose tracking nominally employed by a SteamVR device, Base Stations version 1.0 and 2.0, are not designed to be used in a moving environment [10]. Therefore, the TrackIR 5 visual pose tracking system was chosen for its capability of withstanding the movement of the moving base, availability, and ease of installation due to its small size, as shown in Fig. 3. The TrackIR 5 is an infrared (IR) transceiver, receiving the IR refracted by a passive TrackClip attached to the user's head, shown in Fig. 4. It is assumed that the known shape allows the TrackIR 5 proprietary algorithm to identify the pose of the TrackClip at an update frequency of approximately 65 Hz. However, the proprietary software layer hinders the access to the raw unfiltered measurements. This further prevents the development of a customized observation and noise model for the TrackIR 5.

B. SIMONA Research Simulator (SRS)

While a range of different moving base systems are available, the system often chosen, and available for integration and testing, is a Stewart platform. One such Stewart platform is the flight simulator at TU Delft, namely the SIMONA Research Simulator (SRS).

A Stewart platform inherently has six degrees of freedom and is formed by two connected rigid frames [17], [18], the upper moving and the bottom fixed frames, as depicted in Fig. 5. These two frames are connected by six

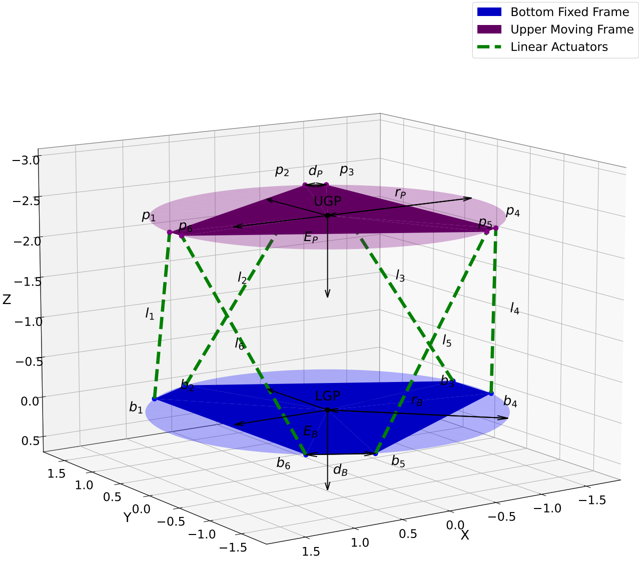


Fig. 5. Simplified geometric representation of a Stewart platform.

cylindrical actuators with Hooke joints at each end, shown as the six dotted lines in Fig. 5. Hooke joints provide 2 degrees of freedom.

The bottom fixed frame, or the base frame, is fixed to the ground and has three pair of Hooke joints, six in total. These Hooke joints are placed at a constant radius r_B from the center point of the frame, namely Lower Gimbal Point (LGP), also depicted in Fig. 5. Each Hooke joint pairs are spaced 120° on the lower fixed frame. The spacing between the two joints in a pair is symbolized by d_B. In addition, the LGP is here the origin of the inertial reference frame $\mathbb{E}_I$ assuming a flat and non-rotating Earth.

The upper moving frame has a similar geometry, with all six joints placed at a constant radius r_P and spacing between the two joints of a pair is given by d_P. Also, note the subscript P meaning 'Platform', demarking both symbols to the upper platform, i.e., upper moving frame. The upper moving reference frame $\mathbb{E}_P$ is attached to the upper moving frame and its origin is at the center of all upper joints, the Upper Gimbal Point (UGP).

Moreover, $\boldsymbol{b}_i^I$ and $\boldsymbol{p}_i^P$ represent these joint coordinates of the bottom and upper frame respectively, where the superscript I and P indicates that their components are expressed in the inertial $\mathbb{E}_I$ and upper moving reference frame $\mathbb{E}_P$. By expressing the joint coordinates in their respective frames the values can be found by basic trigonometry.

Moreover, the data of its sensors are published onto the Delft University Environment for Communication and Activation (DUECA). This middleware enables real-time simulations by using both inter-module communication as well as communication between different modules on a distributed network of computers [19]. Moreover, DUECA uses a publish and subscribe design that enables modules, a self-contained possible real-time computational element, to activate when receiving or sending data. The publishing of the sensor data happens at a specified rate of 100 Hz.

1) Linear Encoders: The SRS's pose is not measured directly but is computed indirectly from the length of the six hydraulic actuators [20]. The length of each hydraulic actuator is measured by one Heidenhain LC 415 linear encoder [21], depicted in Fig. 7. Once measured, the data is published onto DUECA at a chosen rate of 100 Hz.

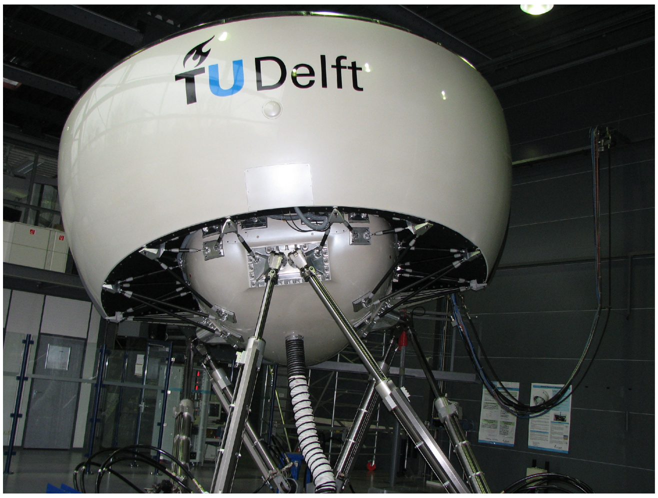


Fig. 6. SIMONA Research Simulator (SRS) using a Stewart platform as moving base simulator. [22]

Fig. 7. Heidenhain LC 415 [21], an absolute linear encoder used for measuring hydraulic actuator length on the SRS.

Fig. 8. SRS IMU

The measured length l_{i_m} by these six linear encoders of each hydraulic actuators can be described as a function of the Stewart platform's joint coordinates $\boldsymbol{b}_i^I$ and $\boldsymbol{p}_i^P$, position $\boldsymbol{c}_P^I$, attitude $\boldsymbol{e}_P^I$, and a Gaussian white noise value n_{l_i}:

$$l_{i_m} = \left\| \boldsymbol{c}_P^I + \boldsymbol{T}_{PI}^T \boldsymbol{p}_i^P - \boldsymbol{b}_i^I \right\| + n_{l_i} \quad \forall i = 1, \dots, 6 \tag{1}$$

Here, $\boldsymbol{c}_P^I$ is the position of the UGP from the LGP and is expressed in Cartesian coordinates w.r.t. $\mathbb{E}_I$:

$$\boldsymbol{c}_P^I = \begin{pmatrix} x_P & y_P & z_P \end{pmatrix} \{\mathbb{E}_I\} \tag{2}$$

$\{\mathbb{E}_I\}$ is the symbolic notation for the column of the unity vectors of the inertial reference frame $\mathbb{E}_I$.

In Eq. (1), the transposed transformation matrix $\boldsymbol{T}_{PI}^T$ transforms the Cartesian coordinates from the Inertial $\mathbb{E}_I$

to the upper moving reference frame $\mathbb{E}_P$. Such a matrix is dependent on the attitude of the upper moving frame $\boldsymbol{e}_P^I$ w.r.t. $\mathbb{E}_I$. This attitude $\boldsymbol{e}_P^I$ is represented by the Euler-Rodrigues quaternion formulation in order to have computationally superior transformation matrices when compared to the more intuitive Euler angle formulation [23]:

$$\boldsymbol{e}_P^I = \begin{pmatrix} e_{0_P} & e_{x_P} & e_{y_P} & e_{z_P} \end{pmatrix} \begin{Bmatrix} 1 \\ \mathbb{E}_I \end{Bmatrix} \tag{3}$$

And with the norm equal to one to represent an attitude:

$$\left\| \boldsymbol{e}_P^I \right\| = e_{0_P}^2 + e_{x_P}^2 + e_{y_P}^2 + e_{z_P}^2 = 1 \tag{4}$$

Having defined the quaternion attitude $\boldsymbol{e}_P^I$, the transformation matrix $\boldsymbol{T}_{PI}$ can be written in terms of the quaternion:

$$\boldsymbol{T}_{PI} = 2 \begin{bmatrix} 0.5 - e_{y_P}^2 - e_{z_P}^2 & e_{x_P} e_{y_P} + e_{0_P} e_{z_P} & e_{x_P} e_{z_P} - e_{0_P} e_{y_P} \\ e_{x_P} e_{y_P} - e_{0_P} e_{z_P} & 0.5 - e_{x_P}^2 - e_{z_P}^2 & e_{y_P} e_{z_P} + e_{0_P} e_{x_P} \\ e_{x_P} e_{z_P} + e_{0_P} e_{y_P} & e_{y_P} e_{z_P} - e_{0_P} e_{x_P} & 0.5 - e_{x_P}^2 - e_{y_P}^2 \end{bmatrix} \tag{5}$$

2) Inertial Measurement Unit (IMU): The SRS's IMU is an ISIS-IMU Rev. C model, a MEMS based IMU, depicted in Fig. 8. The specific force $\underline{\boldsymbol{f}}_{P_m}$ and angular rate $\underline{\boldsymbol{\omega}}_{P_m}$ measured by the IMU is subsequently published onto DUECA at a chosen rate of 100 Hz. This IMU is attached to the upper moving frame of the SRS but not in the center, i.e., UGP, of said frame. Moreover, the distance $\underline{\boldsymbol{c}}_{\text{IMU}}$ from the UGP to the IMU expressed in the upper moving reference frame $\mathbb{E}_P$ was measured to be [24]:

$$\boldsymbol{c}_{\text{IMU}}^P = \begin{pmatrix} 0.305 & 0 & 0.0105 \end{pmatrix} \{\mathbb{E}_P\} \tag{6}$$

3) Setpoints: The setpoints are the output of the motion cueing system. They consist of position, velocity, acceleration, and attitude expressed in the inertial reference frame $\mathbb{E}_I$, as well as the angular rate and angular acceleration given in the upper moving reference frame $\mathbb{E}_P$. In parallel, these setpoints are limited by the physical aspects of the Stewart platform, given the current estimated position, velocity, acceleration, attitude, angular rate and angular acceleration of the Stewart platform in this case. Some prominent characteristics of the SRS related to the setpoints are that it has a fast, low lag/latency response between setpoints and actual motion. Moreover, the setpoints are given not only as the zeroth order state, e.g., position, but up to second order, e.g., the acceleration.

A motion system with *"an inner loop feedback and feedback linearising control result in first order response of the system, from desired to actual accelerations."* [25] Thus, if this assumption holds where an inner loop feedback and feedback linearising control systems are used to control the Stewart platform, a simplified model can be construed to predict the state of the Stewart platform. Given the desired acceleration, i.e., setpoints, the actual acceleration can be approximated by the response of a first order system.

No literature was found on the implementation of setpoints into a Kalman Filter (KF). However, it can be assumed that these setpoints do provide a window to the future state of the system. Moreover, it is highly likely that the noise models on these setpoints are non-Gaussian, so an algorithm that is at least lenient to non-Gaussian variables needs to be selected.

III. Algorithm

In order to fuse the sensors' measurements together and estimate the HMD's cabin-fixed pose, the Unscented Kalman Filter (UKF) was used. In the past, the UKF was used successfully to estimate the pose of both HMD [26]–[29] and Stewart platform [30] separately.

The UKF was chosen due to the fact that it handles nonlinear, continuous, multivariate problems, with noise that can be modestly non-Gaussian [31]. However, it does not handle occlusions, which would improve robustness and allow for entities to be hidden. In practice, the UKF was found to have similar computational cost to the EKF [30], [32]–[34], but improved estimation performance.

A. KF Framework

The UKF is a minimum mean-square-error sequential estimator [31] based on the KF framework [35].

The KF framework estimates the state $\underline{\hat{\boldsymbol{x}}}$ based on direct or indirect measurements of the actual state $\underline{\boldsymbol{x}}$. The KF framework estimates the state $\underline{\hat{\boldsymbol{x}}}$ of a system by combining measurements of the system, including the knowledge of the measurement's probability distribution, with the previous state and its covariance in a Gaussian approximate optimal manner [31]. The KF framework works in two phases, the prediction phase and the correction or update phase.

1) Prediction Phase: The previous estimated state $\underline{\hat{\boldsymbol{x}}}_{k-1}$ and previous state covariance $\boldsymbol{P}_{\hat{x}_{k-1}}$ are projected to the next time step ($k-1 \rightarrow k$) to the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ and predicted state covariance $\boldsymbol{P}_{\hat{x}_k}^-$. The predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ is the projection in time of the previous estimated state $\underline{\hat{\boldsymbol{x}}}_{k-1}$ itself and an optional external input $\underline{\boldsymbol{u}}_{k-1}$ using the process model F, usually based on a physical model of the system:

$$\underline{\boldsymbol{x}}_k^- = \mathrm{F}\left(\underline{\boldsymbol{x}}_{k-1}, \underline{\boldsymbol{u}}_{k-1}\right) \tag{7}$$

Meanwhile, the uncertainty of the process, i.e., process noise $\underline{\boldsymbol{\nu}}$, is appended to the previous state covariance $\boldsymbol{P}_{\hat{x}_{k-1}}$ resulting in the predicted state covariance $\underline{\hat{\boldsymbol{x}}}_k^-$.

2) Correction Phase: The predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ is corrected to the posterior state $\underline{\hat{\boldsymbol{x}}}_k$ using an observation $\underline{\boldsymbol{y}}_k$ of the system. This is achieved by adding the Kalman Gain K scaled difference between the estimated $\underline{\hat{\boldsymbol{y}}}_k$ and the actual observation $\underline{\boldsymbol{y}}_k$, i.e., the innovation $\underline{\boldsymbol{\epsilon}}_k$, to the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$. Here, the estimated observation $\underline{\hat{\boldsymbol{y}}}_k$ is the result of observing the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ through the observation model H,

$$\underline{\hat{\boldsymbol{y}}}_k = \mathrm{H}\left(\underline{\hat{\boldsymbol{x}}}_k^-\right) \tag{8}$$

The Kalman Gain $\boldsymbol{K}$ mentioned above minimizes the difference between the estimated $\underline{\hat{\boldsymbol{y}}}_k$ and actual observation $\underline{\boldsymbol{y}}_k$ by adjusting the estimated state $\underline{\hat{\boldsymbol{x}}}_k$ while taking into account the predicted state covariance $\boldsymbol{P}^{-}_{\hat{x}_k}$ and the observation noise $\underline{\boldsymbol{o}}$.

B. *Unscented Kalman Filter (UKF)*

Using the outlined KF framework, the UKF manages to solve discrete-time nonlinear systems with sequential probabilistic inference [31] that can be described by the combination of the process F and observation H models:

$$\underline{\boldsymbol{x}}_k = \mathrm{F}\left(\underline{\boldsymbol{x}}_{k-1}, \underline{\boldsymbol{u}}_{k-1}, \underline{\boldsymbol{\nu}}_{k-1}\right) \tag{9}$$

$$\underline{\boldsymbol{y}}_k = \mathrm{H}\left(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{o}}_k\right) \tag{10}$$

The expected value $E\left|...\right|$ of both the process $\underline{\boldsymbol{\nu}}$ and observation noise random variables $\underline{\boldsymbol{o}}$ are modelled as zero-mean Gaussian noises. Their respective autocovariance are equal to $\boldsymbol{Q}$ and $\boldsymbol{R}$ by definition:

$$E\left|\underline{\boldsymbol{\nu}}(t)\right| = 0,\ E\left|\underline{\boldsymbol{\nu}}(t)\underline{\boldsymbol{\nu}}^T(\tau)\right| = \boldsymbol{Q}\ \rightarrow \underline{\boldsymbol{\nu}}(t) \approx \aleph\left(0, \boldsymbol{Q}\right) \tag{11}$$

$$E\left|\underline{\boldsymbol{o}}(t)\right| = 0,\ E\left|\underline{\boldsymbol{o}}(t)\underline{\boldsymbol{o}}^T(\tau)\right| = \boldsymbol{R}\ \rightarrow \underline{\boldsymbol{o}}(t) \approx \aleph\left(0, \boldsymbol{R}\right) \tag{12}$$

The UKF uses the true nonlinear process F and observation models H, rather than approximating them. However, the UKF approximates the assumed Gaussian distribution of the state, and represents this distribution with a minimal set of deterministically sampled points around the state itself named sigma-points. The sigma-points' 'weights' and 'width' around the state are based on three parameters α, β, and κ. The propagation of this set of sampled points through the true nonlinear process, and subsequent true observation model will result in an expression of the posterior mean and its covariance. Also, by including the process $\underline{\boldsymbol{\nu}}$ and observation noise $\underline{\boldsymbol{o}}$ into the state their distributions are represented directly in the sigma-point samples and propagated through the process F and observation model H accordingly.

IV. Configurations and their Stochastic Systems

In general, the sensors shape the state $\underline{\boldsymbol{x}}$, input $\underline{\boldsymbol{u}}$ and observation $\underline{\boldsymbol{y}}$, and in combination with the algorithm, they shape the stochastic process F and observation H models. Given the available sensors presented in Section II, it is possible to create multiple sensor combinations that are capable of estimating the HMD's cabin-fixed pose while using the HMD's IMU in order to enable fast pose updates or, in the nomenclature of KF, fast predictions. Therefore, three different configurations were chosen for the following reasons:

- **Configuration 1** retains the HMD's fast pose updates with the minimal amount of physical sensors systems by only using the linear encoders of the SRS alongside the in-cabin visual pose tracking and the aforementioned HMD's IMU. Therefore, this configuration removes the motion base's IMU that was necessary in previous work [8], [9] while maintaining full pose estimation of both the HMD and SRS alongside other states and HMD's IMU bias observability.
- **Configuration 2** adds fast pose updates to the SRS subsystem by attaching an additional IMU to the SRS platform alongside the linear encoders, in-cabin visual pose tracking and the aforementioned HMD's IMU. However, this configuration does not use differential IMU [8], [9], but uses both IMU's measurements to help predict each system's state in the inertial frame, thereby reducing process model complexity while maintaining full state observability and full pose estimation of both the HMD and SRS alongside other states.
- **Configuration 3** uses the SRS's control system setpoints in a novel manner to help predict the state alongside the linear encoders, in-cabin visual pose tracking and the aforementioned HMD's IMU. This configuration was designed to ascertain if improvements could be made to the state estimation when no IMU is available on the Stewart platform, as not all Stewart platform are equipped with an IMU.

The HMD's cabin-fixed pose, i.e., the pose that should be used for the graphical application, is the HMD's pose from the UGP expressed in $\mathbb{E}_P$:

$$\boldsymbol{c}^P_{HP} = \begin{pmatrix} x_{HP} & y_{HP} & z_{HP} \end{pmatrix} \{\mathbb{E}_P\} \tag{13}$$

$$\boldsymbol{e}^P_H = \begin{pmatrix} e_{0_H} & e_{x_H} & e_{y_H} & e_{z_H} \end{pmatrix} \begin{Bmatrix} 1 \\ \{\mathbb{E}_P\} \end{Bmatrix} \tag{14}$$

A. *Configuration 0 — Baseline*

The visual pose tracking attached to the cabin already measures the HMD's proper pose, i.e., the cabin-fixed pose, with the position indicated by $\boldsymbol{c}^P_{HP_m}$ and attitude as $\boldsymbol{e}^P_{H_m}$. By disabling the HMD's IMU the problem highlighted in Fig. 1 by the red dashing line is removed. However, this also removes the fast pose updates provided by the HMD's IMU. This configuration is the minimal sensor configuration to achieve a cabin-fixed pose estimation of a VR/AR system within an enclosed cabin on any moving base simulator and will serve as a baseline used for comparison.

This configuration is further detailed in Fig. 9. Here, the cabin-fixed HMD pose $\boldsymbol{c}^P_{HP_m}$, $\boldsymbol{e}^P_{H_m}$ is observed by the visual pose tracking. The visual pose tracking used, TrackIR 5, has a proprietary abstraction layer that prevents the reading of the underlying raw measurements, as stated in Section II-A2, and are thus taken as is.

B. *Configuration 1 — IMU on HMD*

Configuration 1 enables the use of the HMD's IMU in order to enable the fast pose updates with the minimal amount of physical sensors. As the HMD's IMU measures to the inertial frame and not to the upper motion reference frame $\mathbb{E}_P$, the state of the UGP should be included as well.

This configuration, as shown in Fig. 10, has the linear encoders indirectly observing the UGP's position $\underline{\boldsymbol{c}}_P$ and

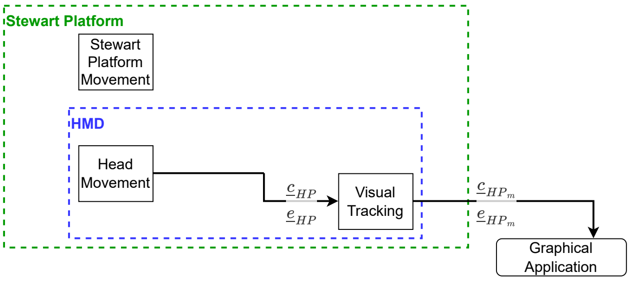


Fig. 9. Configuration 0: A VR/AR system in an enclosed cabin on a Stewart platform, with only visual pose tracking measuring the pose of the HMD.

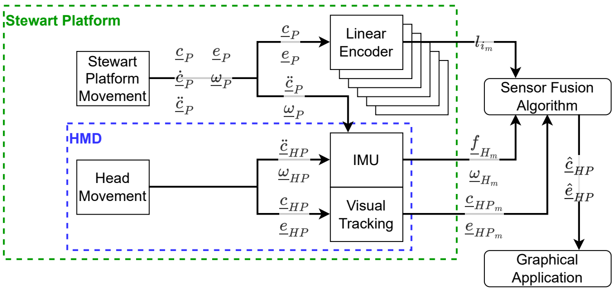


Fig. 10. Configuration 1: A VR/AR system with an IMU in an enclosed cabin on a Stewart platform. The kinematics are adapted by also estimating the Stewart platform's pose.

attitude $\underline{\boldsymbol{e}}_P$ trough the inverse kinematics, as per Eq. (1). In order for the UKF to be able to estimate the UGP's position $\underline{\boldsymbol{c}}_P$ and attitude $\underline{\boldsymbol{e}}_P$, the UGP's acceleration $\ddot{\underline{\boldsymbol{c}}}_P$ and the angular acceleration $\dot{\underline{\boldsymbol{\omega}}}_P$ are seen as white, Gaussian noise sources $\underline{\boldsymbol{\nu}}_{\ddot{c}_P}$ and $\underline{\boldsymbol{\nu}}_{\dot{\omega}_P}$ respectively. These are part of the process noise $\underline{\boldsymbol{\nu}}_P$:

$$\underline{\boldsymbol{\nu}}_P = \begin{Bmatrix} \boldsymbol{\nu}^I_{\ddot{c}_P} \\ \boldsymbol{\nu}^P_{\dot{\omega}_P} \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} \nu_{\ddot{x}_P} & \nu_{\ddot{y}_P} & \nu_{\ddot{z}_P} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \nu_{\dot{p}_P} & \nu_{\dot{q}_P} & \nu_{\dot{r}_P} \end{pmatrix} \{\mathbb{E}_P\} \end{Bmatrix} \tag{15}$$

In turn, these noise sources increase the covariance on the estimated velocity $\dot{\underline{\boldsymbol{c}}}_P$ and angular rate $\dot{\underline{\boldsymbol{e}}}_P$ respectively. Consequently, the aforementioned increase in covariance propagates through the integration scheme to the estimated position $\underline{\boldsymbol{c}}_P$ and attitude $\underline{\boldsymbol{e}}_P$ covariances. This finally allows the UKF to estimate the UGP's position $\hat{\underline{\boldsymbol{c}}}_P$, attitude $\underline{\boldsymbol{e}}_P$, velocity $\dot{\underline{\boldsymbol{c}}}_P$, and angular rate $\dot{\underline{\boldsymbol{e}}}_P$ [36].

Based on the process noise vector $\underline{\boldsymbol{\nu}}_P$ and the lack of input vector $\underline{\boldsymbol{u}}_P$, the Stewart platform's part $\underline{\boldsymbol{x}}_P$ of the whole stochastic state $\underline{\boldsymbol{x}}$ is defined as:

$$\underline{\boldsymbol{x}}_P = \begin{pmatrix} \boldsymbol{c}^{IT}_P & \dot{\boldsymbol{c}}^{IT}_P & \boldsymbol{e}^{IT}_P & \boldsymbol{\omega}^{PT}_P \end{pmatrix}^T \tag{16}$$

Here, the inertial position $\boldsymbol{c}^I_P$ and attitude $\boldsymbol{e}^I_P$ w.r.t. $\mathbb{E}_I$ were already defined in Eq. (2) and (3) respectively. The inertial velocity $\dot{\boldsymbol{c}}^I_P$ of the UGP is expressed in the inertial reference frame $\mathbb{E}_I$:

$$\dot{\boldsymbol{c}}^I_P = \begin{pmatrix} \dot{x}_P & \dot{y}_P & \dot{z}_P \end{pmatrix} \{\mathbb{E}_I\} \tag{17}$$

Finally, the angular rates $\boldsymbol{\omega}^P_P$ are expressed in the upper motion reference frame $\mathbb{E}_P$:

$$\boldsymbol{\omega}^P_P = \begin{pmatrix} p_P & q_P & q_P \end{pmatrix} \{\mathbb{E}_P\} \tag{18}$$

In this equation, $\{\mathbb{E}_P\}$ is the symbolic notation for the column of the unity vectors of the inertial reference frame $\mathbb{E}_P$.

The HMD subsystem state vector $\underline{\boldsymbol{x}}_H$, process noise vector $\underline{\boldsymbol{\nu}}_H$, and input vector $\underline{\boldsymbol{u}}_H$ are different due to the HMD's IMU. The HMD's IMU feeds the prediction model F_H of the HMD's subsystem and thus its measurements will serve as the only input $\underline{\boldsymbol{u}}$:

$$\underline{\boldsymbol{u}} = \begin{Bmatrix} \underline{\boldsymbol{u}}_P \\ \underline{\boldsymbol{u}}_H \end{Bmatrix} = \begin{Bmatrix} 0 \\ \boldsymbol{f}^H_{H_m} \\ \boldsymbol{\omega}^H_{H_m} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \begin{pmatrix} f_{x_{H_m}} & f_{y_{H_m}} & f_{z_{H_m}} \end{pmatrix} \{\mathbb{E}_H\} \\ \begin{pmatrix} p_{H_m} & q_{H_m} & r_{H_m} \end{pmatrix} \{\mathbb{E}_H\} \end{Bmatrix} \tag{19}$$

By including the IMU bias $\underline{\boldsymbol{\lambda}}$ to the state $\underline{\boldsymbol{x}}$, said bias $\underline{\boldsymbol{\lambda}}$ can be estimated online during the correction phase as it is observable. The IMU measurements can be further broken down by also including the random walk in velocity $\underline{\boldsymbol{n}}_f$ and angle $\underline{\boldsymbol{n}}_\omega$ [1], [37]–[39], a Gaussian white noise source, resulting in:

$$\underline{\boldsymbol{f}}_{H_m} = \underline{\boldsymbol{f}}_H + \underline{\boldsymbol{n}}_{f_H} + \underline{\boldsymbol{\lambda}}_{f_H} \tag{20}$$

$$\underline{\boldsymbol{\omega}}_{H_m} = \underline{\boldsymbol{\omega}}_H + \underline{\boldsymbol{n}}_{\omega_H} + \underline{\boldsymbol{\lambda}}_{\omega_H} \tag{21}$$

In addition to the velocity/angle random walk as a Gaussian white source on the IMU, the change over time of the bias $\dot{\underline{\boldsymbol{\lambda}}}$, i.e., acceleration and rate random walk, are defined as $\underline{\boldsymbol{\mu}}_f$ and $\underline{\boldsymbol{\mu}}_\omega$ [1], [37]–[39].

In addition to the Stewart platform's process noise vector $\underline{\boldsymbol{\nu}}_P$, defined in Eq. (15), the process noise vector $\underline{\boldsymbol{\nu}}$ can be defined as:

$$\underline{\boldsymbol{\nu}} = \begin{Bmatrix} \underline{\boldsymbol{\nu}}_P \\ \underline{\boldsymbol{\nu}}_H \end{Bmatrix} = \begin{Bmatrix} \underline{\boldsymbol{\nu}}_P \\ \boldsymbol{n}^H_{f_H} \\ \boldsymbol{\mu}^H_{f_H} \\ \boldsymbol{n}^H_{\omega_H} \\ \boldsymbol{\mu}^H_{\omega_H} \end{Bmatrix} = \begin{Bmatrix} \underline{\boldsymbol{\nu}}_P \\ \begin{pmatrix} n_{f_{x_H}} & n_{f_{y_H}} & n_{f_{z_H}} \end{pmatrix} \{\mathbb{E}_H\} \\ \begin{pmatrix} \mu_{f_{x_H}} & \mu_{f_{y_H}} & \mu_{f_{z_H}} \end{pmatrix} \{\mathbb{E}_H\} \\ \begin{pmatrix} n_{p_H} & n_{q_H} & n_{r_H} \end{pmatrix} \{\mathbb{E}_H\} \\ \begin{pmatrix} \mu_{p_H} & \mu_{q_H} & \mu_{r_H} \end{pmatrix} \{\mathbb{E}_H\} \end{Bmatrix} \tag{22}$$

Besides the Stewart platform's state $\underline{\boldsymbol{x}}_P$ established in Eq. (16), the HMD part $\underline{\boldsymbol{x}}_H$ of the whole stochastic state $\underline{\boldsymbol{x}}$ is defined, based on the input vector $\underline{\boldsymbol{u}}_H$, as:

$$\underline{\boldsymbol{x}} = \begin{Bmatrix} \underline{\boldsymbol{x}}_P \\ \underline{\boldsymbol{x}}_H \end{Bmatrix} = \begin{Bmatrix} \underline{\boldsymbol{x}}_P \\ \boldsymbol{c}^I_H \\ \dot{\boldsymbol{c}}^I_H \\ \boldsymbol{e}^I_H \\ \boldsymbol{\lambda}^P_{f_H} \\ \boldsymbol{\lambda}^P_{\omega_H} \end{Bmatrix} = \begin{Bmatrix} \underline{\boldsymbol{x}}_P \\ \begin{pmatrix} x_H & y_H & z_H \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \dot{x}_H & \dot{y}_H & \dot{z}_H \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} e_{0_H} & e_{x_H} & e_{y_H} & e_{z_H} \end{pmatrix} \begin{Bmatrix} 1 \\ \mathbb{E}_I \end{Bmatrix} \\ \begin{pmatrix} \lambda_{x_H} & \lambda_{y_H} & \lambda_{z_H} \end{pmatrix} \{\mathbb{E}_H\} \\ \begin{pmatrix} \lambda_{p_H} & \lambda_{q_H} & \lambda_{r_H} \end{pmatrix} \{\mathbb{E}_H\} \end{Bmatrix} \tag{23}$$

As shown, the last rows of the state $\underline{\boldsymbol{x}}_H$ introduce the bias of both the accelerometer $\underline{\boldsymbol{\lambda}}_{f_H}$ and the gyroscope $\underline{\boldsymbol{\lambda}}_{\omega_H}$ so the UKF can estimate them online in addition to the rest of the state.

Given the definitions of the state $\underline{\boldsymbol{x}}$, the input $\underline{\boldsymbol{u}}$, and the process noise $\underline{\boldsymbol{\nu}}$, a continuous version of the process model f, in preparation of the discrete process model F, can be constructed:

$$\begin{aligned}
\dot{\underline{\boldsymbol{x}}}(t) &= \mathrm{f}(\underline{\boldsymbol{x}}(t), \underline{\boldsymbol{u}}(t), \underline{\boldsymbol{\nu}}) \\
\begin{Bmatrix} \dot{\underline{\boldsymbol{x}}}_P \\ \dot{\underline{\boldsymbol{x}}}_H \end{Bmatrix} &= \begin{Bmatrix} \mathrm{f}_P(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{\nu}}_P) \\ \mathrm{f}_H(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{u}}_H(t), \underline{\boldsymbol{\nu}}_H) \end{Bmatrix} \\
\begin{Bmatrix} \dot{\boldsymbol{c}}_P^I \\ \ddot{\boldsymbol{c}}_P^I \\ \dot{\boldsymbol{e}}_P^I \\ \dot{\boldsymbol{\omega}}_P^P \\ \dot{\boldsymbol{c}}_H^I \\ \ddot{\boldsymbol{c}}_H^I \\ \dot{\boldsymbol{e}}_H^I \\ \dot{\boldsymbol{\lambda}}_{f_H}^H \\ \dot{\boldsymbol{\lambda}}_{\omega_H}^H \end{Bmatrix} &= \begin{Bmatrix} \dot{\boldsymbol{c}}_P^I \\ \boldsymbol{\nu}_{\ddot{c}_P}^I \\ \boldsymbol{\Omega}_{e_P} \boldsymbol{\omega}_P^P \\ \boldsymbol{\nu}_{\dot{\omega}_P}^P \\ \dot{\boldsymbol{c}}_H^I \\ T_{HI}^T \left\{ \boldsymbol{f}_{H_m}^H - \boldsymbol{\lambda}_{f_H}^H - \boldsymbol{n}_{f_H}^H \right\} + \boldsymbol{g} \\ \boldsymbol{\Omega}_{e_H} \left\{ \boldsymbol{\omega}_{H_m}^H - \boldsymbol{\lambda}_{\omega_H}^H - \boldsymbol{n}_{\omega_H}^H \right\} \\ \boldsymbol{\mu}_{f_H}^H \\ \boldsymbol{\mu}_{\omega_H}^H \end{Bmatrix}
\end{aligned} \tag{24}$$

Here, T_{HI}^T is based on the attitude $\boldsymbol{e}_H^I$ in a similar fashion as in Eq. (5). This results in the transformation matrix changing the coordinates from $\mathbb{E}_H$ to $\mathbb{E}_I$. Moreover, $\boldsymbol{\Omega}_{e_\square}$ is a quaternion based matrix facilitating its own rotation based on a rotation rate $\boldsymbol{\omega}_\square^\square$ w.r.t. $\mathbb{E}_\square$. Thus, depending on which subsystem is the focus of the equation, the SRS or HMD, the $\square$ subscript can be replaced with P or H respectively.

$$\boldsymbol{\Omega}_{e_\square} = \frac{1}{2} \begin{bmatrix} -e_{x_\square} & -e_{y_\square} & -e_{z_\square} \\ e_{0_\square} & -e_{z_\square} & e_{y_\square} \\ e_{z_\square} & e_{0_\square} & -e_{x_\square} \\ -e_{y_\square} & e_{x_\square} & e_{0_\square} \end{bmatrix} \tag{25}$$

The aforementioned inverse kinematics of the Stewart platform, namely Eq. (1), is used to express the continuous observation model h_P:

$$\begin{aligned}
\underline{\boldsymbol{y}}_P(t) &= \mathrm{h}_P\left(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{o}}_P\right) \\
\{l_{i_m}\} &= \left\| \boldsymbol{c}_P^I + \boldsymbol{T}_{PI}^T\left[\underline{\boldsymbol{e}}_P\right] \boldsymbol{p}_i^P - \boldsymbol{b}_i^I \right\| + n_{l_i} \;\; \forall i = 1, \dots, 6
\end{aligned} \tag{26}$$

The Stewart platform's observation noise vector $\underline{\boldsymbol{o}}_P$ consists of the superimposed Gaussian white noise n_{l_i} $\forall i = 1, \dots, 6$.

As stated in Section IV-A, the visual pose tracking measures the HMD's cabin-fixed pose $\boldsymbol{c}_{HP_m}^P$, $\boldsymbol{e}_{H_m}^P$, i.e., the pose of the HMD w.r.t. $\mathbb{E}_P$ and also expressed in $\mathbb{E}_P$. In essence, the HMD's cabin-fixed pose is the result from subtracting the UGP's pose from the HMD's pose. This is due to the fact that both SRS and HMD state's poses are defined in the inertial reference frame $\mathbb{E}_I$, symbolically represented by the I superscript. In turn, this allows for the continuous observation model h_H to be expressed as:

$$\begin{aligned}
\underline{\boldsymbol{y}}_P(t) &= \begin{Bmatrix} \boldsymbol{c}_{HP_m}^P \\ \boldsymbol{e}_{H_m}^P \end{Bmatrix} = \mathrm{h}_H\left(\underline{\boldsymbol{x}}(t), \underline{\boldsymbol{o}}_H\right) \\
\begin{Bmatrix} \boldsymbol{c}_{HP}^P + \underline{\boldsymbol{n}}_{c_{HP}} \\ \boldsymbol{e}_H^P + \underline{\boldsymbol{n}}_{e_H^P} \end{Bmatrix} &= \begin{Bmatrix} T_{PI}\{\boldsymbol{c}_H^I - \boldsymbol{c}_P^I\} \\ \boldsymbol{e}_H^I \otimes \{\sim \boldsymbol{e}_P^I\} \end{Bmatrix} + \begin{Bmatrix} \underline{\boldsymbol{n}}_{c_{HP}} \\ \underline{\boldsymbol{n}}_{e_H^P} \end{Bmatrix}
\end{aligned} \tag{27}$$

In this equation, the white Gaussian noise vectors, $\underline{\boldsymbol{n}}_{c_{HP}}$ and $\underline{\boldsymbol{n}}_{e_H^P}$, part of $\underline{\boldsymbol{o}}_H$ corrupt the observations. Also, the transformation matrix $T_{PI}[\boldsymbol{e}_P^I]$ transposes the inertial based coordinates to the upper moving reference frame $\mathbb{E}_P$. Moreover, a deduction between quaternions is an 'added' counter rotation. The counter quaternion rotation is achieved by first taking the conjugate $\sim$ of the quaternion, i.e., switching the signs of the vector part, and subsequently 'adding' the conjugated quaternion $\sim \boldsymbol{e}_P^I$ via quaternion multiplication $\otimes$ [23] to the other quaternion:

$$\begin{aligned}
\boldsymbol{c}_{HP}^P &= \boldsymbol{e}_H^I \otimes \{\sim \boldsymbol{e}_P^I\} \\
&= \begin{bmatrix} e_{0_H} & -e_{x_H} & -e_{y_H} & -e_{z_H} \\ e_{x_H} & e_{0_H} & -e_{z_H} & e_{y_H} \\ e_{y_H} & e_{z_H} & e_{0_H} & -e_{x_H} \\ e_{z_H} & -e_{y_H} & e_{x_H} & e_{0_H} \end{bmatrix} \begin{Bmatrix} e_{0_P} \\ -e_{x_P} \\ -e_{y_P} \\ -e_{z_P} \end{Bmatrix}
\end{aligned} \tag{28}$$

Lastly, having defined both observation models, h_P and h_H, the observation noise vector $\underline{\boldsymbol{o}}$ is defined as:

$$\underline{\boldsymbol{o}} = \left(\underline{\boldsymbol{o}}_P^T \quad \underline{\boldsymbol{o}}_H^T\right)^T = \left(\underline{\boldsymbol{n}}_l^T \quad \underline{\boldsymbol{n}}_{c_{HP}}^T \quad \underline{\boldsymbol{n}}_{e_H^P}^T\right)^T \tag{29}$$

C. *Configuration 2 — IMU on both HMD and SRS*

Configuration 2 is similar to Configuration 1, but has an IMU attached to the SRS's upper moving frame, as depicted in Fig. 11. This is done in order to improve the prediction accuracy and potentially provide fast pose updates.

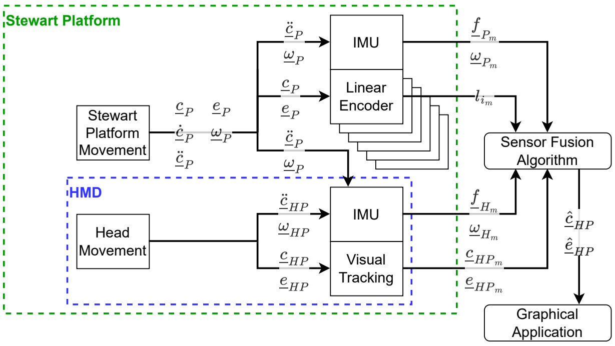


Fig. 11. Configuration 2: Similar to Configuration 1 with an IMU attached to the upper moving frame of the SRS.

Due to the fact that the IMU is attached to the moving upper reference frame $\mathbb{E}_P$, as is the case for the HMD, a similar input $\underline{\boldsymbol{u}}_P$, state $\underline{\boldsymbol{x}}_P$, and process noise vector $\underline{\boldsymbol{\nu}}_P$ can be defined as follows:

$$\underline{\boldsymbol{u}} = \begin{Bmatrix} \underline{\boldsymbol{u}}_P \\ \underline{\boldsymbol{u}}_H \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_{P_m}^P \\ \boldsymbol{\omega}_{P_m}^P \\ \underline{\boldsymbol{u}}_H \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} f_{x_{P_m}} & f_{y_{P_m}} & f_{z_{P_m}} \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} p_{P_m} & q_{P_m} & r_{P_m} \end{pmatrix} \{\mathbb{E}_P\} \\ \underline{\boldsymbol{u}}_H \end{Bmatrix} \tag{30}$$

Here, the input $\underline{\boldsymbol{u}}_H$ related to the HMD was left unaltered from Eq. (19).

$$\underline{\boldsymbol{x}} = \begin{Bmatrix} \underline{\boldsymbol{x}}_P \\ \underline{\boldsymbol{x}}_H \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{c}_P^I \\ \dot{\boldsymbol{c}}_P^I \\ \boldsymbol{e}_P^I \\ \boldsymbol{\lambda}_{f_P}^P \\ \boldsymbol{\lambda}_{\omega_P}^P \\ \underline{\boldsymbol{x}}_H \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} x_P & y_P & z_P \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \dot{x}_P & \dot{y}_P & \dot{z}_P \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} e_{0_P} & e_{x_P} & e_{y_P} & e_{z_P} \end{pmatrix} \begin{Bmatrix} 1 \\ \mathbb{E}_I \end{Bmatrix} \\ \begin{pmatrix} \lambda_{\ddot{x}_P} & \lambda_{\ddot{y}_P} & \lambda_{\ddot{z}_P} \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} \lambda_{p_P} & \lambda_{q_P} & \lambda_{r_P} \end{pmatrix} \{\mathbb{E}_P\} \\ \underline{\boldsymbol{x}}_H \end{Bmatrix} \tag{31}$$

In this equation, the HMD part of the state $\underline{\boldsymbol{x}}_H$ was defined in Eq. (23).

$$\underline{\boldsymbol{\nu}} = \begin{Bmatrix} \underline{\boldsymbol{\nu}}_P \\ \underline{\boldsymbol{\nu}}_H \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{n}_{f_P}^P \\ \boldsymbol{\mu}_{f_P}^P \\ \boldsymbol{n}_{\omega_P}^P \\ \boldsymbol{\mu}_{\omega_P}^P \\ \underline{\boldsymbol{\nu}}_H \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} n_{f_{x_P}} & n_{f_{y_P}} & n_{f_{z_P}} \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} \mu_{f_{x_P}} & \mu_{f_{y_P}} & \mu_{f_{z_P}} \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} n_{p_P} & n_{q_P} & n_{r_P} \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} \mu_{p_P} & \mu_{q_P} & \mu_{r_P} \end{pmatrix} \{\mathbb{E}_P\} \\ \underline{\boldsymbol{\nu}}_H \end{Bmatrix} \tag{32}$$

Also, the process noise vector $\underline{\boldsymbol{\nu}}_H$ related to the HMD is unmodified from Eq. (22).

The Stewart platform's continuous process model part f_P is very similar to its HMD counterpart:

$$\begin{gathered} \dot{\underline{\boldsymbol{x}}}(t) = \mathrm{f}(\underline{\boldsymbol{x}}(t), \underline{\boldsymbol{u}}(t), \underline{\boldsymbol{\nu}}) \\ \begin{Bmatrix} \dot{\underline{\boldsymbol{x}}}_P \\ \dot{\underline{\boldsymbol{x}}}_H \end{Bmatrix} = \begin{Bmatrix} \mathrm{f}_P(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{u}}_P(t), \underline{\boldsymbol{\nu}}_P) \\ \mathrm{f}_H(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{u}}_H(t), \underline{\boldsymbol{\nu}}_H) \end{Bmatrix} \\ \begin{Bmatrix} \dot{\boldsymbol{c}}_P^I \\ \ddot{\boldsymbol{c}}_P^I \\ \dot{\boldsymbol{e}}_P^I \\ \dot{\boldsymbol{\lambda}}_{f_P}^P \\ \dot{\boldsymbol{\lambda}}_{\omega_P}^P \\ \dot{\underline{\boldsymbol{x}}}_H \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{c}}_P^I \\ T_{PI}^T \left\{ \boldsymbol{f}_{P_m}^P - \boldsymbol{\lambda}_{f_P}^P - \boldsymbol{n}_{f_P}^P + \boldsymbol{a}_{\omega_P}^P \right\} + \boldsymbol{g} \\ \boldsymbol{\Omega}_{e_P} \left\{ \boldsymbol{\omega}_{P_m}^P - \boldsymbol{\lambda}_{\omega_P}^P - \boldsymbol{n}_{\omega_P}^P \right\} \\ \boldsymbol{\mu}_{f_P}^P \\ \boldsymbol{\mu}_{\omega_P}^P \\ \mathrm{f}_H(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{u}}_H(t), \underline{\boldsymbol{\nu}}_H) \end{Bmatrix} \end{gathered} \tag{33}$$

The only difference to Eq. (24) is the rotation-moment-linear-acceleration coupling component $\boldsymbol{a}_{\omega_P}^P$ resulting from the SRS's IMU not being located at the UGP [30], [31], and is defined as:

$$\boldsymbol{a}_{\omega_P}^P = \dot{\boldsymbol{\omega}}_P^P \times \boldsymbol{c}_{\mathrm{IMU}}^P + \boldsymbol{\omega}_P^P \times (\boldsymbol{\omega}_P^P \times \boldsymbol{c}_{\mathrm{IMU}}^P) \tag{34}$$

Here, × indicates the vector-cross-product, $\boldsymbol{c}_{\mathrm{IMU}}^P$ is the SRS's IMU location coordinates w.r.t. the UGP expressed in $\mathbb{E}_P$, and $\dot{\boldsymbol{\omega}}_P^P$ is the angular acceleration of $\mathbb{E}_P$ computed by differentiating the angular rate $\boldsymbol{\omega}_P^P$ and taking the mean of all sigma-points. In turn, the angular rate $\boldsymbol{\omega}_P^P$ of $\mathbb{E}_P$ in this configuration is defined as:

$$\boldsymbol{\omega}_P^P = \boldsymbol{\omega}_{P_m}^P - \boldsymbol{\lambda}_{\omega_P}^P - \boldsymbol{n}_{\omega_P}^P \tag{35}$$

No change is necessary to either observation model h_P or h_H as defined in Eq. (27) and (26).

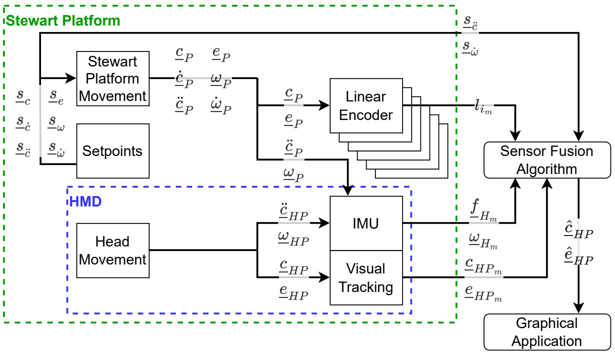

Stewart Platform
$\underline{s}_{\ddot{c}}$
$\underline{s}_{\dot{\omega}}$
Stewart Platform Movement
$\underline{c}_P$ $\underline{e}_P$
$\dot{\underline{c}}_P$ $\underline{\omega}_P$
$\ddot{\underline{c}}_P$ $\dot{\underline{\omega}}_P$
$\underline{s}_c$ $\underline{s}_e$
$\underline{s}_{\dot{c}}$ $\underline{s}_\omega$
$\underline{s}_{\ddot{c}}$ $\underline{s}_{\dot{\omega}}$
Setpoints
$\underline{c}_P$
$\underline{e}_P$
Linear Encoder
l_{i_m}
$\ddot{\underline{c}}_P$
$\underline{\omega}_P$
Sensor Fusion Algorithm
HMD
$\ddot{\underline{c}}_{HP}$
$\underline{\omega}_{HP}$
IMU
Head Movement
$\underline{c}_{HP}$
$\underline{e}_{HP}$
Visual Tracking
$\underline{f}_{H_m}$
$\underline{\omega}_{H_m}$
$\underline{c}_{HP_m}$
$\underline{e}_{HP_m}$
$\hat{\underline{c}}_{HP}$
$\hat{\underline{e}}_{HP}$
Graphical Application

Fig. 12. Configuration 3: similar to Configuration 2 with setpoints, rather than an IMU, used to help predict the Stewart platform's pose.

D. *Configuration 3 — IMU on HMD and Setpoints on SRS*

Contrary to the configuration with both sub-systems having an IMU (Configuration 2), Configuration 3 uses the setpoints $\underline{\boldsymbol{s}}$ of the SRS to predict the future pose of the UGP itself, indicated in Figure 12.

Also depicted in Fig. 12 is the full set of setpoints provided by the SRS, namely the position $\underline{\boldsymbol{s}}_c$, velocity $\underline{\boldsymbol{s}}_{\dot{c}}$, acceleration $\underline{\boldsymbol{s}}_{\ddot{c}}$, attitude $\underline{\boldsymbol{s}}_e$, angular rate $\underline{\boldsymbol{s}}_\omega$, and angular acceleration $\underline{\boldsymbol{s}}_{\dot{\omega}}$ [24]. However, only the acceleration $\underline{\boldsymbol{s}}_{\ddot{c}}$ and the angular acceleration $\underline{\boldsymbol{s}}_{\dot{\omega}}$ are used due to their higher order relationship to the observed pose.

This configuration investigates if software based setpoints of the SRS can be used to replace the IMU as a sensor and the possible impact it would have on the HMD's cabin-fixed pose. The added information from the setpoints should improve the UGP's pose estimation when compared to the configuration with only the HMD IMU (Configuration 1). Moreover, when compared to the configuration with both sub-systems having an IMU (Configuration 2), the question is how Gaussian and accurate the setpoints based process model is when compared to its IMU based counterpart.

In his article about the development of the SRS's control system, Koekebakker [25] states that a motion system with *"an inner loop feedback and feedback linearising control results in first order response of the system, from desired to actual accelerations."* Thus, the SRS's actual acceleration and angular acceleration can be approximated by the transfer function $\mathrm{G}(s)$ as:

$$\mathrm{G}(s) = \frac{\mathrm{Y}(s)}{\mathrm{U}(s)} = \frac{1}{\tau s + 1} = \frac{1/\tau}{s + 1/\tau} \tag{36}$$

Here, the input $\mathrm{U}(s)$ comprises of the setpoints describing the desired second derivative of the system's pose, the output $\mathrm{Y}(s)$ expresses the actual second derivative of the system's pose, both are dependent on the complex variable s, and a time lag τ that indicates the time for the system to reach 63 % of the final step input.

When Eq. 36 is applied in state-space form, it results in the following inertial acceleration $\ddot{\boldsymbol{c}}_P^I$ and body angular acceleration $\dot{\boldsymbol{\omega}}_P^P$:

$$\dddot{\boldsymbol{c}}_P^I = 1/\boldsymbol{\tau}_{\ddot{c}}^I \cdot (\boldsymbol{s}_{\ddot{c}}^I - \ddot{\boldsymbol{c}}_P^I) \tag{37}$$

$$\ddot{\boldsymbol{\omega}}_P^P = 1/\boldsymbol{\tau}_{\dot{\omega}}^P \cdot (\boldsymbol{s}_{\dot{\omega}}^P - \dot{\boldsymbol{\omega}}_P^P) \tag{38}$$

Here, $\cdot$ symbolizes a coefficient-wise operation.

Also, Eq. (37) and (38) show that the Stewart platform's state $\underline{\boldsymbol{x}}_P$ needs to include up to the second derivative of its pose:

$$\underline{\boldsymbol{x}} = \begin{Bmatrix} \underline{\boldsymbol{x}}_P \\ \underline{\boldsymbol{x}}_H \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{c}_P^I \\ \dot{\boldsymbol{c}}_P^I \\ \ddot{\boldsymbol{c}}_P^I \\ \boldsymbol{e}_P^I \\ \boldsymbol{\omega}_P^P \\ \dot{\boldsymbol{\omega}}_P^P \\ \boldsymbol{\tau}_{\ddot{c}_P}^I \\ \boldsymbol{\tau}_{\dot{\omega}_P}^P \\ \underline{\boldsymbol{x}}_H \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} x_P & y_P & z_P \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \dot{x}_P & \dot{y}_P & \dot{z}_P \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \ddot{x}_P & \ddot{y}_P & \ddot{z}_P \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} e_{0_P} & e_{x_P} & e_{y_P} & e_{z_P} \end{pmatrix} \begin{Bmatrix} 1 \\ \mathbb{E}_I \end{Bmatrix} \\ \begin{pmatrix} p_P & q_P & r_P \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} \dot{p}_P & \dot{q}_P & \dot{r}_P \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} \tau_{\ddot{x}} & \tau_{\ddot{y}} & \tau_{\ddot{z}} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \tau_{\dot{p}} & \tau_{\dot{q}} & \tau_{\dot{r}} \end{pmatrix} \{\mathbb{E}_P\} \\ \underline{\boldsymbol{x}}_H \end{Bmatrix} \tag{39}$$

Here, the HMD's state $\underline{\boldsymbol{x}}_H$ is defined as in Eq. (23). Moreover, the Stewart platform's input $\underline{\boldsymbol{u}}_P$ should include the setpoints $\boldsymbol{s}_{\ddot{c}}^I$ and $\boldsymbol{s}_{\dot{\omega}}^P$:

$$\underline{\boldsymbol{u}} = \begin{Bmatrix} \underline{\boldsymbol{u}}_P \\ \underline{\boldsymbol{u}}_H \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{s}_{\ddot{c}}^I \\ \boldsymbol{s}_{\dot{\omega}}^P \\ \underline{\boldsymbol{u}}_H \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} s_{\ddot{x}} & s_{\ddot{y}} & s_{\ddot{z}} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} s_{\dot{p}} & s_{\dot{q}} & s_{\dot{r}} \end{pmatrix} \{\mathbb{E}_P\} \\ \underline{\boldsymbol{u}}_H \end{Bmatrix} \tag{40}$$

In order for the UKF to estimate the time-lag $\boldsymbol{\tau}_{\ddot{c}}^I$ and $\boldsymbol{\tau}_{\dot{\omega}}^P$, influencing the setpoints $\boldsymbol{s}_{\ddot{c}}$ and $\boldsymbol{s}_{\dot{\omega}}$ respectively, a variance needs to be imposed on said lag. This is achieved by defining $\boldsymbol{\eta}_{\tau_{\ddot{c}}}^I$, $\boldsymbol{\eta}_{\tau_{\dot{\omega}}}^P$ as the change of the lag over time $\dot{\boldsymbol{\tau}}_{\ddot{c}}^I$, $\dot{\boldsymbol{\tau}}_{\dot{\omega}}^P$ and is assumed to be Gaussian. $\boldsymbol{\eta}_{\tau_{\ddot{c}}}^I$ and $\boldsymbol{\eta}_{\tau_{\dot{\omega}}}^P$ fullfil a similar function as the noise variables $\boldsymbol{\mu}$ to the bias $\boldsymbol{\lambda}$ of an IMU, whereby the lag's variance allows the UKF to change the lag $\boldsymbol{\eta}$ based on the measurements of the rest of the system.

In principle the Stewart platform's noise vector $\underline{\boldsymbol{\nu}}$ is complete. However, if the resulting process model would have been applied to the UKF, the behavior would not have been optimal because the lag's deviation has to be small, i.e., approximately 10^{-3}, so as not to let the UKF sample sub-zero lags causing unstable mathematical behavior of the UKF. To account for this error in the process model, additional variance is injected into the UKF by introducing two additional Gaussian noise sources, one influencing the inertial acceleration $\boldsymbol{\nu}_{\ddot{c}_P}^I$ of the Stewart platform, and the other influencing the body angular acceleration $\boldsymbol{\nu}_{\dot{\omega}_P}^P$. This results in the definition of the process noise vector $\underline{\boldsymbol{\nu}}$ as:

$$\underline{\boldsymbol{\nu}} = \begin{Bmatrix} \underline{\boldsymbol{\nu}}_P \\ \underline{\boldsymbol{\nu}}_H \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\eta}_{\tau_{\ddot{c}}}^I \\ \boldsymbol{\nu}_{\ddot{c}_P}^I \\ \boldsymbol{\eta}_{\tau_{\dot{\omega}}}^P \\ \boldsymbol{\nu}_{\dot{\omega}_P}^P \\ \underline{\boldsymbol{\nu}}_H \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} \eta_{\tau_{\ddot{x}}} & \eta_{\tau_{\ddot{y}}} & \eta_{\tau_{\ddot{z}}} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \nu_{\ddot{x}_P} & \nu_{\ddot{y}_P} & \nu_{\ddot{z}_P} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \eta_{\tau_{\dot{p}}} & \eta_{\tau_{\dot{q}}} & \eta_{\tau_{\dot{r}}} \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} \nu_{\dot{p}_P} & \nu_{\dot{q}_P} & \nu_{\dot{r}_P} \end{pmatrix} \{\mathbb{E}_P\} \\ \underline{\boldsymbol{\nu}}_H \end{Bmatrix} \tag{41}$$

Having defined the input vector $\underline{\boldsymbol{u}}$, the state vector $\underline{\boldsymbol{x}}$, the process noise vector $\underline{\boldsymbol{x}}$, and having presented the setpoints influence on the state in Eq. (37) and (38), the continuous process model f is defined as follows:

$$\begin{gathered} \dot{\underline{\boldsymbol{x}}}(t) = \mathrm{f}(\underline{\boldsymbol{x}}(t), \underline{\boldsymbol{u}}(t), \underline{\boldsymbol{\nu}}) \\ \begin{Bmatrix} \dot{\underline{\boldsymbol{x}}}_P \\ \dot{\underline{\boldsymbol{x}}}_H \end{Bmatrix} = \begin{Bmatrix} \mathrm{f}_P(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{u}}_P(t), \underline{\boldsymbol{\nu}}_P) \\ \mathrm{f}_H(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{u}}_H(t), \underline{\boldsymbol{\nu}}_H) \end{Bmatrix} \\ \begin{Bmatrix} \dot{\boldsymbol{c}}_P^I \\ \ddot{\boldsymbol{c}}_P^I \\ \dddot{\boldsymbol{c}}_P^I \\ \dot{\boldsymbol{e}}_P^I \\ \dot{\boldsymbol{\omega}}_P^P \\ \ddot{\boldsymbol{\omega}}_P^P \\ \dot{\boldsymbol{\tau}}_{\ddot{c}}^I \\ \dot{\boldsymbol{\tau}}_{\dot{\omega}_P}^P \\ \dot{\underline{\boldsymbol{x}}}_H \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{c}}_P^I \\ \ddot{\boldsymbol{c}}_P^I + \boldsymbol{\eta}_{\tau_{\ddot{c}}}^I \\ 1/\boldsymbol{\tau}_{\ddot{c}}^I \cdot (\boldsymbol{s}_{\ddot{c}}^I - \ddot{\boldsymbol{c}}_P^I) \\ \boldsymbol{\Omega}_{e_P} \boldsymbol{\omega}_P^P \\ \dot{\boldsymbol{\omega}}_P^P + \boldsymbol{\eta}_{\tau_{\dot{\omega}}}^P \\ 1/\boldsymbol{\tau}_{\dot{\omega}}^P \cdot (\boldsymbol{s}_{\dot{\omega}}^P - \dot{\boldsymbol{\omega}}_P^P) \\ \boldsymbol{\nu}_{\ddot{c}_P}^I \\ \boldsymbol{\nu}_{\dot{\omega}_P}^P \\ \mathrm{f}_H(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{u}}_H(t), \underline{\boldsymbol{\nu}}_H) \end{Bmatrix} \end{gathered} \tag{42}$$

Again, no change is necessary to the observation models h_P and h_H as defined in Eq. (27) and (26).

E. Discretization

The UKF as presented in Section III, works with discrete process F and observation models H. The discretization of the continuous models introduced in this section is performed in a number of ways, depending on the type of model.

1) Forward Euler Method: In order to discretize the previously presented continuous process models f the forward Euler method, a first order numerical integration method, was used [30], [31]:

$$\begin{array}{l} \dot{\underline{\boldsymbol{x}}}(t) = \mathrm{f}(\underline{\boldsymbol{x}}(t), \underline{\boldsymbol{u}}(t), \underline{\boldsymbol{\nu}}) \\ \qquad \Big\downarrow \text{Discrete Integration Method} \\ \underline{\boldsymbol{x}}_{k+1} = \mathrm{F}(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{u}}_k, \underline{\boldsymbol{\nu}}) \xrightarrow[\text{Method}]{\text{Forward}} \underline{\boldsymbol{x}}_k + \dot{\underline{\boldsymbol{x}}}_k \Delta t + O\left(\Delta t^2\right) \end{array} \tag{43}$$

The forward Euler method truncates the Taylor series after the first order [40], leading to the error $O\left(\Delta t^2\right)$ in Eq. (43). This error is known as the local truncation error, but will here be referred to as the integration error. Since the system's motion is discretized by the sensors and subsequently propagated through the imperfect integration method, this error is non-Gaussian, i.e., colored by the motion itself.

Furthermore, this integration error is present in the discrete prediction model F and should not be disregarded, as such a system would trust the predicted state of the process model F too much. This effect can be mitigated by increasing the appropriate noise values, as described in Section VI.

2) Quaternion Integration: The quaternion attitudes are integrated following the analytical method discussed in [31], [41] to increase attitude prediction. This is due to the fact that increasing the attitude prediction accuracy not only benefits the final estimated attitude, but also propagates through the various IMU measurements and positions by transforming them into other reference frames and thus was deemed worth the additional computational workload.

3) Discrete Observation Models: The discretization of the continuous observation models h does not require any method and these models are used as is in a discrete fashion:

$$\underline{\boldsymbol{y}}(t) = \mathrm{h}\left(\underline{\boldsymbol{x}}(t), \underline{\boldsymbol{o}}\right) \rightarrow \underline{\boldsymbol{y}}_k = \mathrm{H}\left(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{o}}\right) \tag{44}$$

V. Method

A. Data Gathering

In the enclosed cabin of the SRS, depicted in Fig. 6, the TrackIR 5 sensor was installed along with the Pimax 8k-x. The Pimax 8k-x was strapped to the left seat's headrest, as can be seen in Fig. 13. Next, the SRS would move along a preprogrammed set of setpoints. The real sensor data was saved via eCAL [42], version 5.11, a middleware enabling interprocess communication.

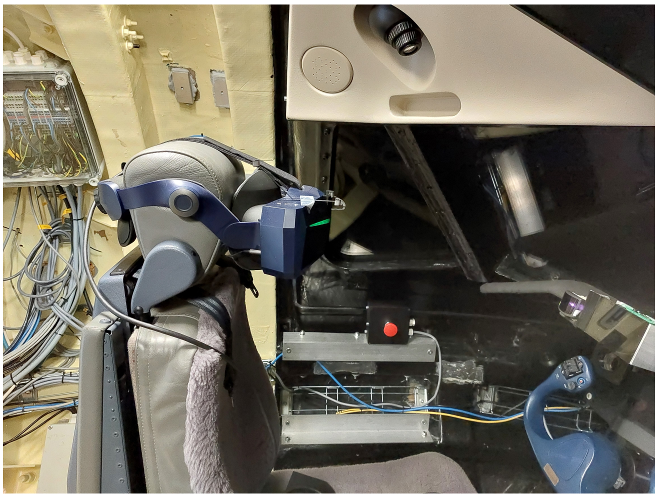


Fig. 13. The Pimax 8k-x strapped the the left seat's headrest.

The resulting recordings were loaded by a Python (version 3.11) [43] script, using Numpy (version 1.26) [44] for its efficient array computing abilities. Subsequently, these recordings were stepped through time-wise, taking into account the data's recorded timestamps, and supplied to the presented algorithm. This approach resulted in a deterministic implementation, meaning that given the same parameters, algorithm, and data, the exact same results were attained. This also meant that the innovations of the different configurations are comparable for the same motion profile due to the fact that the input into the observation model is also deterministic.

B. Performance Metrics

The accuracy of the estimated state $\underline{\hat{\boldsymbol{x}}}$, and in this case particular the estimated cabin-fixed pose ($\underline{\hat{\boldsymbol{c}}}_{HP}$, $\underline{\hat{\boldsymbol{e}}}_{HP}$) as defined in Eq. (13) and Eq. (14), are the true performance indicators. However, the accuracy of the estimated state is not observable as the true state is not known. For this reason, the performance and quality of the algorithm is ascertained with the innovation $\underline{\boldsymbol{\epsilon}}$ and its covariance $\boldsymbol{P}_{\hat{y}}$ [45].

1) Innovation $\underline{\boldsymbol{\epsilon}}$: is the difference between the actual observation $\underline{\boldsymbol{y}}$ and the estimated observation $\underline{\hat{\boldsymbol{y}}}$, and as such is used as a variable in a KF. Normally, the innovation $\underline{\boldsymbol{\epsilon}}$, without comparing it to its covariance $\boldsymbol{P}_{\hat{y}}$, is mostly meaningless. However, due to the fact that this Python implementation is deterministic, the resulting innovation of each configuration can be compared to one another to an extent. The extent being that a part of the innovation $\underline{\boldsymbol{\epsilon}}$ is due to the observation sensor itself.

2) Estimated Observation Covariance $\boldsymbol{P}_{\hat{y}}$: an UKF variable, the diagonal of said matrix $\boldsymbol{P}_{\hat{y}}$ corresponds to the expected auto-variance $\underline{\boldsymbol{\sigma}}_{\epsilon}^2$ of the innovation vector $\underline{\boldsymbol{\epsilon}}$. Thus, in theory, 95 % of the innovation should lay between $2 \cdot \sqrt{\underline{\boldsymbol{\sigma}}_{\epsilon}^2}$ [46], i.e., sigma-bounds 2σ. This means that the level of appropriate tuning of the UKF's noise parameters can be ascertained, to an extent, by plotting the innovation alongside its sigma-bounds 2σ.

Reversely, given a properly tuned UKF and no bias in the sensors responsible for the UKF's update, the sigma-bounds give a theoretical performance indicator.

3) State Prediction Rate: the graphics pipeline of a typical VR/AR system requests the estimated state of the HMD at a high and asynchronous rate [47]. A performance analysis solely based on the innovation $\underline{\boldsymbol{\epsilon}}$ and its covariance $\boldsymbol{P}_{\hat{y}}$ fails to capture the potential benefits of a higher state prediction rate for such a high and asynchronous polling rate system. For this reason, it is beneficial for the algorithm to have a high prediction rate, especially if the update rate is relatively low.

For example, when comparing two systems where one has a 'tighter' estimated observation covariance $\boldsymbol{P}_{\hat{y}}$, and the other has a higher state prediction rate, the latter could, in certain conditions, be more accurate from a state estimation perspective. If the user's system would poll the estimated state at a high enough rate, the former system would likely have to give the same state multiple times, while the latter would be able to provide a new state estimation based on the latest input data albeit with larger estimation errors per estimate.

VI. Parameters

In order for the real measurements to be applied to the UKF, certain parameters need to be established first.

The UKF needs its scaling parameters (α, β, κ), an initial estimated stochastic state $\underline{\hat{\boldsymbol{x}}}_0$, and corresponding covariance $\boldsymbol{P}_{\hat{x}_0}$. Moreover, the process $\boldsymbol{Q}$ and observation noise matrices $\boldsymbol{R}$ need to be established as well.

Starting with the process noise matrix $\boldsymbol{Q}$, the values for the HMD are presented in Tab. I. Here, the velocity $\underline{\boldsymbol{\sigma}}_{f_H}$ and angle random walk $\underline{\boldsymbol{\sigma}}_{\omega_H}$ are based on static measurements results, which were subsequently scaled up by 10 % as a safety margin. For the HMD's velocity $\underline{\boldsymbol{\sigma}}_{f_H}$ and angle random walk $\underline{\boldsymbol{\sigma}}_{\omega_H}$, no additional noise is needed to reduce the impact of the integration error because the IMU's measurements are published and used at 600 Hz; and the HMD's IMU statically noise parameters are large when compared to the resulting integration error.

The acceleration $\underline{\boldsymbol{\sigma}}_{\mu_{f_H}}$ and rate random walk $\underline{\boldsymbol{\sigma}}_{\mu_{\omega_H}}$ were initially computed via Overlapping Allan Deviation [1], [38], [39], [48], but were found too large to be effective at estimating the bias $\underline{\boldsymbol{\lambda}}$ of the respective sensors. Rather than solely estimating the bias $\underline{\boldsymbol{\lambda}}$, the algorithm was using said bias to try to partially solve the aforementioned integration error. In order to improve this behavior, the found values were roughly scaled down by a factor of ten.

TABLE I
HMD's IMU noise parameters

Description	Var.	Value	Unit	
$\underline{\boldsymbol{n}}_{f_H} \approx \aleph\big(0, \underline{\boldsymbol{\sigma}}_{f_H}\big)$ Accelerometer's Velocity Random Walk	$\sigma_{f_{x_H}}$	8.26	$\times 10^{-2}$ m/s^2	
	$\sigma_{f_{y_H}}$	5.85		
	$\sigma_{f_{z_H}}$	5.60		
$\underline{\boldsymbol{\mu}}_{f_H} \approx \aleph\Big(0, \underline{\boldsymbol{\sigma}}_{\mu_{f_H}}\Big)$ Accelerometer's Acceleration Random Walk	$\sigma_{\mu_{f_{\square_H}}}$	1.00	$\times 10^{-4}$ m/s^3	$\forall\square = x, y, z$
$\underline{\boldsymbol{n}}_{\omega_H} \approx \aleph\big(0, \underline{\boldsymbol{\sigma}}_{\omega_H}\big)$ Gyroscope's Angle Random Walk	σ_{p_H}	4.64	$\times 10^{-3}$ rad/s	
	σ_{q_H}	3.96		
	σ_{r_H}	3.30		
$\underline{\boldsymbol{\mu}}_{\omega_H} \approx \aleph\Big(0, \underline{\boldsymbol{\sigma}}_{\mu_{\omega_H}}\Big)$ Gyroscope's Rate Random Walk	$\sigma_{\mu_{\square_H}}$	1.00	$\times 10^{-5}$ rad/s^2	$\forall\square = p, q, r$

The process noise values related to the SRS are shown in Tab. II. The values for Configuration 1, where only the HMD has an IMU, are based on the expectation on the upper bounds of the acceleration and angular acceleration to be encountered in the motion of the SRS. If these estimated values are set too high, the estimation accuracy of the UKF decreases due to the UKF's increased state's covariance $\boldsymbol{P}_{\hat{x}}$ elements, causing the sampled sigma-points to potentially miss local non-linearities and trust the observation sensors potentially too much. Conversely, estimating these noise parameters too low causes the prediction to lag behind due to the small state's covariance $\boldsymbol{P}_{\hat{x}}$, causing the system to seem reluctant to change its state based upon observations.

Static measurements were slightly lower than the manufacturer's manual [49], with the SRS's accelerometer and gyroscope to have a deviation of ≈1×10^{-2} m/s^2 and ≈1.7×10^{-4} rad/s respectively. In this case, the aforementioned integration error, partly due to the relatively low publishing frequency of 100 Hz, causes both the SRS's velocity $\underline{\boldsymbol{\sigma}}_{f_P}$ and angle random walk $\underline{\boldsymbol{\sigma}}_{\omega_P}$ to be scaled up by a factor of four. This behavior was also evident in the work done by Miletović, where the whole process noise covariance matrix $\boldsymbol{Q}$ was ultimately scaled up by a factor of 1000 [30]. The reason for this high factor was probably due to the combination of a different motion profile and that the accelerometer used in his work was ten times more accurate (in terms of standard deviation) than the one presented here.

The noise values for the SRS's acceleration $\underline{\boldsymbol{\sigma}}_{\mu_{f_P}}$ and rate random walk $\underline{\boldsymbol{\sigma}}_{\mu_{\omega_P}}$ follow a similar trend to the HMD one, where they were scaled down to reduce the IMU's bias fluctuating due to the integration error.

Finally, the setpoint lag deviation values are solely based on trial and error. If they are set too small, the UKF has problems converging on the lag, too large and the UKF could potentially estimate a negative lag, causing instability. Moreover, the acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration deviation $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ have different values from the configuration in which only the HMD has an IMU (Configuration 1) as the system can relay on the lag for the prediction model.

The observation deviation values are presented in Tab. III. The values for the linear encoders are taken from the manufacturer's manual [21] and verified with static measurements.

The TrackIR 5 was found to have a positional and attitudinal deviation of ≈3×10^{-5} m and ≈2×10^{-4} rad respectively during static tests. However, this sensor was found to also present a time-lag of approximately one second. This was confirmed by integrating the angular rate of the headset and comparing this to the attitude provided by the TrackIR 5 sensor measurements. Even though the data gathered through the TrackIR was with the HMD strapped to the headrest and thus mostly static w.r.t. to the TrackIR sensor, the free play in the headrest was enough to introduce errors into the observation due to the firm SRS motion profile. To reduce these errors in pose, the TrackIR deviations were scaled up by a factor of 50 and 25 to reduce the impact on the state estimation.

The estimated initial stochastic state $\underline{\hat{\boldsymbol{x}}}_0$ of Configuration 1, where only the HMD has an IMU, is set in Eq. (45). The initial position of the SRS was set to its neutral position and the HMD's position was set to that of the left seat's headrest. Both quaternion attitudes were set to

TABLE II
SRS PROCESS NOISE PARAMETERS

Description	Var.	Value	Unit	
Configuration 1:				
$\underline{\boldsymbol{\nu}}_{\ddot{c}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\ddot{c}_P}\right)$ Acceleration Deviation	$\sigma_{\square_P}$	1.5	m/s^2	$\forall\square = \ddot{x}, \ddot{y}, \ddot{z}$
$\underline{\boldsymbol{\nu}}_{\dot{\omega}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}\right)$ Angular Acceleration Deviation	$\sigma_{\square_P}$	1.5	rad/s^2	$\forall\square = \dot{p}, \dot{q}, \dot{r}$
Configuration 2:				
$\underline{\boldsymbol{n}}_{f_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{f_P}\right)$ Accelerometer's Velocity Random Walk	$\sigma_{f_{\square_P}}$	$4.0{\times}10^{-2}$	m/s^2	$\forall\square = x, y, z$
$\underline{\boldsymbol{\mu}}_{f_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\mu_{f_P}}\right)$ Accelerometer's Acceleration Random Walk	$\sigma_{\mu_{f_{\square_P}}}$	$1.0{\times}10^{-4}$	m/s^3	
$\underline{\boldsymbol{n}}_{\omega_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\omega_P}\right)$ Gyroscope's Angle Random Walk	$\sigma_{\square_P}$	$8.0{\times}10^{-3}$	rad/s	$\forall\square = p, q, r$
$\underline{\boldsymbol{\mu}}_{\omega_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\mu_{\omega_P}}\right)$ Gyroscope's Rate Random Walk	$\sigma_{\mu_{\square_P}}$	$1.0{\times}10^{-5}$	rad/s^2	
Configuration 3:				
$\underline{\boldsymbol{\eta}}_{\tau_{\ddot{c}}} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\tau_{\ddot{c}_P}}\right)$ Translational Lag Deviation	$\sigma_{\tau_\square}$	$1.0{\times}10^{-3}$	s	$\forall\square = \ddot{x}, \ddot{y}, \ddot{z}$
$\underline{\boldsymbol{\nu}}_{\ddot{c}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\ddot{c}_P}\right)$ Acceleration Deviation	$\sigma_{\square_P}$	$5.0{\times}10^{-1}$	m/s^2	
$\underline{\boldsymbol{\eta}}_{\tau_{\dot{\omega}}} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\tau_{\dot{\omega}_P}}\right)$ Rotational Lag Deviation	$\sigma_{\tau_\square}$	$1.0{\times}10^{-3}$	s	$\forall\square = \dot{p}, \dot{q}, \dot{r}$
$\underline{\boldsymbol{\nu}}_{\dot{\omega}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}\right)$ Angular Acceleration Deviation	$\sigma_{\square_P}$	$5.0{\times}10^{-1}$	rad/s^2	

TABLE III
OBSERVATION SENSORS NOISE PARAMETERS

Description	Var.	Value	Unit	
$n_{l_i} \approx \aleph\left(0, \sigma_{l_i}\right),\ \sigma_{l_i} \in \underline{\boldsymbol{\sigma}}_l$ Absolute Linear Encoders Deviation	σ_{l_i}	$5.0{\times}10^{-6}$	m	$\forall i = 1, \dots, 6$
TrackIR 5 Deviation: Positional: $\underline{\boldsymbol{n}}_{c_{HP}} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{c_{HP}}\right)$	$\sigma_{\square_{HP}}$	$1.5{\times}10^{-3}$	m	$\forall\square = x, y, z$
Attitudinal: $\underline{\boldsymbol{n}}_{e^P_H} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{e^P_H}\right)$	$\sigma_{\square_{HP}}$	$5.0{\times}10^{-3}$	rad	$\forall\square = \phi, \theta, \psi$

an identity rotation, and all other values were set to zero.

$$\hat{\underline{\boldsymbol{x}}}_0 = E\left|\underline{\boldsymbol{x}}_0\right| = E\left|\begin{pmatrix}\underline{\boldsymbol{x}}^T_{P_0} & \underline{\boldsymbol{x}}^T_{H_0}\end{pmatrix}^T\right| \tag{45}$$

$$\begin{Bmatrix}\hat{\underline{\boldsymbol{x}}}_{P_0} \\ \hat{\underline{\boldsymbol{x}}}_{H_0}\end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix}\mathbf{0}_{2\times} & -2.39 & \mathbf{0}_{3\times} & 1 & \mathbf{0}_{6\times}\end{pmatrix}^T \\ \begin{pmatrix}0 & -0.55 & -3.5975 & \mathbf{0}_{3\times} & 1 & \mathbf{0}_{9\times}\end{pmatrix}^T \end{Bmatrix} \tag{46}$$

The covariance of the estimated initial stochastic state $\boldsymbol{P}_{\hat{x}_0}$ of Configuration 1 is shown in Eq. (47). The deviation in the expected initial stochastic state are all set independent to one another.

$$\boldsymbol{P}_{\hat{x}_0} = E\left|\left\{\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0\right\}\left\{\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0\right\}^T\right| \tag{47}$$

$$\begin{bmatrix}\boldsymbol{P}_{\hat{x}_{P_0}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{P}_{\hat{x}_{H_0}}\end{bmatrix} = \begin{bmatrix} \operatorname{diag}(\mathbf{1}_{6\times}\ \mathbf{0.44}_{3\times}\ \mathbf{1}_{3\times})^2 & \mathbf{0}_{12\times15} \\ \mathbf{0}_{15\times12} & \operatorname{diag}(\mathbf{1}_{6\times}\ \mathbf{0.52}_{3\times}\ \mathbf{0.1}_{3\times}\ \mathbf{0.05}_{3\times})^2 \end{bmatrix} \tag{48}$$

Taking the deviation values presented in Tab. I, II, and III into account, the process $\boldsymbol{Q}$ and observation noise matrices $\boldsymbol{R}$ can be constructed as follows:

$$\begin{aligned}\boldsymbol{Q} &= \operatorname{diag}\begin{pmatrix}\underline{\boldsymbol{\sigma}}_P & \underline{\boldsymbol{\sigma}}_H\end{pmatrix}^2 \qquad (49) \\ &= \operatorname{diag}\begin{pmatrix}\underline{\boldsymbol{\sigma}}_{\ddot{c}_P} & \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P} & \underline{\boldsymbol{\sigma}}_{f_H} & \underline{\boldsymbol{\sigma}}_{\mu_{f_H}} & \underline{\boldsymbol{\sigma}}_{\omega_H} & \underline{\boldsymbol{\sigma}}_{\mu_{\omega_H}}\end{pmatrix}^2\end{aligned}$$

$$\boldsymbol{R} = \operatorname{diag}\begin{pmatrix}\underline{\boldsymbol{\sigma}}_l & \underline{\boldsymbol{\sigma}}_{c_{HP}} & \underline{\boldsymbol{\sigma}}_{e^P_H}\end{pmatrix}^2 \tag{50}$$

For Configuration 2, where both the SRS and HMD have an IMU, only the SRS's sub-system changes, thus resulting in:

$$\hat{\underline{\boldsymbol{x}}}_{P_0} = \begin{pmatrix}\mathbf{0}_{2\times} & -2.39 & \mathbf{0}_{3\times} & 1 & \mathbf{0}_{9\times}\end{pmatrix}^T \tag{51}$$

$$\boldsymbol{P}_{\hat{x}_{P_0}} = \operatorname{diag}\begin{pmatrix}\mathbf{1}_{6\times} & \mathbf{0.44}_{3\times} & \mathbf{0.1}_{3\times} & \mathbf{0.05}_{3\times}\end{pmatrix}^2 \tag{52}$$

$$\boldsymbol{Q} = \operatorname{diag}\begin{pmatrix}\underline{\boldsymbol{\sigma}}_{f_P} & \underline{\boldsymbol{\sigma}}_{\mu_{f_P}} & \underline{\boldsymbol{\sigma}}_{\omega_P} & \underline{\boldsymbol{\sigma}}_{\mu_{\omega_P}} & \underline{\boldsymbol{\sigma}}_H\end{pmatrix}^2 \tag{53}$$

For Configuration 3, where setpoints are used rather than an IMU for the SRS, the setpoint related values are printed. Here, attention should be paid when selecting a lag starting value and its corresponding covariance. Choosing an initial lag value close to zero with a large covariance could cause a sigma-point to be sampled below zero causing unstable behavior. Here, it was opted to choose the initial lag variance too small to increase robustness at the cost of convergence duration of said lag.

$$\hat{\underline{\boldsymbol{x}}}_{P_0} = \begin{pmatrix}\mathbf{0}_{2\times} & -2.39 & \mathbf{0}_{6\times} & 1 & \mathbf{0}_{9\times} & \mathbf{0.03}_{6\times}\end{pmatrix}^T \tag{54}$$

$$\boldsymbol{P}_{\hat{x}_{P_0}} = \operatorname{diag}\begin{pmatrix}\mathbf{1}_{9\times} & \mathbf{0.44}_{3\times} & \mathbf{1}_{6\times} & \mathbf{10^{-5}}_{6\times}\end{pmatrix}^2 \tag{55}$$

$$\boldsymbol{Q} = \operatorname{diag}\begin{pmatrix}\underline{\boldsymbol{\sigma}}_{\tau_{\ddot{c}_P}} & \underline{\boldsymbol{\sigma}}_{\ddot{c}_P} & \underline{\boldsymbol{\sigma}}_{\tau_{\dot{\omega}_P}} & \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P} & \underline{\boldsymbol{\sigma}}_H\end{pmatrix}^2 \tag{56}$$

Finally, the sigma parameters (α, β, κ) were chosen as follows:

$$\alpha = 0.01, \quad \beta = 2, \quad \kappa = 100 \tag{57}$$

κ is chosen larger than the recommended value [50] to increase mathematical stability in case of Configuration 3 with the small initial lag variance. These values were kept constant for all configurations.

VII. Motion Profile

The motion of the SRS, as pointed out in [30], should be large enough to make sure to explore the nonlinearity of the SRS's stewart platform [51]. The level of nonlinearity impacts the UKF Gaussian approximation in both the prediction as well in the correction phase.

The motion profile consists of the summation of sinusoids, with a hyperbolic tangent based fade-in of ten seconds. The frequency and amplitude of each sinusoid per axis w.r.t. the inertial reference frame $\mathbb{E}_I$ is given in Tab. IV. All initial states are set to zero, except for the z-axis of the position of the UGP which is set to $-2.39\,\text{m}$, the neutral position of the SRS. During the online data gathering the SRS was made to follow the resulting motion profile.

TABLE IV
Amplitudes and Frequencies of the sinusoidal motion profile of the SRS w.r.t. the inertial reference frame $\mathbb{E}_I$.

	Translation		Rotation	
SRS:	Amplitude	Frequency	Amplitude	Frequency
x:	−0.1 m	0.1 Hz	−4.01 °	0.1 Hz
	0.1 m	0.25 Hz	4.01 °	0.25Hz
	0.0123 m	0.65 Hz	0.705 °	0.65 Hz
	0.016 m	0.85 Hz	0.917 °	0.85 Hz
	0.002 m	2 Hz	0.115 °	2 Hz
y:	0.1 m	−0.1 Hz	4.01 °	0.1 Hz
	0.1 m	−0.25 Hz	4.01 °	0.25Hz
	0.0123 m	0.65 Hz	0.705 °	0.65 Hz
	0.016 m	0.85 Hz	0.917 °	0.85 Hz
	0.002 m	2 Hz	0.115 °	2 Hz
z:	0.03 m	0.5 Hz	2.865 °	0.5 Hz
	0.002 m	2 Hz	0.115 °	2 Hz

VIII. Results

The three proposed configurations are implemented and validated by feeding each configuration's UKF the data gathered from running the motion profile on the SRS.

Due to their representativeness, as well as for brevity, only the innovations of the x-axis $\epsilon_{x_{HP}}$ and roll $\epsilon_{\phi_{HP}}$ are depicted in Fig. 14, alongside their sigma-bounds, represented by $\pm 2 \cdot \sqrt{\boldsymbol{P}_{\hat{y}_{x_{HP}}}}$. Fig. 14 depicts how the mismatched sensor information between the HMD's IMU and the lagged visual pose tracking leads to the resulting innovation being heavily colored by the motion profile.

The mean of the second half of these sigma-bounds are also printed in Tab. V. The left side of said table shows the results with the noise parameters presented as per Tab. III. The right side of said table shows the results of the original statically measured noise parameters for the TrackIR, i.e., $\approx 3\times10^{-5}$ m and $\approx 2\times10^{-4}$ rad for the position and attitude respectively. Thus, it shows the results for a potential non-lagged visual tracker scenario.

In Fig. 15, innovations of selected linear encoders are plotted against their 95 % sigma-bounds corresponding to the elements on the diagonal of the estimated observation covariance $\boldsymbol{P}_{\hat{y}}$.

TABLE V
The TrackIR 5 estimated sigma-bounds based on the average estimated observation covariance $\tilde{\boldsymbol{P}}_{\hat{y}}$ of the x-axis and roll attitude ϕ_{HP} per configuration.

	Corrected Noise Parameters		Original Noise Parameters	
Config	$2 \cdot \sqrt{\boldsymbol{P}_{\hat{y}_{x_{HP}}}}$	$2 \cdot \sqrt{\boldsymbol{P}_{\hat{y}_{\phi_{HP}}}}$	$2 \cdot \sqrt{\boldsymbol{P}_{\hat{y}_{x_{HP}}}}$	$2 \cdot \sqrt{\boldsymbol{P}_{\hat{y}_{\phi_{HP}}}}$
0	3.00×10^{-3} m	5.73×10^{-1} °	5.50×10^{-5} m	2.59×10^{-2} °
1	3.15×10^{-3} m	5.74×10^{-1} °	1.30×10^{-4} m	2.75×10^{-2} °
2	3.15×10^{-3} m	5.74×10^{-1} °	1.06×10^{-4} m	2.74×10^{-2} °
3	3.15×10^{-3} m	5.74×10^{-1} °	7.84×10^{-5} m	2.73×10^{-2} °

When observing this figure as a whole, it can be deduced that the plotted innovations go out of bounds at the same time-steps. This behavior is most likely due to a timing issue on the recording side[1]. Looking at the timestamps of the recorded messages regarding the SRS, it was found that the average and most delta times Δt of the messages were equal to 10 ms, i.e., 100 Hz. However, the timing in certain instances would vary between 0 ms to 24 ms as shown in Fig. 16.

To summarize Fig. 15, the absolute mean of the second half of the innovation is printed in Tab. VI, and the mean of the seconds half of its covariance is printed in Tab. VII.

TABLE VI
Average absolute innovation $\widetilde{\text{ABS}}(\boldsymbol{\epsilon})$ for select linear encoder for each configuration.

Config	ϵ_{l_1}	ϵ_{l_3}	ϵ_{l_5}
1	9.21×10^{-5}	9.74×10^{-5}	7.31×10^{-5}
2	1.55×10^{-4}	1.15×10^{-4}	8.61×10^{-5}
3	5.36×10^{-5}	4.23×10^{-5}	3.16×10^{-5}

TABLE VII
The sigma-bounds based on the average estimated observation covariance $\tilde{\boldsymbol{P}}_{\hat{y}}$ for select linear encoder for each configuration.

Config	$2 \cdot \sqrt{P_{\hat{y}_{l_1}}}$	$2 \cdot \sqrt{P_{\hat{y}_{l_3}}}$	$2 \cdot \sqrt{P_{\hat{y}_{l_5}}}$
1	1.86×10^{-4}	1.86×10^{-4}	1.86×10^{-4}
2	1.04×10^{-4}	1.04×10^{-4}	1.04×10^{-4}
3	1.53×10^{-5}	1.53×10^{-5}	1.53×10^{-5}

Fig. 17 and 18 show the estimated lags of Configuration 3, the configuration with setpoints, converge. The impact of the timestamp issue on the lag estimation is clearly visible around the 43 s mark in Fig. 17 and 18, where the lag comes close to zero. This jump in lag estimation of both the translational as well as the rotational lags aligns with especially large innovations $\boldsymbol{\epsilon}$ on the linear encoders caused by the timing issue. Here, the system computed that the reduction in lag was the likely solution, however the system estimated state was wrongly aligned in time with the observation.

[1] This behavior is likely due to the packet coalescing setting of the recording PC's network card [52], hindering the timely arrival of the various messages to the eCAL recording software.

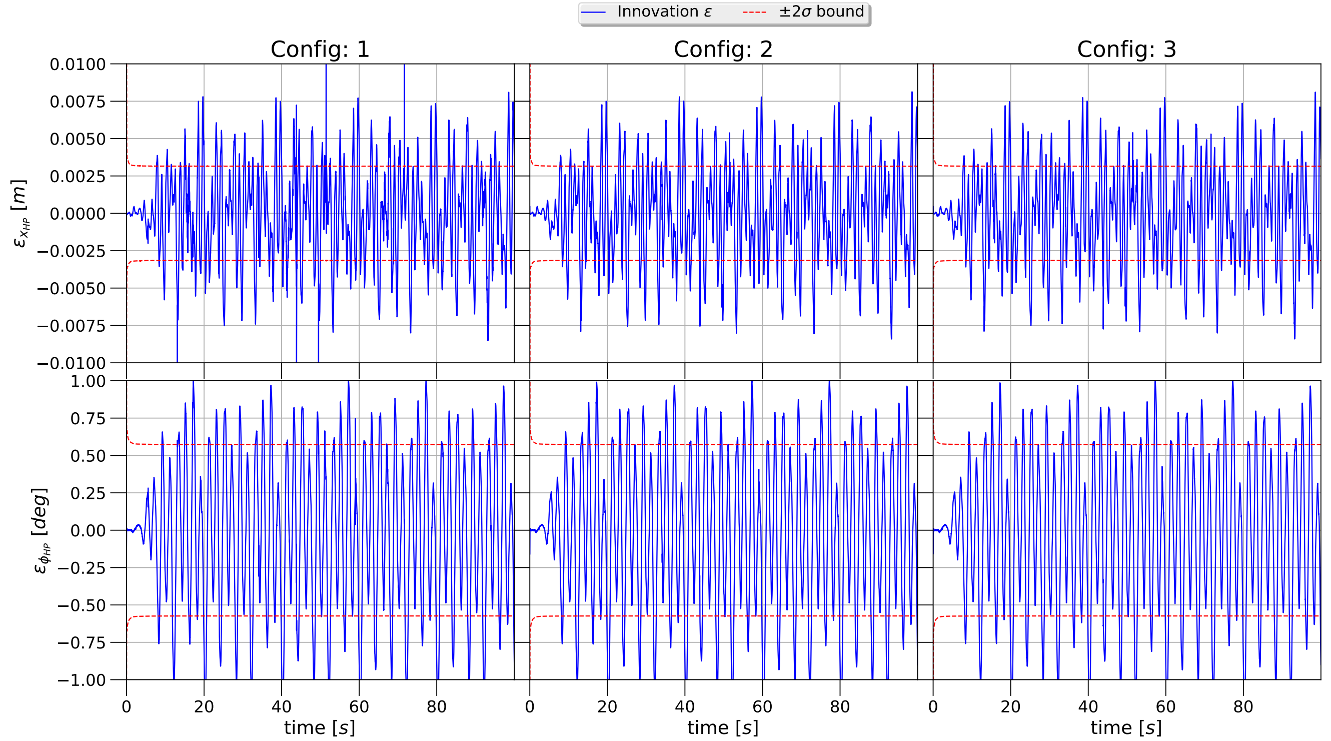


Fig. 14. The TrackIR 5 innovation $\boldsymbol{\epsilon}$ of the x-axis and roll attitude ϕ_{HP} per configuration plotted against its 95 % sigma-bounds (using the corrected noise parameters).

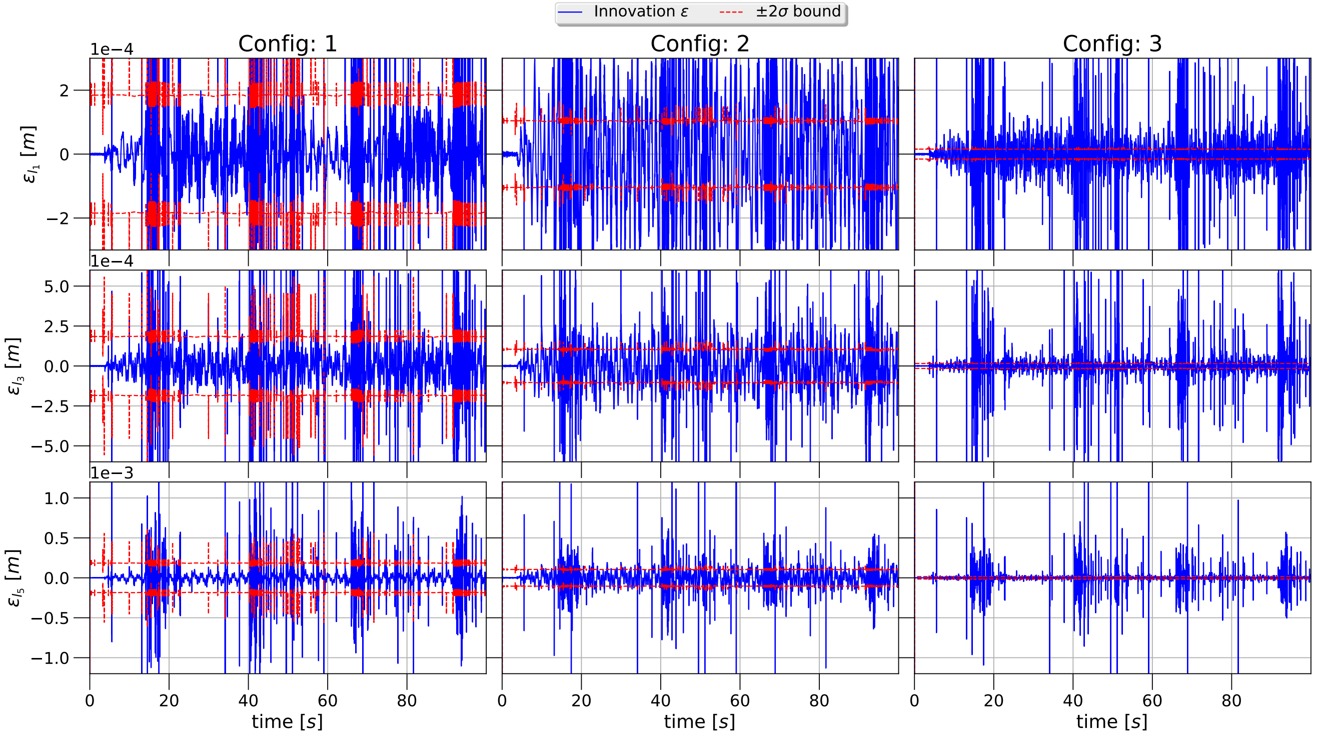


Fig. 15. The SRS's linear encoders innovation $\boldsymbol{\epsilon}$ of the first, third, and fifth leg per configuration. Note the difference in scale of the y-axis per leg.

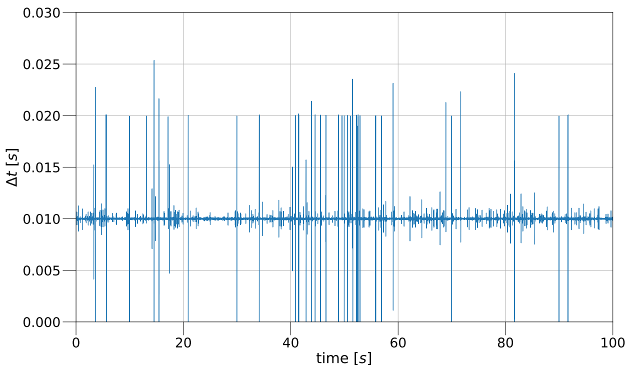


Fig. 16. Time difference Δt in seconds between recorded data messages of the SRS.

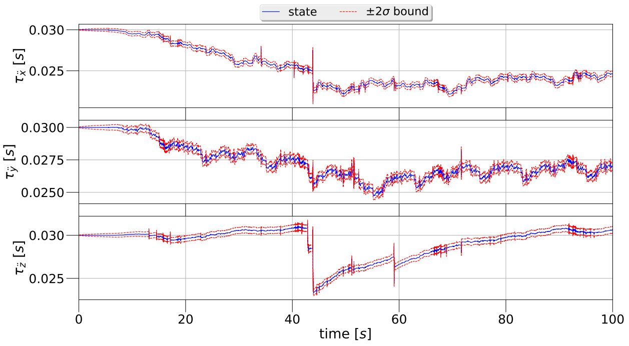


Fig. 17. SRS's translational lag plotted against its covariance.

IX. Discussion

The overall discussion is hindered by the unexpected delay on the TrackIR, which in turn restricts from making final claims on the accuracy of the HMD pose estimation.

The timing issue depicted in Fig. 16 would result in the UKF either over or under estimating certain states of the SRS in particular, due to an incorrect delta-time Δt variable used in the process models. However, as the algorithm in this case is performed in a deterministic manner, the results are still comparable, as each configuration has to deal with the same data at each time-step.

Despite the timing issue, each configuration is still capable of converging and estimating the state, leading to conclude that the presented solutions are robust. This is especially evident from the results of Configuration 3 (setpoints) in Fig. 17 and 18, where time related variables are a major part of its state.

The minimal contributions the configurations seem to have to the sigma-bound, as printed on the left side of Tab. V, is due to the fact that the corrected, i.e., scaled up, noise model of the TrackIR is inaccurate in comparison to the Stewart platform's estimation performance of any configuration presented. Thus, any potential difference in innovation covariance is hindered by the scaled up TrackIR's noise model. Conversely, the original noise parameters, on the right side of Tab. V, cause Configurations 2 (both systems have an IMU) and 3 (setpoints) to theoretically improve compared to Configuration 1 (only the HMD has an IMU). This is expected due to the reduction in process noise. Therefore, it can be concluded that from the perspective of the HMD's pose estimation, the need for improved simulator base's pose estimation is highly dependent on the accuracy of the visual sensor.

Furthermore, in Tab V Configuration 0, the one only using visual pose tracking, is seemingly theoretically superior to the other configurations, due to the reduced innovation deviation. However, Configuration 0 lacks the ability to have accurate interim predictions when compared to Configurations 1, 2, and 3 all of which include the HMD's IMU for fast pose updates. Meaning that the state of all three configurations is updated at 600 Hz, while the state of Configuration 0 only updates at ≈60 Hz. Thus, an argument can be made to opt for a visual only approach given that HMD's visual sensor would have a high enough update rate [7]. However, such an approach would likely yield higher costs due to the superior visual pose tracking sensor employed.

When analyzing Configuration 1, i.e., HMD with IMU but SRS without an IMU, on the presented motion profile, it can be concluded that the noise parameters of the SRS's acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ are tuned slightly too large. This is evident from the innovations in Fig. 15 when timestamps of the SRS's messages are not a problem, e.g., from 20 s to 40 s.

Next, Configuration 2 in Fig. 15, i.e., where both the HMD and SRS have an IMU as input, results in innovations that are coloured by the motion profile. This is likely due to the integration scheme used in the process model, assuming the measurements to be true, and thus constant, in combination with a relatively low SRS's IMU message update rate of 100 Hz. However, when the SRS's timestamps are not an issue, i.e., between 20 s to 40 s, Configuration 2 is within 95 % sigma-bounds. Considering the motion profile and the integration scheme used in

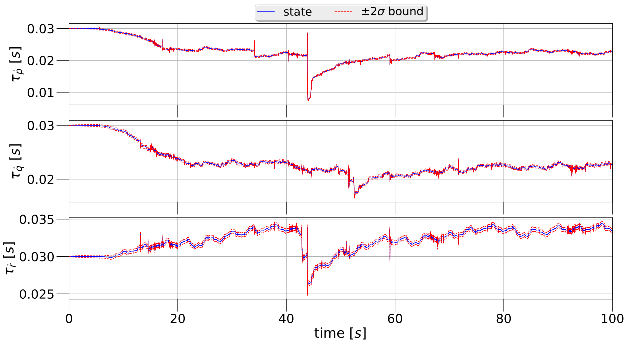


Fig. 18. SRS's rotational lag plotted against its covariance.

combination with the update rate of the SRS'S IMU, the noise parameters of Configuration 2 are well tuned. Increasing the SRS's message rate of the IMU would cause this configuration's process noise parameters to be lowered substantially, in turn increasing the accuracy of the SRS's pose estimation.

In Fig. 15 the innovations of Configuration 3, i.e., HMD with an IMU and SRS with setpoints, are not within sigma-bound even when the SRS's timestamps are as expected. Additional tuning was not found to yield any further improvements. In particular, both lag deviations, $\underline{\boldsymbol{\sigma}}_{\tau_{\ddot{c}_P}}$ and $\underline{\boldsymbol{\sigma}}_{\tau_{\dot{\omega}_P}}$, could not be increased further without compromising the algorithmic stability. This means that there potentially was more variation on the setpoint's lags noise model. A different setpoint model or way of integrating them into the KF would allow for more accuracy gains.

Also, increasing the SRS's deviations on acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ yielded mixed results. Increasing the deviations $\left(\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}, \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}\right)$ would lead to the innovation being mostly within sigma-bounds, but at the cost of an increased absolute innovation values, due to said deviations actually hindering the estimation of the setpoint's lags.

When comparing the innovation of all three configurations in Tab. VI, it can be concluded that in terms of average absolute innovation of the Stewart platform that Configuration 3 is superior despite its stability issues. The inherent algorithmic stability issues of Configuration 3 could be solved by using a multiple model approach [53]. In this scenario, the system would switch between Configuration 1 and 3 based upon the desired motion, i.e., SRS's setpoints. So, unless the SRS is moving along all 6-DOF, the system will use Configuration 1 rather than 3.

Even though the SRS has an IMU in Configuration 2, Configuration 1, the one without an IMU on the SRS, has slightly better performance for this motion profile as can be seen in Fig 15 and printed in Tab. VI, probably due to a couple of factors. First, the linear encoders are precise relative to the SRS's IMU when it comes to their contribution to the estimated pose. Second, to deal with the extra process noise at 100 Hz, the performance of the SRS's IMU is restricted as there are no additional accurate interim state predictions. Thus, it can be concluded that from the perspective of the SRS's pose estimation the addition of an IMU is an improvement only if the update rate is sufficiently high, and/or in combination with a more accurate integration scheme at the cost of a higher computational budget. From the perspective of the HMD's pose estimation, the added benefit of an additional IMU is potentially even smaller due to the estimation performance propagating through the visual pose tracking.

X. Conclusion

To conclude, by having an IMU on a HMD rather than solely basing the pose estimation on visual pose tracking will have an added benefit of having more accurate and numerous predictions between the observations by said visual pose tracking.

The impact of additional sensors or information via setpoints to the moving base on the HMD pose estimation depends on the accuracy of the visual pose tracking sensor, because the moving base pose estimation accuracy propagates through the visual pose tracking sensor. However, a different sensor than the TrackIR 5 to track the HMD within the cabin is recommended for this use case, due to TrackIR 5 large delay and closed proprietary software.

In principle, given accurate observation sensors of the moving base, measuring the absolute pose directly or indirectly, in this case indirectly via linear encoders, a configuration without an IMU and solely relaying on additive noise variables to the higher-order terms of the moving base, i.e., Configuration 1, is a sufficient minimal solution to properly estimate the HMD's cabin-fixed pose.

If an accurate IMU is available on the moving base, it should only be included if the IMU's update rate is high enough, i.e., larger than 100 Hz. This would allow for more accurate predictions between observations of the moving base, and therefore improve the HMD's cabin-fixed pose estimation. Currently, for the motion base's IMU, a frequency of 100 Hz in combination with the chosen integration methods, the integration error is too large relative to the IMU's measurement error, resulting in questionable pose estimation improvements at the cost of system complexity.

Conversely, if no IMU is available on the moving base, but superior pose estimation accuracy is required for the moving base, a setpoint based configuration could be used. However, due to stability issues concerning the lag estimation, a multiple model approach to the UKF algorithm based on two configurations presented in this paper could be a solution, and is therefore recommended for further research. This approach would switch between two models, where one model uses setpoints (Configuration 3), while the other one keeps the additive noise variables to the higher-order terms, but ignores the setpoint related states and variables (Configuration 1). This approach would combine the most accurate configuration while retaining algorithmic robustness.

When replicating this research, care should be taken to solve the ethernet related timing issue to improve the state estimation. Also, an effort should be made to test the configurations interactively, i.e. online.

References

[1] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-696, 8 2007. [Online]. Available: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf

[2] matzman666, "OpenVR InputEmulator," 2020. [Online]. Available: https://github.com/matzman666/OpenVR-InputEmulator

[3] S. Ropelato, F. Zünd, S. Magnenat, M. Menozzi, and R. W. Sumner, "Adaptive tutoring on a virtual reality driving simulator," in *International series on information systems and management in creative eMedia*, vol. 2017, 2017, pp. 12–17.

[4] M. Kaufeld and T. Alexander, "The Impact of Motion on Individual Simulator Sickness in a Moving Base VR Simulator with Head-Mounted Display," in *Virtual, Augmented and Mixed Reality. Multimodal Interaction*, J. Y. Chen and G. Fragomeni, Eds. Cham: Springer International Publishing, 2019, pp. 461–472.

[5] M. McCauley and T. Sharkey, "Cybersickness: Perception of Self-Motion in Virtual Environment." *Presence*, vol. 1, pp. 311–318, 01 1992.

[6] S. Rangelova and E. Andre, "A survey on simulation sickness in driving applications with virtual reality head-mounted displays," *Presence: Teleoperators and Virtual Environments*, vol. 27, no. 1, pp. 15–31, 2019.

[7] B. Blissing, F. Bruzelius, and O. Eriksson, "The Effects on Driving Behavior When Using a Head-mounted Display in a Dynamic Driving Simulator," *ACM Transactions on Applied Perception (TAP)*, vol. 19, no. 1, pp. 1–18, 2022. [Online]. Available: https://doi.org/10.1145/3483793

[8] E. M. Foxlin, "Head tracking relative to a moving vehicle or simulator platform using differential inertial sensors," vol. 4021, no. 1, pp. 133–144, 2000. [Online]. Available: https://doi.org/10.1117/12.389141

[9] X. Guo, C. Sun, P. Wang, and L. Huang, "Vision sensor and dual MEMS gyroscope integrated system for attitude determination on moving base," *Review of Scientific Instruments*, vol. 89, no. 1, p. 015002, 01 2018. [Online]. Available: https://doi.org/10.1063/1.5011703

[10] C. Walko and M.-J. Maibach, "Flying a helicopter with the HoloLens as head-mounted display," *Optical Engineering*, vol. 60, no. 10, p. 103103, 2021. [Online]. Available: https://doi.org/10.1117/1.OE.60.10.103103

[11] "libsurvive," accessed: 15-02-2022. [Online]. Available: https://github.com/cntools/libsurvive

[12] "P2 Pimax 8K Teardown Internal Photos Int Photos Pimax Technology," accessed: 30-01-2022. [Online]. Available: https://fccid.io/2ARKN-P2/Internal-Photos/Int-Photos-4077345

[13] "MPU-6500 Product Specification Revision 1.3," accessed: 31-01-2022. [Online]. Available: https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6500/\#product-documentation

[14] S. M. LaValle, A. Yershova, M. Katsev, and M. Antonov, "Head Tracking for the Oculus Rift," 2014. [Online]. Available: http://lavalle.pl/papers/LavYerKatAnt14.pdf

[15] Pimax Technologies, LLC., "Pimax 8k-x," accessed: 30-01-2022. [Online]. Available: https://pimax.com/product/vision-8k-x/

[16] NaturalPoint INC., "TrackIR 5," accessed: 30-01-2022. [Online]. Available: https://www.naturalpoint.com/trackir/trackir5/

[17] D. Stewart, "A Platform with Six Degrees of Freedom," *Proceedings of the Institution of Mechanical Engineers*, vol. 180, no. 1, pp. 371–386, 1965. [Online]. Available: https://doi.org/10.1243/PIME_PROC_1965_180_029_02

[18] O. Stroosma, M. M. van Paassen, and M. Mulder, "Using the SIMONA Research Simulator for Human-machine Interaction Research," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2012. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2003-5525

[19] M. M. van Paassen, O. Stroosma, and J. Delatour, "DUECA - Data-driven activation in distributed real-time computation," 08 2000.

[20] D. Pool, Q. Chu, M. M. van Paassen, and M. Mulder, "Optimal Reconstruction of Flight Simulator Motion Cues using Extended Kalman Filtering," in *Proceedings of the AiAA Modeling and Simulation Technologies Conference and Exhibit*, F. Weller and M. Duquette, Eds. United States: American Institute of Aeronautics and Astronautics Inc. (AIAA), 2008, pp. 1–20.

[21] "Heidenhain LC 400," accessed: 18-02-2022. [Online]. Available: https://www.heidenhain.com/products/linear-encoders/sealed/lc-400

[22] "SIMONA Research Simulator," accessed: 06-07-2022. [Online]. Available: https://cs.lr.tudelft.nl/simona/facility/visual-display-system/

[23] W. F. Phillips, C. E. Hailey, and G. A. Gebert, "Review of Attitude Representations Used for Aircraft Kinematics," *Journal of Aircraft*, vol. 38, no. 4, pp. 718–737, 2001. [Online]. Available: https://doi.org/10.2514/2.2824

[24] O. Stroosma, "The SIMONA Research Simulator's Motion Software," 2022, unpublished tech report.

[25] S. Koekebakker, "Model Based Control of a Flight Simulator Motion System," Ph.D. dissertation, Delft University, Delft, Netherlands, 12 2001.

[26] A. Kiruluta, M. Eizenman, and S. Pasupathy, "Predictive head movement tracking using a Kalman filter," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 27, no. 2, pp. 326–331, 1997. [Online]. Available: https://doi.org/10.1109/3477.558841

[27] H. C. Kam, Y. K. Yu, and K. H. Wong, "An Improvement on ArUco Marker for Pose Tracking Using Kalman Filter," in *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 6 2018, pp. 65–69.

[28] A. O. Ercan and A. T. Erdem, "On sensor fusion for head tracking in augmented reality applications," in *Proceedings of the 2011 American Control Conference*, 06 2011, pp. 1286–1291.

[29] A. Rhijn, R. van Liere, and J. Mulder, "An analysis of orientation prediction and filtering methods for VR/AR," 04 2005, pp. 67–74.

[30] I. Miletović, D. Pool, O. Stroosma, M. M. van Paassen, and Q. Chu, "Improved Stewart platform state estimation using inertial and actuator position measurements," *Control Engineering Practice*, vol. 62, pp. 102 – 115, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066117300643

[31] R. Merwe and E. Wan, "Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models," *Proceedings of the Workshop on Advances in Machine Learning*, 06 2003.

[32] G. van Rossum, *Python Tutorial*. Python Software Foundation, 9 2018.

[33] S. Konatowski, P. Kaniewski, and J. Matuszewski, "Comparison of Estimation Accuracy of EKF, UKF and PF Filters," *Annual of Navigation*, vol. 23, 12 2016.

[34] D. Hong-de, D. Shao-wu, C. Yuan-cai, and W. Guang-bin, "Performance Comparison of EKF/UKF/CKF for the Tracking of Ballistic Target," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 10, 11 2012.

[35] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[36] R. R. Labbe. (2021) Kalman and Bayesian Filters in Python. Accessed: 25-10-2023. [Online]. Available: https://drive.google.com/file/d/0By_SW19c1BfhSVFzNHc0SjduNzg/view?usp=sharing

[37] "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros," *IEEE Std 952-1997*, pp. 1–84, 10 1998.

[38] N. Kj, A. Sreejith, J. Mathew, M. Sarpotdar, A. Suresh, A. Prakash, M. Safonova, and J. Murthy, "Noise modeling and analysis of an IMU-based attitude sensor: improvement of performance by filtering and sensor fusion," 07 2016, p. 99126W.

[39] J. A. Farrell, F. O. Silva, F. Rahman, and J. Wendel, "IMU Error Modeling Tutorial: INS state estimation with real-time sensor calibration," 10 2021.

[40] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed. Boston, MA: Brooks/Cole, Cengage Learning, 2011. [Online]. Available: http://catdir.loc.gov/catdir/enhancements/fy1303/2010922639-d.html

[41] A. C. B. Chiella, B. O. S. Teixeira, and G. A. S. Pereira, "Quaternion-Based Robust Attitude Estimation Using an Adaptive Unscented Kalman Filter," *Sensors*, vol. 19, no. 10, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/10/2372

[42] Continental AG, *eCAL Documentation*, 2021. [Online]. Available: https://continental.github.io/ecal/index.html

[43] Python Software Foundation, "Python," 2023. [Online]. Available: https://www.python.org/

[44] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 09 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[45] Y. bar shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, 01 2004.

[46] B. D. O. Anderson and . Moore, John B. (John Barratt), *Optimal filtering*, ser. Prentice-Hall information and system sciences series. Englewood Cliffs, N.J.: Prentice-Hall, 1979. [Online]. Available: http://www.zentralblatt-math.org/zmath/en/search/?an=0688.93058

[47] J. Carmack. Latency Mitigation Strategies. Accessed: 25-10-2023. [Online]. Available: https://danluu.com/latency-mitigation/

[48] B. M. Lansdorp and O. A. Saleh, "Power spectrum and Allan variance methods for calibrating single-molecule video-tracking instruments," *Review of Scientific Instruments*, vol. 83, no. 2, p. 025115, 2012. [Online]. Available: https://doi.org/10.1063/1.3687431

[49] "Inertial Measurement Unit (IMU) ISIS-IMU (Rev. C)," accessed: 08-10-2023. [Online]. Available: https://fliphtml5.com/dgcj/qkwi/basic

[50] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., vol. 3068, International Society for Optics and Photonics. SPIE, 1997, pp. 182 – 193. [Online]. Available: https://doi.org/10.1117/12.280797

[51] S. K. Advani, "The Kinematic design of flight simulator motion bases," Ph.D. dissertation, Delft University, Delft, Netherlands, 04 1998.

[52] Microsoft, "Overview of Packet Coalescing," 2023, accessed: 30-10-2023. [Online]. Available: https://learn.microsoft.com/en-us/windows-hardware/drivers/network/overview-of-packet-coalescing

[53] M. J. Quinlan and R. H. Middleton, "Multiple Model Kalman Filters: A Localization Technique for RoboCup Soccer," *Lecture Notes in Computer Science*, no. 5949, pp. 276–287, 2010. [Online]. Available: https://mural.maynoothuniversity.ie/2510/

II

Preliminary Report

1 Introduction

In 1965, Ivan Sutherland was one of the first researchers to describe modern Virtual Reality (VR) systems in his paper 'The Ultimate Display' [66]. In this work, a system was described using a Head-Mounted Display (HMD) to present a visual representation of the virtual world relative to the user. Three years later Sutherland built and presented his first version of 'The Ultimate Display' [65], a VR system showing rudimentary shapes in a virtual room on a HMD. Sutherland made the following observation regarding head tracking [65]:

> "Although stereo presentation is important to the three-dimensional illusion, it is less important than the change that takes place in the image when the observer moves his head."

The VR/Augmented Reality (AR) technology has since developed, and is now a sophisticated tool which can be used for various purposes in various fields, such as engineering, architecture and video game design. A VR/AR system provides an immersive experience as the user's gaze of the virtual world is rendered onto the HMD. If placed in a motion platform simulator, a VR/AR system could provide a larger Field-of-Regard (FoR) than conventional out-of-window visual system currently in use. These would mostly improve rotor-craft and fighter jet simulators, which would then be able to simulate the reality in a more immersive way and thus provide a more realistic exercise. Moreover, if one is to design a motion platform with a VR system from the outset, significant weight reductions are possible, due to the removal of the conventional display technology. This would allow an increase in the responsiveness of the motion based simulator, as the force to weight ratio would be improved, as well as possibly reduce the cost of production.

Nowadays, head tracking in VR and AR systems are commonly performed by a combination of inertial and visual-based sensors. Inertial sensors present on a HMD, namely Micro-Electro-Mechanical Systems (MEMS) based Inertial Measurement Unit (IMU), provide high-frequency low-latency pose, meaning position and attitude, updates. IMU's inherent implementation for pose estimation causes drift over time [73]. On the other hand, visual-based sensors, commonly in the form of light-point or image tracking, track the relative head-pose, eliminating the drift due to the IMU implementation, albeit at a lower frequency than the MEMS based IMU. Two types of VR/AR systems can be distinguished based on the Degree-of-Freedom (DoF). Three-DoF VR/AR systems typically only use a MEMS-based IMU. More advanced systems, which are the focal point of this project, in addition to an IMU also typically include a form of visual tracking to enable reliable six-DoF VR/AR systems. In principle, using the visual-based sensors would suffice for the head tracking, but most VR/AR systems also include MEMS based IMU, resulting in an increase in accuracy and update rate.

However, placing such an advanced six-DoF VR/AR system inside an enclosed motion platform simulator would result in the movement of the motion platform being picked up by the IMU. The sensor would interpret such movement from the motion platform as head movement in a traditional VR/AR system, which will inevitably distort the estimated head-pose. Consequently, the distorted head-pose causes a sensory mismatch between the user's visual and vestibular systems [39, 25, 55], which would have a significant negative impact on the user experience and render the system unusable.

The following sections of the this introduction will provide more detailed description of the research framework, as well as the report itself.

Research Objective and Questions

The aim of this Thesis is to enable the usage of IMUs contained in advanced VR/AR systems within an enclosed motion-based simulator. This will allow for the full potential of such systems to be available for use in such an environment. Thus, the objective of the research is stipulated as follows:

> **Present a solution on the implementation of a Head-Mounted Display with both an Inertial Measurement Unit and visual tracking inside an enclosed motion platform by developing a sensor fusion algorithm that generates a proper relative pose estimate.**

In order to attain the set out objective, the following primary research Research Question is identified:

> **What is an effective sensor fusion algorithm that generates a proper relative pose of a Head-Mounted Display located within an enclosed motion platform.**

The main Research Question is further fractionated in sub-questions which will simplify the process of reaching a solution to the problem, namely:

- *What are the characteristics of an effective state estimation algorithm that implements a head mounted display on a motion platform?*
- *What is the impact of incorporating the motion cues for the motion platform to improve the relative pose estimate?*
- *What is the impact of having an inertial measurement unit on the motion platform when estimating the pose of the head mounted display?*

In order to answer the proposed Research Questions, the general head tracking of a VR/AR system should be understood, including the algorithms used for this purpose. In addition, the pose estimation algorithms of the motion platforms themselves should be investigated. Moreover, a further understanding of the motion cues of the motion platforms and their relation to the movement of the motion platform themselves are required. Also, for finding the necessary characteristics of the algorithm, the sensor noise models should be understood. These elements will be tackled in the upcoming literature review.

Report outline

In the subsequent Chapter 2, a literature review is performed on the topics of related literature, HMD head-tracking, motion platforms and state estimation algorithms followed by a conclusion on the literary section. Next, in Chapter 3, the chosen state estimation algorithm is explained in detail, where after the algorithm is expanded on to deal with the potential latency on one of the observation sensors. Following, in Chapter 4, the individual continuous kinematic models of both the VR/AR system as well as the Stewart platform are created and explained. Using the established individual continuous kinematic models, three configurations are created where the VR/AR systems and Stewart platform are combined in Chapter 5. Specific sensors which will be used in the experiments are described in detail and their noise models parametrized in Chapter 6. Furthermore, in Chapter 7 the continuous kinematic models are discretized, the senor data is generated based on the defined motion, and parts of the algorithm are redefined to implement the quaternion attitude. Moreover, the algorithm, detailed in Chapter 3, the three configurations kinematic models, defined in Chapter 5, and the sensor parameters, found in Chapter 6, are combined and simulated in Chapter 7. Following the conclusion on the analysis of the simulation, an experiment plan is generated in Chapter 8 to validate the initial findings of this Thesis on the SIMONA Research Simulator at TU Delft. Finally, in Chapter 9 this Thesis is concluded and the Research Questions are answered accordingly.

2 Literature Review

In this chapter, the relevant work in the field is presented to gain an insight into the current understandings of the topics of research. Due to the scope of this research this literature study will not only focus on theories regarding this field, but also look into the practical aspects of the equipment to be used to validate the proposed theoretical approach.

First, research related to Virtual Reality (VR)/Augmented Reality (AR) system implementations on motion platforms are presented in Section 2.1. Due to the lack of previous research attempts to build onto, each system individually, VR/AR system and Stewart platform, is further analyzed. For this reason, VR/AR systems are discussed in Section 2.2 and elements, such as the sensors and the algorithm, contributing to the pose estimation are investigated. A similar approach is taken for the motion platforms in Section 2.3, first a general overview is given before diving into the elements that are relevant for the pose estimation. Finally, the literature review is concluded in Section 2.4.

2.1. Previous Research

VR/AR implementations were only found in simulators on motion platform based on an open design [68, 58, 25]. This meant that the visual tracking for the VR/AR system could be fixed to the room the simulator was in, i.e. the inertial reference frame. The pose relative to the cockpit, i.e. the vehicle-fixed pose, was computed by deducting the motion platform's pose from the estimated head-pose.

However, none of the research papers listed above detailed the head-tracking. The expected justification for this is that a solution for the open simulators exists, which is an open-source software package in combination with off-the-shelf components [38]. This solution only works if the absolute position tracking is not mounted on the motion platform but rather on the ceiling or walls instead, thus fixed to the inertial frame of reference. In combination with mounting a trackable object, such as the vive tracker, compatible with the visual system to the chair of the user and deducting the pose of the chair from the pose of the Head-Mounted Display (HMD) in software [58, 25]. This is, however, not possible in fully enclosed simulators such as SIMONA Research Simulator (SRS), where visual tracking can only be done with respect to (w.r.t.) to the moving platform reference frame, from now on called vehicle-fixed. No research was found on the implementation of a VR/AR system in a fully enclosed existing simulators. More details on SRS can be found in Section 2.3.1

2.2. Artificial and Virtual Reality Systems

A six-Degree-of-Freedom (DoF) VR/AR system consist out of a HMD, Inertial Measurement Unit (IMU) fixed to the HMD, and visual tracking [17, 3]. Due to the scope of this Thesis the focus is on the IMU in the HMD and the visual tracking of the VR/AR system rather than the graphics pipeline or the display technology.

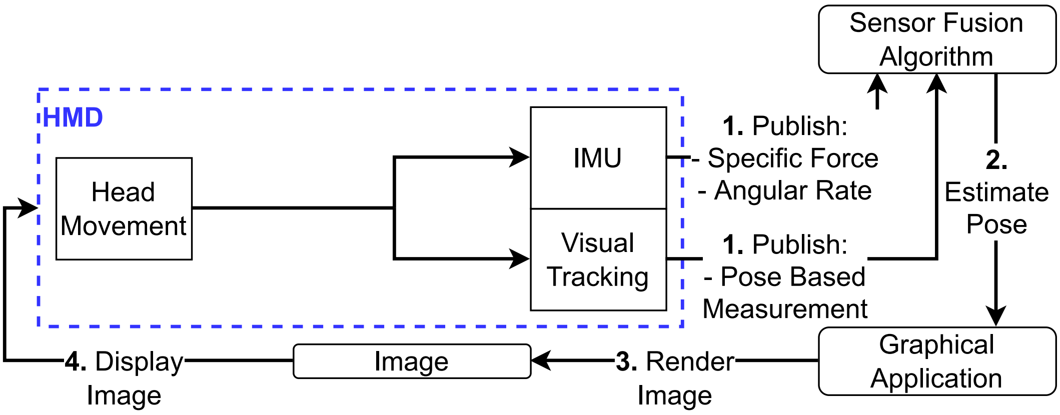


Figure 2.1: General working of a VR/AR system consisting of an IMU and visual tracking.

In Figure 2.1, the general working of a VR/AR system is given. The steps listed in the figure are listed and expanded on if necessary:

1. Sensor data is published onto the data bus.
2. A new pose is estimated based on the sensor data and the previous pose.
3. Virtual world is simulated based on inputs and Image is rendered.
4. Image is send to headset and displayed.

Here, step 1 and 2 are the main consideration in this Thesis. For this reason, Section 2.2.1 and 2.2.2 delve deeper into the IMU and visual tracking sensors. In order to facilitate the modeling and simulation of said sensors.

In Section 2.2.3 various algorithms used to estimate the pose of a HMD within a VR/AR system are listed and discussed.

The user's pose is estimated by combining the measurements of various sensors on the HMD, also called sensor fusion. In this section, however, the focus will be on the various sensors the HMD uses to estimate its own pose.

Figure 2.2: A Head-Mounted Display, the Pimax 8k-x, compatible with SteamVR systems [52].

2.2.1. Inertial Measurement Unit

In the case of a six-DoF VR/AR system, Micro-Electro-Mechanical Systems (MEMS) based IMU are installed in the HMD. These IMU's consist out of an accelerometer, gyroscope, and in certain cases a magnetometer is also present [17, 3].

An accelerometer measures the specific force applied to it and the gyroscope measures the angular rate. To yield position and attitude, a certain set of operations, shown in Figure 2.3, are applied to the specific force and angular rate measured by the IMU. In this figure the attitude is ascertained by integrate the angular rate over time. In turn, the attitude is used to remove the gravity's influence from the specific force, resulting in the accelerations in a reference frame. These accelerations are twice integrated over time while taking the initial velocity and position into account.

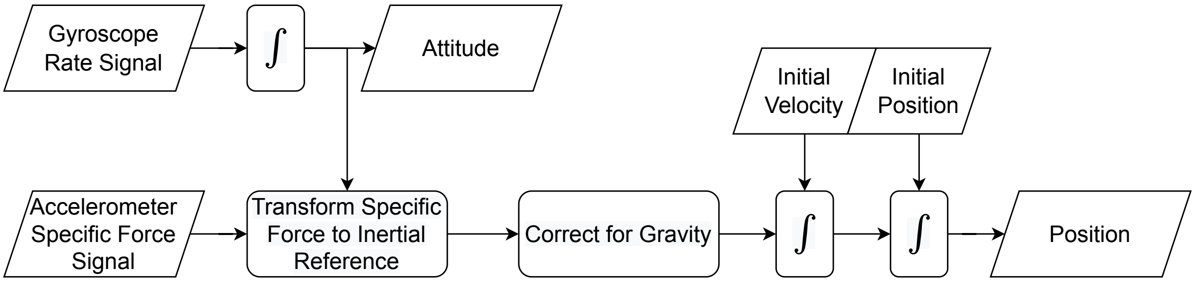


Figure 2.3: Strapdown inertial navigation algorithm example [73].

Both the MEMS accelerometer and gyroscope have errors in their measurement. These inaccuracies are caused by a number of sources, namely [73, 21]:

- The **Constant Bias** of an IMU sensor is the sensors average offset from the true value. The constant bias can be found in both the accelerometer and gyroscope by performing a static measurement, i.e. the sensor is held still. In the case of the accelerometer, the exact attitude should be known to take gravity into account.
- The **Angle/Velocity Random Walk** are the white noise perturbations on the measurements. Called angle random walk in case of the gyroscope and velocity random walk in case of the accelerometer. These perturbations are caused by thermo-mechanical fluctuations, at a higher rate than the sample rate.
- The **Bias Stability** of an IMU is the low frequency wandering of said sensor's measurements around the established constant bias.
- The **Rate/Acceleration Random Walk** are bias changes over a long period of time. The origin of these changes are not always known, but one of the sources are changes in temperature.
- **Calibration Errors** is the collective term for errors related to alignment, scale factors, and linearities. Theses errors can only be found while the IMU is undergoing a known motion.

By analyzing the noise signal of an IMU in a stationary position the Constant Bias, Angle/Velocity Random Walk, Bias Stability, and Rate/Acceleration Random Walk can be found. Angle/Velocity Random Walk, Bias Stability, and Rate/Acceleration Random Walk can be extracted by computing the Allan Variation (AV) [21, 73, 27, 15]. The AV is a time-domain analysis method to estimate the frequency stability of oscillators. AV, has been used successfully to estimate the stochastic errors of an IMU.

Using an IMU solely for pose estimation as a system such as shown in Figure 2.3 causes the estimated pose, i.e. the estimated attitude and position, to drift over time. The estimated pose drifts due to integrating the errors present in both the gyroscope and accelerometer. The position drifts even faster, relatively speaking, in a system such as presented in Figure 2.3, as the attitude is used to remove the influence of the gravity on the specific force. The resulting acceleration is integrated double. Not only scale the errors faster over time when comparing the attitudinal and the positional errors (linear vs quadratically), an error in the attitude estimation causes a seemingly increase in the bias of the accelerometer due to the aforementioned coupling of attitude and positional estimation.

2.2.2. Visual Tracking

Visual tracking is used in six-DoF VR/AR systems [3] as it provides absolute tracking of the position and attitude. The fact that the pose can be deduced directly from the measurements without integration makes visual tracking a good combination with an IMU, capable of removing the drift that plagues an IMU based system. The disadvantage of most visual tracking systems is their slow update rate, relative to the IMU. Two main methods are identified for visual tracking, namely inside-out and outside-in based visual tracking, and can be found in Sections 2.2.2.1 and 2.2.2.2 respectively.

Inside-out

With inside-out tracking, also called 'ego-motion tracking' [17], the HMD uses sensors on the HMD to track its own pose within the environment. The act of the HMD tracking its own pose by sensing the environment can be performed in multiple ways, but most commonly by using a stereo camera on the HMD and an algorithm such as Simultanious Localization And Mapping (SLAM) [28]. In theory the trackable workspace of the HMD has the potential to be infinite and requires no setup. In practice inside-out tracking is computational intensive. Moreover, the tracking performance is strongly influenced by lighting conditions and the requirement that the environment should remain largely static [12]. Two conditions that are not easily guaranteed in an enclosed flight simulator.

The inside of an enclosed moving base simulator is a known environment. For this reason specific markers, called landmarks, can be placed [46, 17]. The addition of such known landmarks reduce the computational cost and lighting requirements when compared to algorithms such as SLAM . Also, unambiguity due to the tracking of the motion on the displays in the cockpit is removed. However, this solution removes the main advantage of inside-out tracking, namely the potential infinite workspace and the fact that no setup is required.

These landmarks can be very advanced, in the case of the HTC vive [46], a popular VR system, the landmarks emit Infrared (IR) light in a specific patter at a specific frequency in multiple locations order to avoid unambiguity. These IR lights are subsequently recognized by the multiple IR-sensor on the headset. Due to the specificity of the signal of the landmarks combined with the upfront known location of these landmarks the pose can be estimated with relatively small computational cost [46, 7].

A current example of inside-out tracking is the current family of SteamVR compatible devices [46, 7]. The system employs at least two landmarks, in this case called Base Stations, emitting a specific IR light pattern using a spinning IR emitter. Inside the Base Stations is an IMU to estimate the pose of the Base Station. Furthermore, the Base Stations relative pose to one another is computed by receiving another's IR light pattern. The IMU inside the Base Station also register any motion. If any motion is registered the Base Stations are shut down and upon restarting their relative pose to the other Base Stations is recomputed. This last feature also protects the spinning IR emitters. This is due to the fact that the Base Stations based HMD-tracking is functional due to the tight time-syncing between the receivers on the headset and the IR emitters. Forces and rotations acting on these Base Stations would speed up or down the emitters rotational speed, potentially causing a time-sync issue. Making these Base Stations unsuitable for use inside a moving simulator.

Outside-in

Outside-in tracking, contrary to inside-out tracking, sensors in the environment track the HMD pose. In most cases the HMD is broadcasting a specific pattern, actively or passively.

For the active case, IR emitters are most often fixed to the HMD. One or multiple IR sensors are placed in the environment to pick up the distinguish pattern indicating the HMD pose. An current example of active outside-in tracking is the Oculus Rift [34]. Whereby, two IR sensors detect the specific pattern broadcasted by the IR transmitters on the HMD. In these kind of systems synchronization is often a requirement between the IR emitters and the sensors.

A passive pattern consist mostly out of a known 3D shape attached to the HMD and tracked by a mono or stereo camera system setup in the environment. In certain tracking systems, such as TrackIR [69] and ART [4] systems, the known 3D shape is reflective and reflects back the IR light send by the camera. Hereby forgoing the need for synchronization.

The advantages and disadvantages are the same as the outside-in tracking with landmarks as the pattern emitter and sensor locations are swapped from HMD to environment and vice versa.

2.2.3. Head-Mounted Display Tracking Algorithm

Head-Mounted Display tracking is a key element in any VR/AR system, in this section the various HMD tracking algorithms found are summarized in relation to this Thesis. The aim of any tracking algorithm is is to provide the most accurate estimate on the current or future state. Errors such as noise, apparent latency and jitter diminish the experience and can cause virtual sickness [56, 39, 74, 67].

For the HMD tracking for the Oculus Rift, LaValle et al. presented a Complementary Filter (CF) based head tracking solution for a low-cost MEMS sensors VR implementation [34]. Here, certain IMU related problems were addressed. The designed CF, a computationally inexpensive sensor fusion technique consisting of a low-pass and a high-pass filter [45], assumed the average specific force over a short period of time approximates the gravity vector and deal with the drifting attitude tilt, the growing tilt a result of integrating the gyroscope errors. By calibrating and centering the magnetometer, part of their IMU package, the yaw drift was also successfully eliminated. However, the positional drift was only limited by using the kinematics of the upper body and assuming the lower body to remain stationary. Hence, it was concluded that the CF in combination with only the MEMS based IMU were insufficient for six-DoF and it was recommended that a visual tracking system should be added for improved accuracy.

In many cases of HMD tracking using an attached IMU and a form of visual tracking a Kalman Filter (KF)-type of sensor fusion algorithm was used [26, 24, 14, 56]. A KF obtains an optimal estimate of a stochastic system given the process model and sensor measurements. The standard KF works on linear process models and observation models [23]. Extension to the standard KF algorithm, such as Extended Kalman Filter (EKF), Iterative Extended Kalman Filter (IEKF), and Unscented Kalman Filter (UKF) [41], have been developed to better deal with the nonlinear process, observation and noise models.

2.3. Stewart Platform Based Flight Simulator

In this section, a short study is compiled on Stewart platform flight simulators, in particular their motion platforms. A Stewart platform has six degrees of freedom and is formed by two connected rigid frames [63]. The two rigid frames are connected by six cylindrical actuators with Hooke joints, a joint with only 2 degrees of freedom, at each end. A simplified geometric representation of a Stewart platform is shown in Figure 2.4.

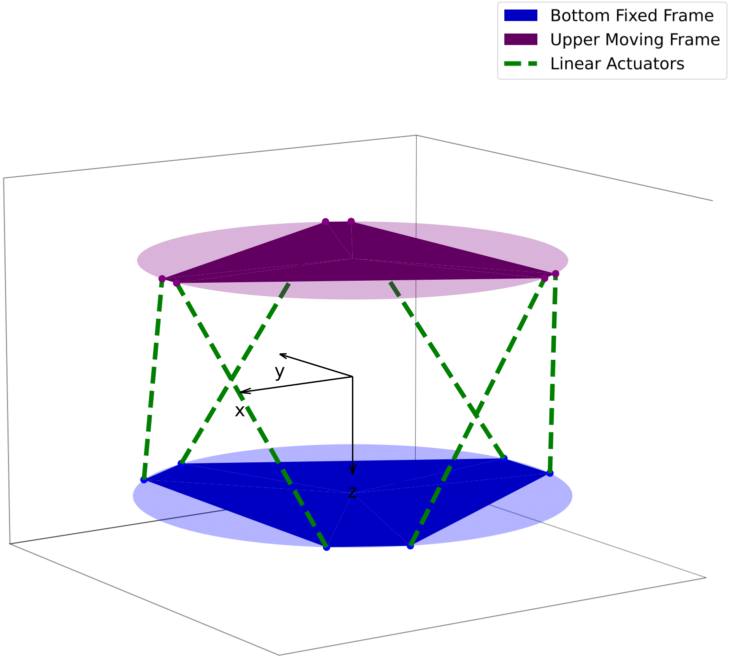


Figure 2.4: Simplified geometric representation of a Stewart platform.

While a range of different motion platform systems are available, the motion platform often chosen, and available for integration and testing, is a Stewart platform [63]. Of the motion platforms, the motion system, motion cues, and the algorithms that estimate the pose of the simulator are of interest. These aspects of the motion platforms need to be understood in order to integrate a VR/AR system into a motion platform simulator.

First, a high-level overview of a ground-based flight simulator is provided. Second, the Flight Simulator provided, SRS, is further explored in terms of the previous discussed topics. Next, algorithms used to estimate the upper moving frame pose of a Stewart platform are analyzed in Section 2.3.2. Final, the motion cues are looked into.

Ground-based flight simulators provide a cost-effective method to asses pilot-vehicle behavior [1]. Moreover, ground-based flight simulators grant the added benefit that there is total control of the situation and environment within the simulation. Motion platform simulators in particular, capable of synthesizing the motion stimuli for the pilots, improve fidelity [1].

A high-level overview of the workings of a motion platform simulator is shown in Figure 2.5 based on work done by Advani in [1].

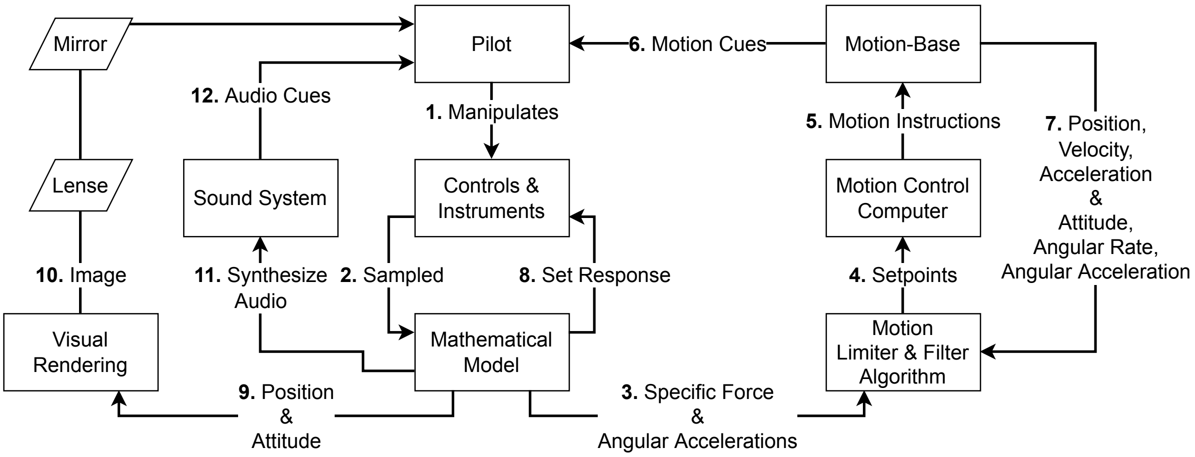


Figure 2.5: A high-level overview of the workings of a motion platform simulator.

In this figure, **1.** the pilot manipulates the controls. **2.** The controls are sampled, and serve as the input to the mathematical model of the simulated vehicle. **3.** The model generates the specific force and angular acceleration, and passed through a Motion Limiter and Filter Algorithm. **4.** Resulting in the Motion Command Signals, or **setpoints**. **5.** The Motion Control Computer interpret the setpoints to generate the Motion Instructions specific for the Motion-Base at hand. **6.** The generated Motion Cues are felt by the pilot. **7.** In order to compute the setpoints the estimated position, velocity, acceleration, attitude, angular rate, and angular acceleration of the motion platform are required. **8.** Based upon the state of the virtual model a response is computed for the Controls and Instruments. **9.** At the same time, the virtual pose, position and attitude, of the simulated vehicle initiate the generation of the visual image. **10.** The projected image is refracted through a lense and reflected onto a mirror to the pilot. **11.** Meanwhile, audio fragments are synthesized and combined based on the state of the simulated vehicle. **12.** Audio is played over the Sound System to the pilot.

2.3.1. SIMONA Research Simulator

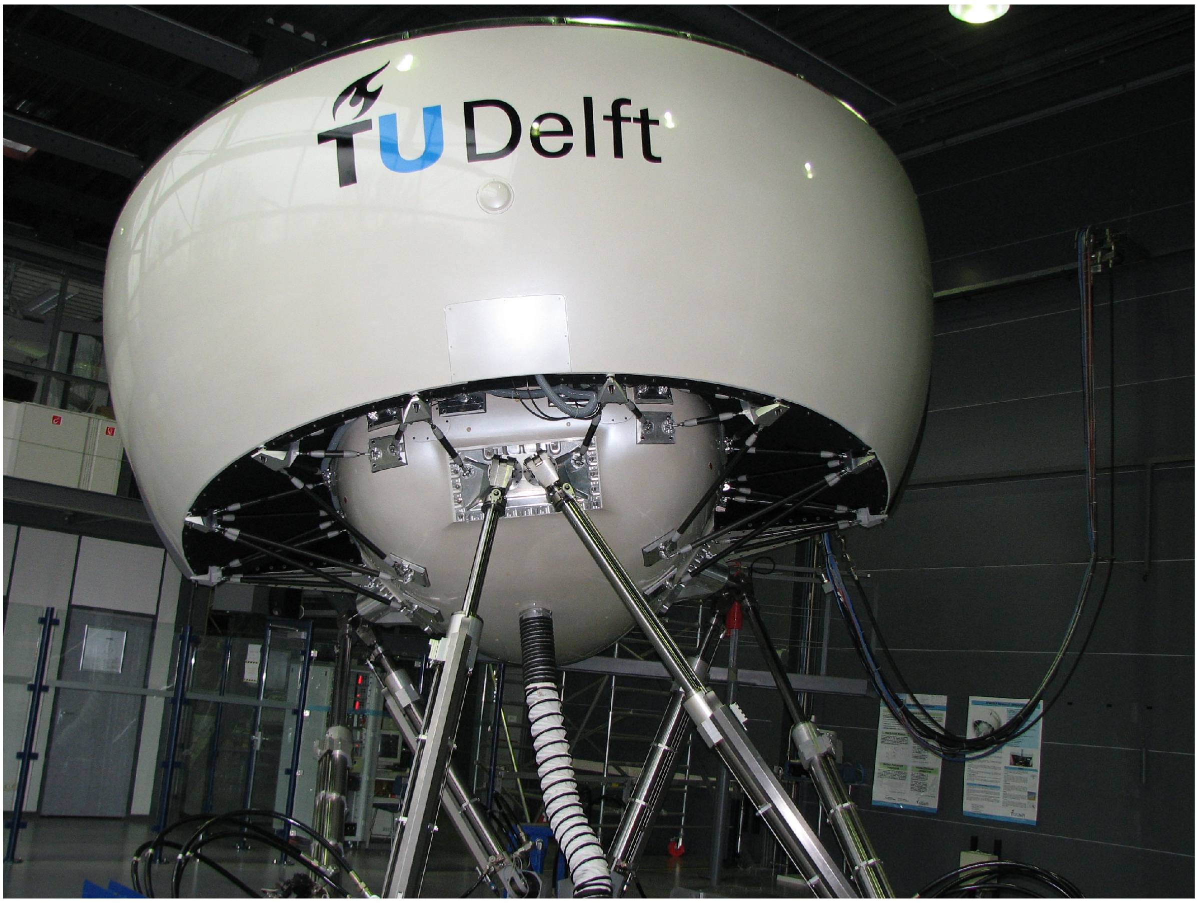


Figure 2.6: SIMONA Research Simulator [62]

Presented in Figure 2.6 is SIMONA Research Simulator (SRS), a high fidelity moving base simulator used for research purposes at TU Delft. The moving base has a Stewart platform based design [1], granting it six-DoF. This specific Stewart platform has six hydraulic actuators, with a stroke length of 1.25 m. A absolute linear encoder is attached to each of the six hydraulic actuators, measuring the length of each actuator. Moreover, an IMU is attached to the moving platform, capable of measuring the specific force and angular rate.

2.3.2. Pose Estimation Algorithms

In this section the pose estimation algorithms used in case of a Stewart Platform are investigated. It was found that the pose, position and attitude, of a Stewart platforms can be estimated in one of two generalized methods.

The first kind of method uses the forward kinematic, by computing the pose from the measured lengths of the connecting linear actuators. Due to the possibility of multiple poses corresponding to a measured set of the connecting linear actuators a closed algebraic form is not possible and iterative schemes such as Newton-Raphson (NR) or Gauss-Newton (GN) method [29, 13] are most commonly applied.

The second type of method uses the reverse kinematics of the first method, known as the inverse kinematics. The inverse kinematics compute the lengths of the connecting linear actuators from the pose of the Stewart platform. These computations are trivial. In practice, the inverse kinematics are often used in combination with a type of KF [54, 43].

Also, using a sensor fusion algorithm, such as IEKF or UKF, in combination with the inverse kinematics has the added benefit that other, higher order states can be estimated online [41, 23]. These higher order states are beneficial for control purposes as mentioned in [54]. Moreover, KF methods take into account the Gaussian error spread of sensors, improving the estimated pose. Moreover, the integration of extra sensors to reduce estimation error further and increase estimation robustness is trivial [43].

When comparing the EKF, and by extension IEKF, to the UKF, the latter was found to capture up until the second order when propagating the estimated state and its covariance using the Scaled Unscented Transformation (SUT) when compared to the Taylor expansion [41]. The EKF only managing to capture the first order fully as it only linearizes around the current estimated state. Meaning that the UKF should be more capable in estimating non-linear models.

In practice, the UKF was found to have similar computational cost to the EKF [59, 43, 30, 20] but improved estimation performance.

2.3.3. Setpoints

In most cases, setpoints, or the Motion Command Signals, can be seen as the requested state or the future state of the Stewart platform. These setpoints are the result of the filtered specific force and angular acceleration of the mathematical model, and can consist of position, velocity, acceleration, attitude, angular rate and angular acceleration. In parallel, these setpoints are limited by the physical aspects of the Stewart platform, given the current estimated position, velocity, acceleration, attitude, angular rate and angular acceleration of the Stewart platform.

Koekebakker states in [29] that a motion system with *'An inner loop feedback and feedback linearising control result in first order response of the system, from desired to actual accelerations.'* Meaning that if the assumption holds where an inner loop feedback and feedback linearising control systems are used to control the Stewart platform, a simplified model can be construed to predict the state of the Stewart platform. Given the desired acceleration, i.e. setpoints, the actual acceleration can be approximated by the response of a first order system.

No literature was found on the implementation of setpoints into a KF. In a way this makes sense as to compute the setpoints the state should already have been estimated by the system beforehand. However, it can be assumed that these setpoints do provide a window to the future state of the system. Moreover, it is highly likely that a setpoint are not Gaussian, so an algorithm needs to be selected that is at least lenient to non-Gaussian variables.

2.4. Literary Summary

As above mentioned, open design motion platform simulators are already capable of hosting a VR/AR system. Hence, due to the lack of research, particularly encompassing the implementation of a VR/AR system in an enclosed motion platform, this Thesis is set out to offer enhanced performance of advanced six-DoF VR/AR systems.

One thing to consider when working with an IMU are the various noise sources and their impact on the pose estimation. A pose estimated purely by measuring specific force and angular rate can only stay correct for short periods of time. Moreover, the AV method can be used to compute the noise sources on an IMU.

Visual tracking complements the IMU well as the pose can be deduced directly from the measurements, negating the drift caused by the IMU. Irrespectively of inside-out or outside-in tracking the use of landmarks or known 3D patterns is beneficial as the environment, meaning the cockpit in the flight simulator, is a known entity. A notable majority of the found literature was successfully utilizing a type of KF to estimate the pose of the HMD.

Motion Command Signals, or setpoints, could be employed to improve the estimation performance but literature here is limited. Recently sensor fusion algorithms, such as the UKF, have been used to estimate the Stewart platform's state. Since types of KF have been used to estimate both systems individually, i.e. VR/AR system and Stewart platform, it stands to reason that a KF is a proven choice to be used when combining both into one system. Hence, the algorithm should be capable of estimating a stochastic multivariate non-linear system and the UKF seems like a preferred choice.

3

Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a Minimum Mean-Square-Error (MMSE) sequential estimator [41] based on the Kalman Filter (KF) framework [23].

The KF framework, the UKF is based on, estimates the state $\underline{\hat{\boldsymbol{x}}}$ based on direct or indirect measurements of the actual state $\underline{\boldsymbol{x}}$. The KF framework estimates the state $\underline{\hat{\boldsymbol{x}}}$ of a system by combining measurements of the system, including the knowledge of the measurement's probability distribution, with the previous state and its covariance in a Gaussian approximate optimal manner [41]. The KF framework works in two phases, the prediction and the correction phase, the latter also called the update phase.

In the prediction phase, the previous estimated state $\underline{\hat{\boldsymbol{x}}}_{k-1}$ and previous state covariance $\boldsymbol{P}_{\hat{x}_{k-1}}$ are projected to the next time step $(k-1 \rightarrow k)$ to the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ and predicted state covariance $\boldsymbol{P}^-_{\hat{x}\ k}$. The predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ is the projection in time of the previous estimated state $\underline{\hat{\boldsymbol{x}}}_{k-1}$ itself and an external input $\underline{\boldsymbol{u}}_{k-1}$ using the process model F, usually based on a physical model of the system, i.e.,

$$\underline{\boldsymbol{x}}_k^- = \mathrm{F}\left(\underline{\boldsymbol{x}}_{k-1}, \underline{\boldsymbol{u}}_{k-1}\right) \tag{3.1}$$

While at the same time, the uncertainty of the process, i.e. process noise $\underline{\boldsymbol{\nu}}$, is appended to the previous state covariance $\boldsymbol{P}_{\hat{x}_{k-1}}$ resulting in the predicted state covariance $\underline{\hat{\boldsymbol{x}}}_k^-$.

In the correction phase, the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ is corrected to the posterior state $\underline{\hat{\boldsymbol{x}}}_k$ using an observation $\underline{\boldsymbol{y}}_k$ of the system. This is achieved by adding the Kalman Gain $\boldsymbol{K}$ scaled difference between the estimated $\underline{\hat{\boldsymbol{y}}}_k$ and the actual observation $\underline{\boldsymbol{y}}_k$ to the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$. Here, the estimated observation $\underline{\hat{\boldsymbol{y}}}_k$ is the result of observing the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ through the observation model H, i.e.

$$\underline{\hat{\boldsymbol{y}}}_k = \mathrm{H}\left(\underline{\hat{\boldsymbol{x}}}_k^-\right) \tag{3.2}$$

The Kalman Gain $\boldsymbol{K}$ mentioned above minimizes the difference between the estimated $\underline{\hat{\boldsymbol{y}}}_k$ and actual observation $\underline{\boldsymbol{y}}_k$ by adjusting the estimated state $\underline{\hat{\boldsymbol{x}}}_k$ while taking into account the predicted state covariance $\boldsymbol{P}^-_{\hat{x}\ k}$ and the observation noise $\underline{\boldsymbol{o}}$.

Using the outlined KF framework, the UKF manages to solve discrete-time nonlinear systems with sequential probabilistic inference [41] that can be described by the combination of the process F and observation H models, i.e.,

$$\underline{\boldsymbol{x}}_k = \mathrm{F}\left(\underline{\boldsymbol{x}}_{k-1}, \underline{\boldsymbol{u}}_{k-1}, \underline{\boldsymbol{\nu}}_{k-1}\right) \tag{3.3}$$

$$\underline{\boldsymbol{y}}_k = \mathrm{H}\left(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{o}}_k\right) \tag{3.4}$$

In comparison to its peers, the UKF uses the true nonlinear process F and observation models H, rather than approximating them. However, the UKF approximates the assumed Gaussian distribution of the state and represents this distribution with a minimal set of deterministically sampled points around the state itself. The propagation of this set of sampled points, called sigma-point samples, through the true nonlinear process and subsequent true observation model and following the KF framework will result in an expression of the posterior mean and its covariance. Also, by including the process $\underline{\boldsymbol{\nu}}$ and observation noise $\underline{\boldsymbol{o}}$ into the state their distributions are represented in the sigma-point samples and propagated through the process F and observation model H accordingly.

The deterministically sampling used in the UKF is called Scaled Unscented Transformation (SUT) and is elaborated on in Section 3.1. Following the section on the SUT, and which a central part of the UKF, the algorithm itself will be explained in Section 3.2. In Section 2.3.1 a latency in the observation sensors was mentioned, due to this an extension to the UKF is introduced in Section 3.3 to deal with latency of observation sensors in the system.

3.1. Scaled Unscented Transformation

The SUT is a method for propagating a variable's probability distribution through a nonlinear function. This is achieved by propagating weighted samples, sampled based on the variable's probability distribution, through said function. Hereafter, the variable's propagated probability distribution can be extracted by means of weighted mean and covariance of the propagated weighted samples.

The SUT in particular selects the state $\underline{\boldsymbol{x}}$ itself and two weighted samples $\boldsymbol{\mathcal{X}}$ per dimension L of the state $\underline{\boldsymbol{x}}$. These samples are sampled following the probabilistic spread around the state $\underline{\boldsymbol{x}}$ as indicated by the covariance $\boldsymbol{P}_{\hat{\underline{x}}}$ of said state. The collection of weights $\boldsymbol{w}$ and samples $\boldsymbol{\mathcal{X}}$ are called sigma-points $\mathbb{S} = \{\boldsymbol{w}, \boldsymbol{\mathcal{X}}\}$. Moreover, the overall dimension of the sigma-point sample matrix equals to $(2L+1, L)$. In order to compute weights $\boldsymbol{w}$ and control the probabilistic spread of the samples $\boldsymbol{\mathcal{X}}$ around the state $\underline{\boldsymbol{x}}$ three new parameters are introduced:

α: Controls the 'distance' between the sigma-points and the state best estimate $\hat{\underline{\boldsymbol{x}}}$,
Should be $0 \leq \alpha \leq 1$.

β: Controls the influence of the higher order moments of the probability distribution,
Should be $\beta \geq 0$.

κ: Should be $\kappa \geq 0$ to guarantee positive semi-definiteness of the state covariance matrix $\boldsymbol{P}_{\hat{\underline{x}}}$ and provide computational robustness. However, exact value of κ is not critical for the performance of the algorithm.

Two sets of weights are needed to deduce the probability distribution from the sigma-point samples $\boldsymbol{\mathcal{X}}$, $\boldsymbol{w}^{(m)}$ for computing the weighted mean and $\boldsymbol{w}^{(c)}$ for deducing the weighted covariance. The weights $\boldsymbol{w}^{(m)}$ and $\boldsymbol{w}^{(c)}$ are derived in Equations 3.6, 3.7 and 3.8. An interim constant ζ, in Equation 3.5, is computed to simplify Equations 3.6 to 3.8.

$$\zeta = \alpha^2 (L + \kappa) - L \tag{3.5}$$

$$\boldsymbol{w}_0^{(m)} = \frac{\zeta}{L + \zeta} \tag{3.6}$$

$$\boldsymbol{w}_0^{(c)} = \frac{\zeta}{L + \zeta} + (1 - \alpha^2 + \beta) \tag{3.7}$$

$$\boldsymbol{w}_i^{(m)} = \boldsymbol{w}_i^{(c)} = \frac{1}{2(L + \zeta)} \quad i = 1, \dots, 2L \tag{3.8}$$

Here, the zeroth weight has the largest influence. The zeroth sigma-point sample represent the current state of the system and thus the best estimate at that point in time. Furthermore, the weights $\boldsymbol{w}^{(m)}$ and $\boldsymbol{w}^{(c)}$ are constant throughout the algorithm as long as the dimension of the state L does not change.

Concurrent to creating both sets of the sigma-point weights, $\boldsymbol{w}^{(m)}$ and $\boldsymbol{w}^{(c)}$, the sigma-point samples are computed from the estimated state $\hat{\underline{\boldsymbol{x}}}$. The zeroth sigma-point sample is the estimated state $\hat{\underline{\boldsymbol{x}}}$ itself. The other sigma-point samples are computed by adding or subtracting the estimated state $\hat{\underline{\boldsymbol{x}}}$ with the ith column of the square root of the weighted state covariance $\left(\sqrt{(L + \zeta)\, \boldsymbol{P}_{\hat{\underline{x}}}}\right)_i$, as per Equation 3.9. Furthermore, the Cholesky method [10] is an efficient way to solve for the square root of the weighted covariance matrix $\left(\sqrt{(L + \zeta)\, \boldsymbol{P}_{\hat{\underline{x}}}}\right)$.

$$\boldsymbol{\mathcal{X}} = \left[\hat{\underline{\boldsymbol{x}}} \quad \hat{\underline{\boldsymbol{x}}} + \left(\sqrt{(L + \zeta)\, \boldsymbol{P}_{\hat{\underline{x}}}}\right)_i \quad \hat{\underline{\boldsymbol{x}}} - \left(\sqrt{(L + \zeta)\, \boldsymbol{P}_{\hat{\underline{x}}}}\right)_i\right] \quad i = 0 \dots L - 1 \tag{3.9}$$

Here, Equation 3.9 results in a $(L, 2L+1)$ sized matrix. The non-zeroth sigma-point samples, columns $(1, \dots, 2L)$, are spread around the estimated state $\hat{\underline{\boldsymbol{x}}}$, approximating the probability distribution $\boldsymbol{P}_{\hat{x}}$ with a minimal set of sigma-point samples.

3.2. Unscented Kalman Filter Algorithm

In this section the complete UKF will be explained, with the SUT, explained in Section 3.1, at its core. First, the augmentation of the state and its covariance with the noise random variables will be discussed in Section 3.2.1. Second, the initialization of the algorithm will be expanded on in Section 3.2.2. After, the two different phases of the algorithm will be discussed. Starting with the prediction phase, in which the process model F and a possible input $\underline{\boldsymbol{u}}$ are used to predict the next state $\underline{\boldsymbol{x}}$, as further discussed in Section 3.2.3. While, the correction phase employs the observation model H and an observation $\underline{\boldsymbol{y}}$ to correct the state estimate, as per Section 3.2.4.

3.2.1. Augmentation of State and Covariance

One of the advantages of the UKF is the ability to deal with the influence of Gaussian noise on the sensors according to the true process and observation system models. The UKF does so by augmenting the state $\underline{\boldsymbol{x}}$ with the process $\underline{\boldsymbol{\nu}}$ and observation noise $\underline{\boldsymbol{o}}$ resulting in the augmented state $\underline{\boldsymbol{x}}^a$, i.e.,

$$\underline{\boldsymbol{x}}^a = \begin{bmatrix}\underline{\boldsymbol{x}}^T & \underline{\boldsymbol{\nu}}^T & \underline{\boldsymbol{o}}^T\end{bmatrix}^T \tag{3.10}$$

It must be noted that the augmentation of the state increases the dimension of the state to $L = L^x + L^v + L^o$, with L^x the dimension of the state, L^v the dimension of the process noise, and L^o the dimension of the observation noise. The addition of the noise random variables, $\underline{\boldsymbol{\nu}}$ and $\underline{\boldsymbol{o}}$, leads to an increase in computational cost at the gain of the precision of introducing the noise random variables according to their true respective model.

Likewise, the state covariance is also augmented with the covariances of the process and observation noise random variables, $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively, resulting in the augmented state covariance $\boldsymbol{P}^a_{\hat{x}}$, i.e.,

$$\boldsymbol{P}^a_{\hat{x}} = \begin{bmatrix}\boldsymbol{P}_{\hat{x}} & 0 & 0 \\ 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \boldsymbol{R}\end{bmatrix} \tag{3.11}$$

The uncertainty of both process and observation noise, by augmenting them to the state and covariance, are captured in the sigma-points $\mathbb{S}$ and can thus also be propagated in the true process and observation models in the same manner as the uncertainty on the state.

The expected value, indicated by $E\left|...\right|$, of both the process and observation noise random variables, $\underline{\boldsymbol{\nu}}$ and $\underline{\boldsymbol{o}}$ respectively, are equal to zero over time t, and characterized as they are both modelled as zero-mean Gaussian noise. Their respective autocovariance are equal to $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively by definition, i.e.,

$$E\left|\underline{\boldsymbol{\nu}}(t)\right| = 0 \qquad E\left|\underline{\boldsymbol{\nu}}(t)\underline{\boldsymbol{\nu}}^T(\tau)\right| = \boldsymbol{Q} \quad \rightarrow \quad \underline{\boldsymbol{\nu}}(t) \approx \aleph\left(0, \boldsymbol{Q}\right) \tag{3.12}$$

$$E\left|\underline{\boldsymbol{o}}(t)\right| = 0 \qquad E\left|\underline{\boldsymbol{o}}(t)\underline{\boldsymbol{o}}^T(\tau)\right| = \boldsymbol{R} \quad \rightarrow \quad \underline{\boldsymbol{o}}(t) \approx \aleph\left(0, \boldsymbol{R}\right) \tag{3.13}$$

3.2.2. Initialization

At initialization, the zeroth time step indicated by subscript 0, the expected state and its expected covariance are used for the initial state estimate $\hat{\underline{\boldsymbol{x}}}_0$ and its covariance $\boldsymbol{P}_{\hat{x}_0}$, i.e.,

$$\hat{\underline{\boldsymbol{x}}}_0 = E\left|\underline{\boldsymbol{x}}_0\right| \tag{3.14}$$

$$\boldsymbol{P}_{\hat{x}_0} = E\left|[\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0][\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0]^T\right| \tag{3.15}$$

Combining the information from Equations 3.10 to 3.15, the initial expected augmented state $\hat{\underline{\boldsymbol{x}}}^a_0$ and its covariance $\boldsymbol{P}^a_{\hat{x}\,0}$ can be written as,

$$\begin{aligned}\underline{\hat{\boldsymbol{x}}}_0^a &= E\left|\underline{\boldsymbol{x}}_0^a\right| = E\left|\left[\underline{\boldsymbol{x}}_0^T \quad \underline{\boldsymbol{\nu}}^T \quad \underline{\boldsymbol{o}}^T\right]^T\right| = \left[\underline{\hat{\boldsymbol{x}}}_0^T \quad 0 \quad 0\right]^T \\ \boldsymbol{P}^a_{\hat{x}\,0} &= E\left|\left[\underline{\boldsymbol{x}}_0^a - \underline{\hat{\boldsymbol{x}}}_0^a\right]\left[\underline{\boldsymbol{x}}_0^a - \underline{\hat{\boldsymbol{x}}}_0^a\right]^T\right| \qquad = \begin{bmatrix} \boldsymbol{P}_{\hat{x}_0} & 0 & 0 \\ 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \boldsymbol{R} \end{bmatrix}\end{aligned} \tag{3.16}$$

In parallel, the sigma-point weights, $\boldsymbol{w}^{(m)}$ and $\boldsymbol{w}^{(c)}$, can be computed, using Equations 3.6 to 3.8, with the augmented state's dimension L known.

3.2.3. Prediction

In the prediction phase, the process model F in combination with a possible input $\underline{\boldsymbol{u}}$ are used to project the previous best state estimate $\underline{\hat{\boldsymbol{x}}}_{k-1}$ and its covariance $\boldsymbol{P}_{\hat{x}_{k-1}}$ over one time step to the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ and its covariance $\boldsymbol{P}^-_{\hat{x}\,k}$ at time step k, the $-$ superscript indicating the predicted status. In order to do so, new sigma-point samples $\boldsymbol{\mathcal{X}}^a_{k-1}$ are drawn around and including the augmented previous best state $\underline{\hat{\boldsymbol{x}}}^a_{k-1}$ using its covariance $\boldsymbol{P}^a_{\hat{x}\,k-1}$ in the SUT, as described in Section 3.1 in Equation 3.9 resulting in,

$$\boldsymbol{\mathcal{X}}^a_{k-1} = \left[\underline{\hat{\boldsymbol{x}}}^a_{k-1} \quad \underline{\hat{\boldsymbol{x}}}^a_{k-1} + \left(\sqrt{(L+\zeta)\,\boldsymbol{P}^a_{\hat{x}\,k-1}}\right)_i \quad \underline{\hat{\boldsymbol{x}}}^a_{k-1} - \left(\sqrt{(L+\zeta)\,\boldsymbol{P}^a_{\hat{x}\,k-1}}\right)_i\right] \qquad i = 0 \ldots L-1 \tag{3.17}$$

As in Equation 3.9, the subscript i indicates the column.
The rows of the sigma-points that represent the state will be indicated as $\boldsymbol{\mathcal{X}}^x$, the process noise $\boldsymbol{\mathcal{X}}^v$, and the observation noise $\boldsymbol{\mathcal{X}}^o$, i.e.,

$$\boldsymbol{\mathcal{X}}^a_{k-1} = \begin{bmatrix} \boldsymbol{\mathcal{X}}^x_{k-1} \\ \boldsymbol{\mathcal{X}}^v_{k-1} \\ \boldsymbol{\mathcal{X}}^o_{k-1} \end{bmatrix} \tag{3.18}$$

This results in the matrix size of the state part of $(L^x, 2L+1)$, the process noise part of $(L^v, 2L+1)$, and the observation noise part of $(L^o, 2L+1)$. The process model F in combination with a possible input $\underline{\boldsymbol{u}}$ and the influence of the noise, in the form of the process noise sigma-point samples $\boldsymbol{\mathcal{X}}^v_{k-1}$, are then utilized to project the state part of the previous sigma-points samples $\boldsymbol{\mathcal{X}}^x_{k-1}$ to the next time step, i.e. the predicted state sigma-point samples $\boldsymbol{\mathcal{X}}^{x,-}_k$, i.e.,

$$\boldsymbol{\mathcal{X}}^{x,-}_k = \mathrm{F}\left(\boldsymbol{\mathcal{X}}^x_{k-1}, \boldsymbol{\mathcal{X}}^v_{k-1}, \underline{\boldsymbol{u}}_{k-1}\right) \tag{3.19}$$

From the predicted state sigma-point samples $\boldsymbol{\mathcal{X}}^{x,-}_k$ the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ and its covariance $\boldsymbol{P}^-_{\hat{x}_k}$ are computed by means of weighted means and weighted variance using the sigma-point weights $\boldsymbol{w}^{(m)}$ and $\boldsymbol{w}^{(c)}$ respectively, i.e.,

$$\underline{\hat{\boldsymbol{x}}}_k^- = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(m)} \boldsymbol{\mathcal{X}}^{x,-}_{k,i} \tag{3.20}$$

$$\boldsymbol{P}^-_{\hat{x}\,k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left[\boldsymbol{\mathcal{X}}^{x,-}_{k,i} - \underline{\hat{\boldsymbol{x}}}_k^-\right]\left[\boldsymbol{\mathcal{X}}^{x,-}_{k,i} - \underline{\hat{\boldsymbol{x}}}_k^-\right]^T \tag{3.21}$$

Here, the subscript i indicates the column.

3.2.4. Correction

In the correction phase, the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ and its covariance $\boldsymbol{P}^-_{\hat{x}\,k}$ are corrected to the posterior state $\underline{\hat{\boldsymbol{x}}}_k$ and its covariance $\boldsymbol{P}_{\hat{x}_k}$. This is achieved by, at the arrival of a measurement $\underline{\boldsymbol{y}}_k$, projecting the predicted state sigma-points samples $\boldsymbol{\mathcal{X}}^{x,-}_k$ through the observation model alongside the observation noise sigma-point samples $\boldsymbol{\mathcal{X}}^o$ into the estimated observation points $\boldsymbol{\mathcal{Y}}_k$, i.e.,

$$\boldsymbol{\mathcal{Y}}_k = \mathrm{H}(\boldsymbol{\mathcal{X}}^{x,-}_k, \boldsymbol{\mathcal{X}}^o_k) \tag{3.22}$$

Similar to the prediction phase, the estimated observation $\underline{\hat{\boldsymbol{y}}}_k$ and its covariance $\boldsymbol{P}_{\hat{y}_k}$ are determined by taking the weighted means and weighted variance of the propagated observation points $\boldsymbol{\mathcal{Y}}_k$ using the sigma-point weights $\boldsymbol{w}^{(m)}$ and $\boldsymbol{w}^{(c)}$ respectively, i.e.,

$$\underline{\hat{\boldsymbol{y}}}_k = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(m)} \boldsymbol{\mathcal{Y}}_k \tag{3.23}$$

$$\boldsymbol{P}_{\hat{y}_k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left[\boldsymbol{\mathcal{Y}}_{k,i} - \underline{\hat{\boldsymbol{y}}}_k\right] \left[\boldsymbol{\mathcal{Y}}_{k,i} - \underline{\hat{\boldsymbol{y}}}_k\right]^T \tag{3.24}$$

Also, the weighted cross-covariance $\boldsymbol{P}_{\hat{x}_k \hat{y}_k}$ between the predicted state $\underline{\hat{\boldsymbol{x}}}_k$ and the estimated observation points $\underline{\hat{\boldsymbol{y}}}_k$ can be computed as:

$$\boldsymbol{P}_{\hat{x}_k \hat{y}_k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left[\boldsymbol{\mathcal{X}}_{k,i}^{x,-} - \underline{\hat{\boldsymbol{x}}}_k^-\right] \left[\boldsymbol{\mathcal{Y}}_k - \underline{\hat{\boldsymbol{y}}}_k\right]^T \tag{3.25}$$

Here, the subscript i indicates the column. Both the estimated observation $\underline{\hat{\boldsymbol{y}}}_k$ and its covariance $\boldsymbol{P}_{\hat{y}_k}$ are required to correct the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ and its covariance $\boldsymbol{P}_{\hat{x}_k}^-$.

The Kalman gain is defined to minimize the error optimally between the predicted state propagated through the observation model by changing the predicted state based on the ratio between the predicted estimated state's covariance and the observation covariance. The Kalman gain can be computed by dividing the cross-covariance $\boldsymbol{P}_{\hat{x}_k \hat{y}_k}$, found in Equation 3.25 by the observation covariance $\boldsymbol{P}_{\hat{y}_k}$, shown in Equation 3.24, i.e.,

$$\boldsymbol{K}_k = \boldsymbol{P}_{\hat{x}_k \hat{y}_k} \boldsymbol{P}_{\hat{y}_k}^{-1} \tag{3.26}$$

Finally, scaling the the innovation $\underline{\boldsymbol{\epsilon}}_k$, meaning the difference between the actual observation $\underline{\boldsymbol{y}}_k$ and the estimated observation $\underline{\hat{\boldsymbol{y}}}_k$, with the Kalman gain $\boldsymbol{K}_k$ and adding the result to the predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ results in the posterior state $\underline{\hat{\boldsymbol{x}}}_k$, i.e.,

$$\begin{aligned} \underline{\hat{\boldsymbol{x}}}_k &= \underline{\hat{\boldsymbol{x}}}_k^- + \boldsymbol{K}_k \left[\underline{\boldsymbol{y}}_k - \underline{\hat{\boldsymbol{y}}}_k\right] \\ &= \underline{\hat{\boldsymbol{x}}}_k^- + \boldsymbol{K}_k \underline{\boldsymbol{\epsilon}}_k \end{aligned} \tag{3.27}$$

The posterior covariance $\boldsymbol{P}_{\hat{x}_k}$ on the other hand, is computed by subtracting the predicted state covariance $\boldsymbol{P}^-_{\hat{x}\,k}$ the observation covariance $\boldsymbol{P}_{\hat{y}_k}$ scaled with the Kalman gain $\boldsymbol{K}$, found in Equation 3.26, i.e.,

$$\boldsymbol{P}_{\hat{x}_k} = \boldsymbol{P}^-_{\hat{x}\,k} - \boldsymbol{K}_k \boldsymbol{P}_{\hat{y}_k} \boldsymbol{K}_k^T \tag{3.28}$$

From examining Equations 3.27 and 3.28, in which it was shown how to compute the posterior state $\underline{\hat{\boldsymbol{x}}}_k$ and its covariance $\boldsymbol{P}_{\hat{x}_k}$, the role of the Kalman gain $\boldsymbol{K}$ becomes more apparent. As a relative large Kalman gain $\boldsymbol{K}$ will favor the observation model, or in other words the correction phase, while a relative small Kalman gain $\boldsymbol{K}$ will favor the process model or, in other words, the prediction phase. The Kalman gain $\boldsymbol{K}$ itself scales with the cross-covariance $\boldsymbol{P}_{\hat{x}_k \hat{y}_k}$ and inversely with the observation covariance $\boldsymbol{P}_{\hat{y}_k}$. These two terms themselves both positively scale with the process $\boldsymbol{Q}$ and observation $\boldsymbol{R}$ noise distributions respectively after being propagated through the process F and observation H models. The takeaway for following chapters is that the process $\boldsymbol{Q}$ and observation $\boldsymbol{R}$ noise distributions could be employed to tune the UKF.

3.3. Latency Compensation

In this section a method to optimally integrate the delayed sensor into the KF-framework will be explained. In the next section, proof will be given why this method is optimal in the KF-framework and implemented into the UKF.

A sensor's latency is defined when the current sensor output actually corresponds to the state of the system at some point in the past, as depicted in Figure 3.1. The latency of an observation sensor, the measurement of the sensor used in the correction phase, is defined as the measurement $\underline{y}_{l \to k}$ taken at time-step l but only presented to the system at time-step k, hence the subscript $l \to k$.

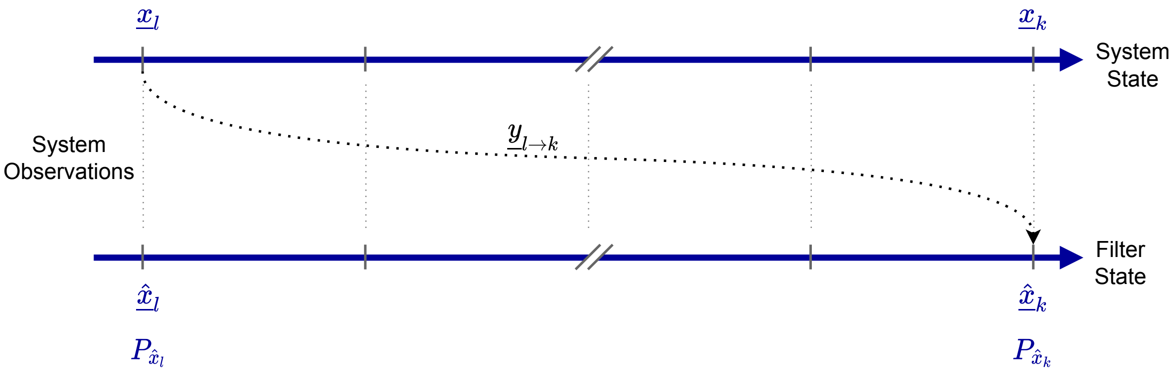


Figure 3.1: Latency on the observation $\underline{y}_{l \to k}$.

The latency of an observation sensor could be an issue when designing state estimators, such as the UKF. When not accounting for the latency of an observation sensor the KF-framework will relate and correct the current predicted state $\hat{\underline{x}}_k^-$ with the delayed observation $\underline{y}_{l \to k}$. Correcting the current predicted state $\hat{\underline{x}}_k^-$ towards the measured state $\underline{x}_l$ at time l. The error of not taking the latency into consideration scales with the change in the state $\underline{x}$ over the latency time period in relation to the process $\boldsymbol{Q}$ noise or in other words the precision of the delayed sensor.

Multiple methods for dealing with observational latency have been proposed [2, 5, 33, 41]. By far the easiest method would be to simply ignore the latency of the sensor. This method could work if it is the case that the latency on the sensor was relative small to the motion of the system and the precision of that sensor was already lacking. More precisely, in this case the system would not move much during the span of the latency when compared to the precision, the impact of the latency could be ignored. This case could simply be solved by increasing the observation noise distribution $\boldsymbol{R}$. However, the observation noise distribution $\boldsymbol{R}$ would no longer be zero-mean, but the error incurred would be probably rather small to the precision of the rest of the system. If this is not the case, crucial information would be lost with this method.

3.3.1. Smoothed Delayed Sensor Fusion

The approach used in this Thesis is the UKF implementation of the Smoothed Delayed Sensor Fusion (SDSF) as found in [41] and builds upon Larsen's method [33]. The core idea of the algorithm extension will be explained in this section. In the next section, Section 3.3.2, the SDSF will be implemented on the UKF algorithm itself.

The SDSF corrects the state optimally based on delayed observation given the knowledge on the latency of the observation. However, the SDSF does not estimate the delay on the observation. The core idea of this algorithm is to extend the estimated state and its covariance with a copy of the itself corresponding to the expected time-step of the delayed measurement, i.e. time-step l, also depicted in Figure 3.2. The copy of the state and its covariance is made at the end of the time-step and is highlighted in green in Figure 3.2. Moreover, the copy of the state, i.e. the past estimated state $\hat{\underline{x}}_l$, is left **unaltered** by the prediction phase, see time-steps $l+1$ and $k-1$ in Figure 3.2. By extension of the state being unaltered its covariance $\boldsymbol{P}_{\hat{x}_l}$ will also be unchanged by the prediction phase. Meanwhile, the cross-covariance terms between the current and past estimated state $\boldsymbol{P}_{\hat{x}_k \hat{x}_l}$, $\boldsymbol{P}_{\hat{x}_l \hat{x}_k}$ are maintained alongside the current estimated state $\hat{\underline{x}}_k$ and its covariance $\boldsymbol{P}_{\hat{x}_k}$.

At the time of arrival of the delayed observation $\underline{y}_{l \to k}$ at time-step k, both the past $\hat{\underline{x}}_l$ and current estimated state $\hat{\underline{x}}_k$, alongside all their covariances, are corrected. The past estimated state $\hat{\underline{x}}_l$ is corrected in the expected KF-framework manner, the only difference being that all elements involved, i.e. state, observation, covariance, are of the same time-step. Every correction of a past estimated state is indicated by a superscript $+$.

The current estimated state $\hat{\underline{\boldsymbol{x}}}_k$ is corrected by utilizing the maintained cross-covariance between the current and past estimated state to transfer the correction from the past estimated state $\hat{\underline{\boldsymbol{x}}}_l$ to the current estimated state $\hat{\underline{\boldsymbol{x}}}_k$. If the corrected past estimated state is no longer required, i.e. no delayed sensor's observation is still inbound that was measured at the past estimated state's time-step, said past estimated state and respective covariances can be removed. In Figure 3.2, the to-be removed state is highlighted in red and crossed out at the respective time-step.

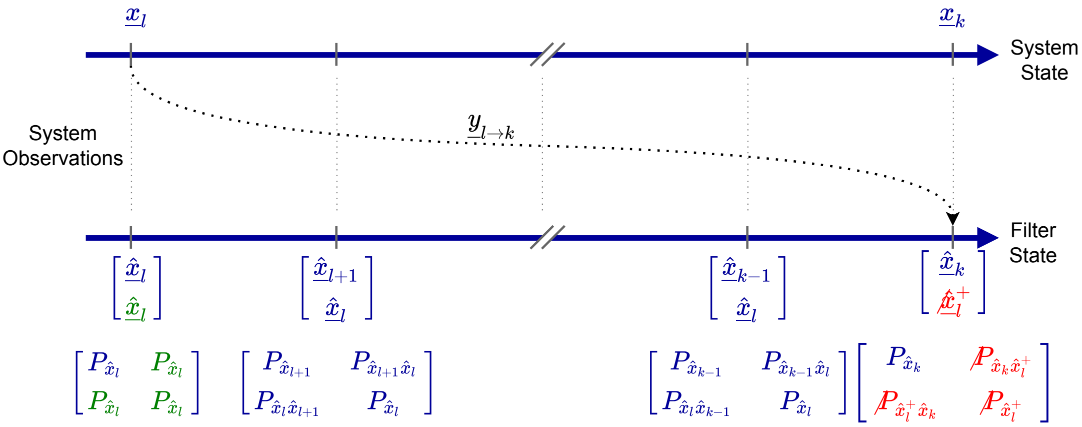

$\underline{x}_l$
$\underline{x}_k$
System State
System Observations
$\underline{y}_{l\to k}$
Filter State
$\begin{bmatrix}\hat{\underline{x}}_l\\ \hat{\underline{x}}_l\end{bmatrix}$
$\begin{bmatrix}\hat{\underline{x}}_{l+1}\\ \hat{\underline{x}}_l\end{bmatrix}$
$\begin{bmatrix}\hat{\underline{x}}_{k-1}\\ \hat{\underline{x}}_l\end{bmatrix}$
$\begin{bmatrix}\hat{\underline{x}}_k\\ \cancel{\hat{\underline{x}}_l^+}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_l} & P_{\hat{x}_l}\\ P_{\hat{x}_l} & P_{\hat{x}_l}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_{l+1}} & P_{\hat{x}_{l+1}\hat{x}_l}\\ P_{\hat{x}_l\hat{x}_{l+1}} & P_{\hat{x}_l}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_{k-1}} & P_{\hat{x}_{k-1}\hat{x}_l}\\ P_{\hat{x}_l\hat{x}_{k-1}} & P_{\hat{x}_l}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_k} & \cancel{P}_{\hat{x}_k\hat{x}_l^+}\\ \cancel{P}_{\hat{x}_l^+\hat{x}_k} & \cancel{P}_{\hat{x}_l^+}\end{bmatrix}$

Figure 3.2: The estimated state $\hat{\underline{\boldsymbol{x}}}_l$ and its covariance $\boldsymbol{P}_{\hat{x}_l}$ are copied at time-step l. At the arrival of the delayed observation $\underline{\boldsymbol{y}}_{l\to k}$, the copied estimated state $\hat{\underline{\boldsymbol{x}}}_l$ and its covariance $\boldsymbol{P}_{\hat{x}_l}$ are used to correct both the past and current estimated state.
Green to signal the copy of the state at the end of the time-step.
Red and $\cancel{x}$ to signal the removal of the past estimated state before the end of the time-step.

Moreover, at the arrival of a non-delayed observation, similar to arrival of a delayed observation, both the current and the past estimated state are corrected. Meaning, the current estimated state is used to improve past estimated state(s) based on subsequent future observation. This causes the past estimated state to be smoothed during the latency period by non-delayed observations [18]. In this case, the current estimated state is corrected in the standard KF-framework manner, all elements, i.e. state, covariance, and observation, are of the same time-step k. However, the past estimated state is corrected by using the maintained cross-covariance between the current and past estimated state to transfer the correction from the current estimated state $\hat{\underline{\boldsymbol{x}}}_k$ to the current estimated state $\hat{\underline{\boldsymbol{x}}}_l$.

This event, the arrival of a non-delayed observation, is also portrait in Figure 3.3. Here a non-delayed observation $\underline{\boldsymbol{y}}_n$ at time-step n corrects both the current as well as the past estimated state. The now improved past estimated state is indicated by the superscript $+$.

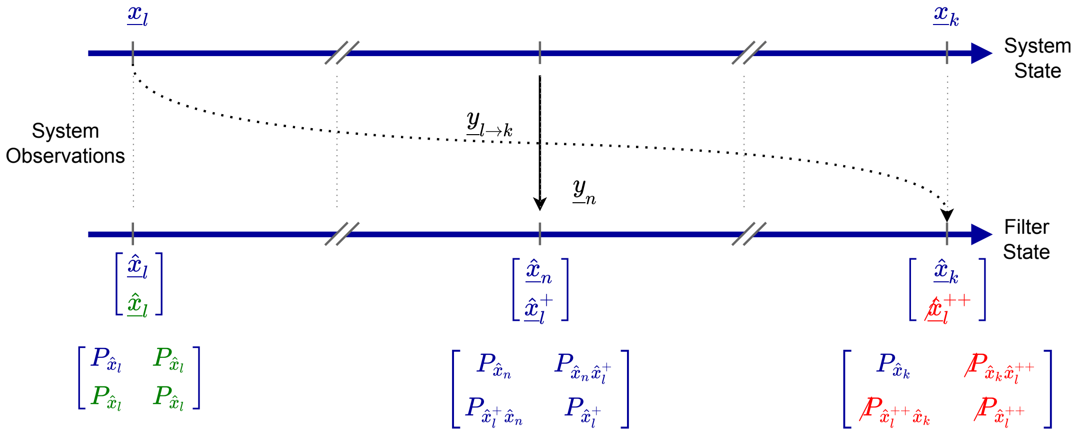

$\underline{x}_l$
$\underline{x}_k$
System State
System Observations
$\underline{y}_{l\to k}$
$\underline{y}_n$
Filter State
$\begin{bmatrix}\hat{\underline{x}}_l \\ \hat{\underline{x}}_l\end{bmatrix}$
$\begin{bmatrix}\hat{\underline{x}}_n \\ \hat{\underline{x}}_l^+\end{bmatrix}$
$\begin{bmatrix}\hat{\underline{x}}_k \\ \cancel{\hat{\underline{x}}}_l^{++}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_l} & P_{\hat{x}_l} \\ P_{\hat{x}_l} & P_{\hat{x}_l}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_n} & P_{\hat{x}_n\hat{x}_l^+} \\ P_{\hat{x}_l^+\hat{x}_n} & P_{\hat{x}_l^+}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_k} & \cancel{P}_{\hat{x}_k\hat{x}_l^{++}} \\ \cancel{P}_{\hat{x}_l^{++}\hat{x}_k} & \cancel{P}_{\hat{x}_l^{++}}\end{bmatrix}$

Figure 3.3: The current and past estimated state are corrected by a future non-delayed observation $\underline{y}_{l+n}$. The correction on the past estimated state $\hat{\underline{x}}_l$ is indicated by the superscript $+$.
Green to signal the copy of the state at the end of the time-step.
Red and $\cancel{x}$ to signal the removal of the past estimated state before the end of the time-step.

To clarify, there can only be one current state, but it is possible to hold more than one past estimated state. This is the case if the latency of the sensor is larger than its time step duration or there exist multiple sensors with a delay. The maximum amount of past estimated states M required per delayed sensor at a time after the first latency period depends on the sensor's frequency and its latency. Here the sensor's frequency and latency are expressed as the amount of discrete time-steps between measurements, Δt and d respectively, i.e.,

$$M = \frac{d}{\Delta t} \tag{3.29}$$

The amount of past estimated states M per sensor is reduced if the physical measurements happen to overlap time-step wise.

For example, depicted in Figure 3.4 is a sensor with a latency of two time-steps measuring every time-step. The system needs two past estimated states after the first latency period. In this figure at certain time-steps three past estimated states are depicted, but it must be noted that the red crossed out state is removed at the same time the current state is copied (highlighted in green). This ensures that the maximum number of past estimated states in this example is two.

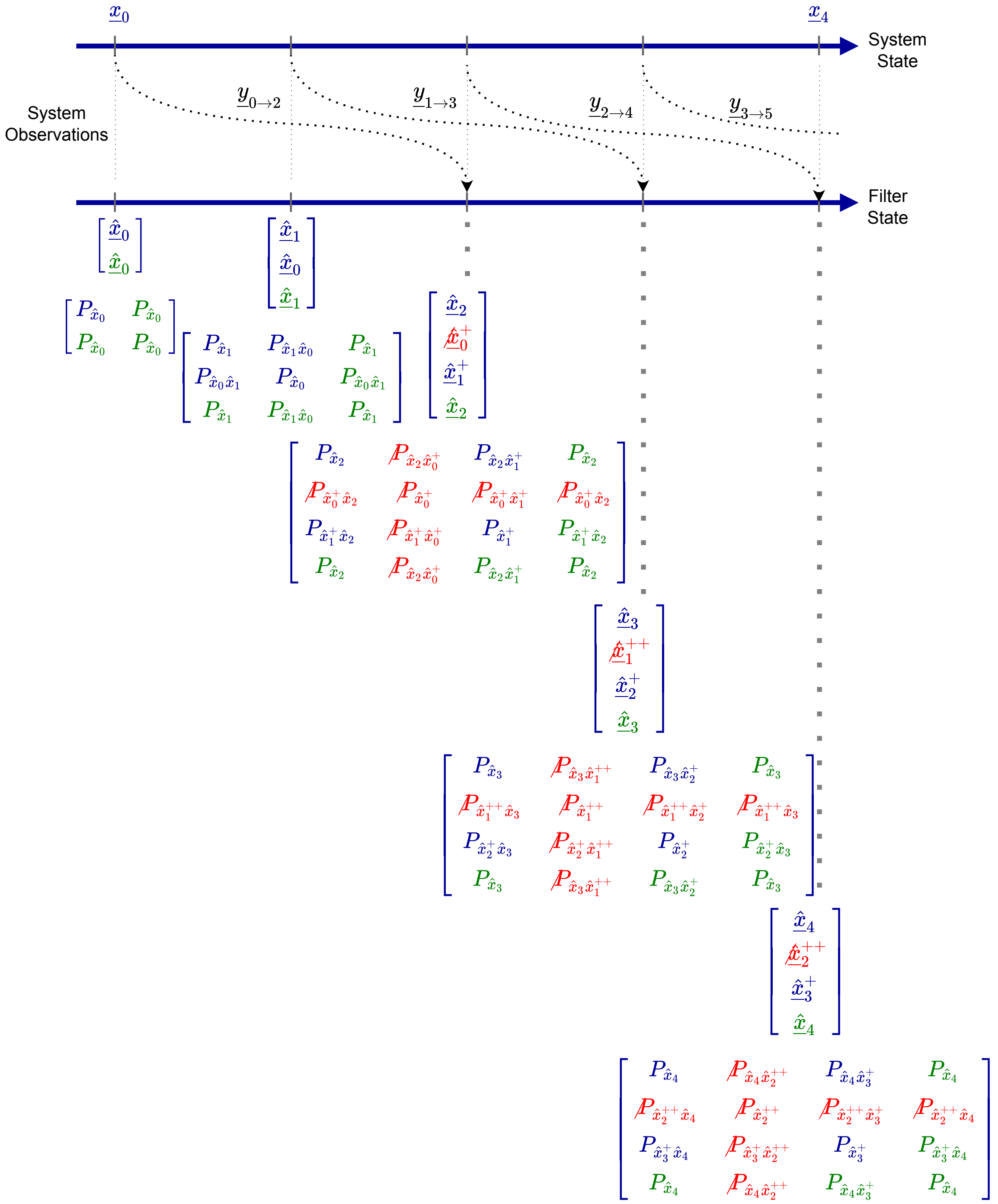

$\underline{x}_0$
$\underline{x}_4$
System State
$\underline{y}_{0\rightarrow 2}$
$\underline{y}_{1\rightarrow 3}$
$\underline{y}_{2\rightarrow 4}$
$\underline{y}_{3\rightarrow 5}$
System Observations
Filter State

Figure 3.4: An example of a sensor with a latency of two time-steps measuring every time-step needing two past estimated states and their covariances after the first latency period.
Green to signal the copy of the state at the end of the time-step.
Red and $\not{x}$ to signal the removal of the past estimated state before the end of the time-step.
Due to space restrictions the state and respective covariances are placed staggered. The vertical dotted line connects the state to the correct time-step.

Also interesting in this example is that each delayed observation corrects the next past estimated state, i.e. delayed observation $\hat{\underline{y}}_{0\rightarrow 2}$ corrects the past estimated state $\hat{\underline{x}}_1$ at time-step 2, and so on.

The opposite can also happen, where the delay period d is shorter than the time between measurements Δt, depicted in Figure 3.5. In this figure a sensor has a latency of one time-step and measures every three time-steps. This configuration results in needing a past estimated state every three time-steps for the duration of one time-step.

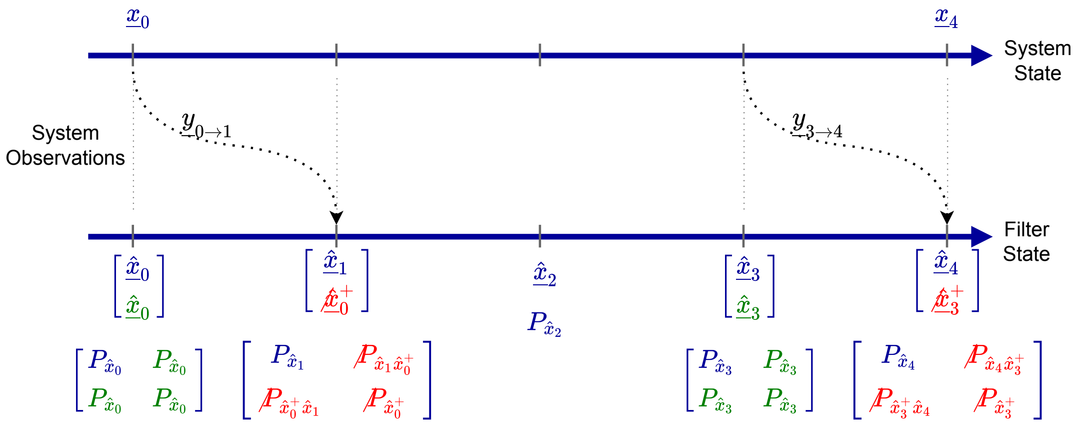

$\underline{x}_0$
$\underline{x}_4$
System State
System Observations
$\underline{y}_{0\rightarrow 1}$
$\underline{y}_{3\rightarrow 4}$
Filter State
$\begin{bmatrix}\hat{\underline{x}}_0\\ \hat{\underline{x}}_0\end{bmatrix}$
$\begin{bmatrix}\hat{\underline{x}}_1\\ \cancel{\hat{\underline{x}}}_0^+\end{bmatrix}$
$\hat{\underline{x}}_2$
$P_{\hat{x}_2}$
$\begin{bmatrix}\hat{\underline{x}}_3\\ \hat{\underline{x}}_3\end{bmatrix}$
$\begin{bmatrix}\hat{\underline{x}}_4\\ \cancel{\hat{\underline{x}}}_3^+\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_0} & P_{\hat{x}_0}\\ P_{\hat{x}_0} & P_{\hat{x}_0}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_1} & \cancel{P}_{\hat{x}_1\hat{x}_0^+}\\ \cancel{P}_{\hat{x}_0^+\hat{x}_1} & \cancel{P}_{\hat{x}_0^+}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_3} & P_{\hat{x}_3}\\ P_{\hat{x}_3} & P_{\hat{x}_3}\end{bmatrix}$
$\begin{bmatrix}P_{\hat{x}_4} & \cancel{P}_{\hat{x}_4\hat{x}_3^+}\\ \cancel{P}_{\hat{x}_3^+\hat{x}_4} & \cancel{P}_{\hat{x}_3^+}\end{bmatrix}$

Figure 3.5: An example of a sensor with a latency of one time-step measuring every three time-steps. Here, the system needs one past state every three time-steps.
Green to signal the copy of the state at the end of the time-step.
Red and $\cancel{x}$ to signal the removal of the past estimated state before the end of the time-step.

To end this section the rules of SDSF can be summed up as follows:

- **Prediction Phase**
  - Only predict the next current state, no alteration of past estimated state(s).
- **Correction Phase**
  - Correct all states, including past estimated state(s).
  - At the end of this phase:
    If a past estimated state is no longer required it can be removed.
- **Every Iteration**:
  - If any delayed sensor is expected to make a physical observation, copy current estimated state before starting the next iteration.

In the correction phase some optimizations can be applied. If the correction is based on a delayed observation and the past estimated state relevant to the delayed observation would be no longer required after this correction phase for an even longer delayed sensor, the update of the relevant past estimated state can be skipped. However, the skipped corrected past estimated state $\hat{\underline{x}}_l^+$ is the best estimate of the system at time-step l, having fused all the measurements and could be of note for logging purposes for example.

3.3.2. Smoothed Delayed Unscented Kalman Filter

In the previous section the rules were explained for copying, removing and altering the past estimated state and its covariance. In this section, the rule-set is implemented into the UKF resulting in the Smoothed Delayed Unscented Kalman Filter (SDUKF). First, the reason is given for copying the past estimated state from an optimal Kalman gain perspective. Next, the equation of the UKF in Section 3.2 are re-examined in order to to deal with the multiple states in different time-steps.

Optimal Kalman Gain

The reason for copying the estimated state and maintaining cross-covariance between the current and past estimated states can be traced back to the Kalman gain $\boldsymbol{K}$ required to correct the current predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ with a delayed observation $\underline{\boldsymbol{y}}_{l\to k}$, i.e.,

$$\underline{\hat{\boldsymbol{x}}}_k = \underline{\hat{\boldsymbol{x}}}_k^- + \boldsymbol{K}_{l\to k}\left[\underline{\boldsymbol{y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right] \tag{3.30}$$

In this case, the Kalman gain $\boldsymbol{K}_{l\to k}$ represents the optimal correction of the current predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ based on the delayed estimated observation $\underline{\hat{\boldsymbol{y}}}_{l\to k}$. Where, this Kalman gain $\boldsymbol{K}_{l\to k}$ can be computed as,

$$\boldsymbol{K}_{l\to k} = \boldsymbol{P}_{\hat{x}_k\hat{y}_{l\to k}}\boldsymbol{P}_{\hat{y}_{l\to k}}^{-1} \tag{3.31}$$

Here, the delayed estimated observation's covariance $\boldsymbol{P}_{\hat{y}_{l\to k}}$ relates the certainty of the observation. So when multiplied with the delayed estimated innovation $\left[\underline{\boldsymbol{y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right]$ scales the error appropriately. The cross-covariance $\boldsymbol{P}_{\hat{x}_k\hat{y}_{l\to k}}$ holds the relation between the current estimated state $\underline{\hat{\boldsymbol{x}}}_k$ and the past estimated observation $\underline{\hat{\boldsymbol{y}}}_{l\to k}$. Moreover, the cross-covariance $\boldsymbol{P}_{\hat{x}_k\hat{y}_{l\to k}}$ translates the scaled error to the current predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ in Equation 3.30.

Extended State

In order to compute the delayed estimated observation $\underline{\hat{\boldsymbol{y}}}_{l\to k}$, its covariance $\boldsymbol{P}_{\hat{y}_{l\to k}}$ and the cross-covariance $\boldsymbol{P}_{\hat{x}_k\hat{y}_{l\to k}}$ the extended estimated state $\underline{\hat{\boldsymbol{x}}}_k^e$ and its covariance $\boldsymbol{P}_{\hat{x}_k^e}$ are defined as:

$$\underline{\hat{\boldsymbol{x}}}_k^a = \begin{bmatrix} \underline{\hat{\boldsymbol{x}}}_k^e \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \underline{\hat{\boldsymbol{x}}}_k \\ \underline{\hat{\boldsymbol{x}}}_l \\ \vdots \\ \underline{\hat{\boldsymbol{x}}}_{l+Md} \\ 0 \\ 0 \end{bmatrix} \tag{3.32}$$

$$\boldsymbol{P}_{\hat{x}_k}^a = \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k^e} & 0 & 0 \\ 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \boldsymbol{R} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k} & \boldsymbol{P}_{\hat{x}_k\hat{x}_l} & \cdots & \boldsymbol{P}_{\hat{x}_k\hat{x}_{l+Md}} & 0 & 0 \\ \boldsymbol{P}_{\hat{x}_l\hat{x}_k} & \boldsymbol{P}_{\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_l\hat{x}_{l+Md}} & 0 & 0 \\ \vdots & & \ddots & & & \\ \boldsymbol{P}_{\hat{x}_{l+Md}\hat{x}_k} & \boldsymbol{P}_{\hat{x}_{l+Md}\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_{l+Md}} & 0 & 0 \\ 0 & 0 & & 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & & 0 & 0 & \boldsymbol{R} \end{bmatrix} \tag{3.33}$$

The extended estimated state $\underline{\hat{\boldsymbol{x}}}_k^e$ and its covariance $\boldsymbol{P}_{\hat{x}_k^e}$ extend the current estimated state $\underline{\hat{\boldsymbol{x}}}_k$ and its covariance $\boldsymbol{P}_{\hat{x}_k}$ with the past ones. Here the M and d in the subscript are the same as the ones defined for Equation 3.29, i.e. the maximum amount of states and delay period in time-steps respectively. This also means that the explanation in this section limits itself to one delayed sensor. However, upgrading the equations in this section to deal with multiple delayed sensors is trivial. The difficult part in dealing with multiple delayed sensors is keeping track of the multiple past states in relation to the delayed observations, i.e. the removing and copying of the estimated state and its covariance as explained in the previous section.

Sigma-Points

While, expanding and shrinking the extended state $\underline{\hat{\boldsymbol{x}}}_k^e$ and its covariance $\boldsymbol{P}_{\hat{x}_k^e}$ it is important to keep track of the changing state dimension L, i.e.,

$$L = mL^x + L^v + L^o \tag{3.34}$$

Where, m is the current amount of states in the extended state $\underline{\hat{\boldsymbol{x}}}_k^e$. If the maximum past states M are in use for the observation sensor would simply equate to $m = M + 1$. A consequence of the changing state dimension L are the sigma-point weights, computed in Equations 3.5 to 3.8, who depend on the state dimension L.

Prediction Phase

At the start of the prediction phase the sigma-point samples are drawn based on Equation 3.17 which is restated below for convenience.

$$\boldsymbol{\mathcal{X}}^{a}_{k-1} = \left[\underline{\hat{\boldsymbol{x}}}^{a}_{k-1} \quad \underline{\hat{\boldsymbol{x}}}^{a}_{k-1} + \left(\sqrt{(L+\zeta)\,\boldsymbol{P}^{a}_{\hat{x}_{k-1}}}\right)_i \quad \underline{\hat{\boldsymbol{x}}}^{a}_{k-1} - \left(\sqrt{(L+\zeta)\,\boldsymbol{P}^{a}_{\hat{x}_{k-1}}}\right)_i\right] \qquad i = 0 \ldots L-1$$

Here, the subscript i indicates the column.

$$= \begin{bmatrix} \boldsymbol{\mathcal{X}}^{x}_{k-1} \\ \boldsymbol{\mathcal{X}}^{x}_{l} \\ \vdots \\ \boldsymbol{\mathcal{X}}^{x}_{l+Md} \\ \boldsymbol{\mathcal{X}}^{v} \\ \boldsymbol{\mathcal{X}}^{o} \end{bmatrix} \tag{3.35}$$

In this step, the cross-covariances between past and current states are expressed in the sigma-point samples. An important aspect of the SDUKF, as each column holds a variation on the past and current estimated states expressing the augmented state's covariance $\boldsymbol{P}^{a}_{\hat{x}_{k-1}}$, which includes the expanded state's covariance $\boldsymbol{P}^{e}_{\hat{x}_{k-1}}$.

For example, the sigma-point samples $\boldsymbol{\mathcal{X}}^{x}_{l}$, which are sampled around the oldest estimated state $\underline{\hat{\boldsymbol{x}}}_{l}$, are of size $(L^x, 2L+1)$. Its first $2L^x$ columns, represents the cross-covariance $\boldsymbol{P}_{\hat{x}_l \hat{x}_k}$ with the current previous estimated state $\underline{\hat{\boldsymbol{x}}}_{k-1}$. The next $2L^x$ columns, represents its own covariances $\boldsymbol{P}_{\hat{x}_l}$, and so on.

After sampling the sigma-point samples $\boldsymbol{\mathcal{X}}^{a}_{k-1}$, only the ones based on the current previous estimated state $\underline{\hat{\boldsymbol{x}}}_{k-1}$, meaning $\boldsymbol{\mathcal{X}}^{x,-}_{k}$, are propagated to the next time-step using the process model F in similar manner to Equation 3.19, i.e.,

$$\begin{aligned} \boldsymbol{\mathcal{X}}^{x,e,-}_{k} &= f(\boldsymbol{\mathcal{X}}^{x,e}_{k-1}, \boldsymbol{\mathcal{X}}^{v}_{k-1}, \underline{\boldsymbol{u}}_{k-1}) \\ \begin{bmatrix} \boldsymbol{\mathcal{X}}^{x,-}_{k} \\ \boldsymbol{\mathcal{X}}^{x}_{l} \\ \boldsymbol{\mathcal{X}}^{x}_{l+d} \\ \vdots \\ \boldsymbol{\mathcal{X}}^{x}_{l+Md} \end{bmatrix} &= \begin{bmatrix} f(\boldsymbol{\mathcal{X}}^{x}_{k-1}, \boldsymbol{\mathcal{X}}^{v}_{k-1}, \underline{\boldsymbol{u}}_{k-1}) \\ \boldsymbol{\mathcal{X}}^{x}_{l} \\ \boldsymbol{\mathcal{X}}^{x}_{l+d} \\ \vdots \\ \boldsymbol{\mathcal{X}}^{x}_{l+Md} \end{bmatrix} \end{aligned} \tag{3.36}$$

Resulting in the predicted extended state sigma-point samples $\boldsymbol{\mathcal{X}}^{x,e,-}_{k}$.

Consistent with the original UKF algorithm in Equation 3.20, the predicted extended state $\underline{\hat{\boldsymbol{x}}}^{e,-}_{k}$ is computed by taking the weighted mean of the predicted extended state sigma-point samples $\boldsymbol{\mathcal{X}}^{x,e,-}_{k}$, i.e.,

$$\underline{\hat{\boldsymbol{x}}}^{e-}_{k} = \sum_{i=0}^{2L} \boldsymbol{w}^{(m)}_{i} \boldsymbol{\mathcal{X}}^{x,e,-}_{k,i} \tag{3.37}$$

Here, the subscript i indicates the column. Only the sigma-points samples $\boldsymbol{\mathcal{X}}^{x,-}_{k}$ were projected over one time-step, the other sigma-point samples were left unaltered. This allows the computation of the predicted extended state $\underline{\hat{\boldsymbol{x}}}^{e,-}_{k}$ to be simplified to,

$$\begin{bmatrix} \underline{\hat{\boldsymbol{x}}}^{-}_{k} \\ \underline{\hat{\boldsymbol{x}}}_{l} \\ \vdots \\ \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{2L} \boldsymbol{w}^{(m)}_{i} \boldsymbol{\mathcal{X}}^{x,-}_{k,i} \\ \underline{\hat{\boldsymbol{x}}}_{l} \\ \vdots \\ \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix} \tag{3.38}$$

Following the definition of the predicted extended state $\underline{\hat{\boldsymbol{x}}}^{e,-}_{k}$ in Equation 3.38, its covariance $\boldsymbol{P}^{-}_{\hat{x}^{e}_{k}}$ can be computed. Following the same principle as with the definition of the predicted state's covariance in Equation 3.21, i.e.,

$$\boldsymbol{P}^{-}_{\hat{x}^{e}_{k}} = \sum_{i=0}^{2L} \boldsymbol{w}^{(c)}_{i} \left(\boldsymbol{\mathcal{X}}^{x,e,-}_{k,i} - \underline{\hat{\boldsymbol{x}}}^{e-}_{k}\right)\left(\boldsymbol{\mathcal{X}}^{x,e,-}_{k,i} - \underline{\hat{\boldsymbol{x}}}^{e-}_{k}\right)^{T}$$

can be further expanded into,

$$\begin{bmatrix} \boldsymbol{P}^{-}_{\hat{x}_k} & \boldsymbol{P}^{-}_{\hat{x}_k\hat{x}_l} & \cdots & \boldsymbol{P}^{-}_{\hat{x}_k\hat{x}_{l+Md}} \\ \boldsymbol{P}^{-}_{\hat{x}_l\hat{x}_k} & \boldsymbol{P}_{\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_l\hat{x}_{l+Md}} \\ \vdots & & \ddots & \\ \boldsymbol{P}^{-}_{\hat{x}_{l+Md}\hat{x}_k} & \boldsymbol{P}_{\hat{x}_{l+Md}\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_{l+Md}} \end{bmatrix} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \begin{bmatrix} \boldsymbol{\mathcal{X}}^{x,-}_{k,i} - \underline{\hat{\boldsymbol{x}}}^{-}_{k} \\ \boldsymbol{\mathcal{X}}^{x}_{l,i} - \underline{\hat{\boldsymbol{x}}}_{l} \\ \vdots \\ \boldsymbol{\mathcal{X}}^{x}_{l+Md,i} - \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\mathcal{X}}^{x,-}_{k,i} - \underline{\hat{\boldsymbol{x}}}^{-}_{k} \\ \boldsymbol{\mathcal{X}}^{x}_{l,i} - \underline{\hat{\boldsymbol{x}}}_{l} \\ \vdots \\ \boldsymbol{\mathcal{X}}^{x}_{l+Md,i} - \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix}^T \tag{3.39}$$

Within the predicted extended state's covariance only the ones related to the predicted state $\underline{\hat{\boldsymbol{x}}}^{-}_{k}$ are updated and is indicated by the superscript $-$. This fact could be used to minimize the computational load.

Correction Phase

The execution of the correction phase depends if the incoming observation is from a sensor with latency or not. However, the difference in execution is small.

The general rule is the following, when an observation $\underline{\boldsymbol{y}}$ arrives, the rows of sigma-point samples corresponding to the system's state at which the physical measurement took place are used to correct the predicted $\underline{\hat{\boldsymbol{x}}}^{-}_{k}$ and the past states.

This means that when a **delayed observation** $\underline{\boldsymbol{y}}_{l\to k}$ arrives at time step k the sigma-point samples $\boldsymbol{\mathcal{X}}^{x}_{l}$, corresponding to the past estimated state $\underline{\hat{\boldsymbol{x}}}_{l}$, are used. And, when a **non-delayed measurement** $\underline{\boldsymbol{y}}_{k}$ arrives the predicted sigma-point samples $\boldsymbol{\mathcal{X}}^{x,-}_{k}$, corresponding to the current predicted state $\underline{\hat{\boldsymbol{x}}}^{-}_{k}$, are used.

Following the structure of the nominal UKF correction phase in Section 3.2.4, the first step is the propagation of the relevant sigma-point samples through the observation model H resulting in the estimated observation points $\boldsymbol{\mathcal{Y}}$. Even though the sigma-point samples are drawn around the estimated state of the corresponding time step, only $2L^x$ columns of the matrix are based on their own covariance. The rest, $2(m-1)$ are sampled based their cross-covariance with the other estimated states's covariances within the extended state $\underline{\hat{\boldsymbol{x}}}^{e}_{k}$. And together, as the past sigma-points $\boldsymbol{\mathcal{X}}^{x}_{l}$ corresponding to time-step l are propagated in case of the delayed observation $\underline{\boldsymbol{y}}_{l\to k}$, i.e.,

$$\boldsymbol{\mathcal{Y}}_{l\to k} = \mathrm{H}(\boldsymbol{\mathcal{X}}^{x}_{l}, \boldsymbol{\mathcal{X}}^{o}) \tag{3.40}$$

Resulting in the delayed estimated observation points $\boldsymbol{\mathcal{Y}}_{l\to k}$. The the predicted sigma-points $\boldsymbol{\mathcal{X}}^{x,-}_{k}$ are used in case of the non-delayed observation $\underline{\boldsymbol{y}}_{k}$, i.e.,

$$\boldsymbol{\mathcal{Y}}_{k} = \mathrm{H}(\boldsymbol{\mathcal{X}}^{x,-}_{k}, \boldsymbol{\mathcal{X}}^{o}) \tag{3.41}$$

Resulting in the (current) estimated observation points $\boldsymbol{\mathcal{Y}}_{k}$.

The computation of the estimated observation $\underline{\hat{\boldsymbol{y}}}$ and its covariance $\boldsymbol{P}_{\hat{y}}$ in both the delayed and non-delayed case are very similar and almost identical to the original UKF Equations 3.23 and 3.22. In both cases they are found by taking the weighted mean and weighted covariance. The delayed case is defined as,

$$\underline{\hat{\boldsymbol{y}}}_{l\to k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(m)} \boldsymbol{\mathcal{Y}}_{l\to k} \tag{3.42}$$

$$\boldsymbol{P}_{\hat{y}_{l\to k}} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left(\boldsymbol{\mathcal{Y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right)\left(\boldsymbol{\mathcal{Y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right)^T \tag{3.43}$$

And the non-delayed case can be expressed as,

$$\underline{\hat{\boldsymbol{y}}}_{k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(m)} \boldsymbol{\mathcal{Y}}_{k} \tag{3.44}$$

$$\boldsymbol{P}_{\hat{y}_{k}} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left(\boldsymbol{\mathcal{Y}}_{k} - \underline{\hat{\boldsymbol{y}}}_{k}\right)\left(\boldsymbol{\mathcal{Y}}_{k} - \underline{\hat{\boldsymbol{y}}}_{k}\right)^T \tag{3.45}$$

The non-delayed case looks identical to the original UKF Equations 3.23 and 3.22. However, these estimated observation points $\boldsymbol{\mathcal{Y}}$ also include the propagated cross-covariance terms as discussed above.

The weighted cross-covariances, $\boldsymbol{P}_{\hat{x}_k^e \hat{y}_{l\to k}}$ and $\boldsymbol{P}_{\hat{x}_k^e \hat{y}_k}$, between the predicted extended state $\underline{\hat{\boldsymbol{x}}}_k^{e-}$ and the delayed $\underline{\hat{\boldsymbol{y}}}_{l\to k}$ or non-delayed estimated observation $\underline{\hat{\boldsymbol{y}}}_k$ respectively are calculated similar to Equation 3.25, i.e.,

$$\boldsymbol{P}_{\hat{x}_k^e \hat{y}_{l\to k}} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left(\boldsymbol{\mathcal{X}}_{k,i}^{x,e,-} - \underline{\hat{\boldsymbol{x}}}_k^{e-}\right) \left(\boldsymbol{\mathcal{Y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right)^T$$

Unfolding $\boldsymbol{\mathcal{X}}_k^{x,e,-}$ and $\underline{\hat{\boldsymbol{x}}}_k^{e-}$ using Equations 3.36 and 3.38, i.e.,

$$\begin{bmatrix} \boldsymbol{P}_{\hat{x}_k \hat{y}_{l\to k}} \\ \boldsymbol{P}_{\hat{x}_l \hat{y}_{l\to k}} \\ \vdots \\ \boldsymbol{P}_{\hat{x}_{l+Md} \hat{y}_{l\to k}} \end{bmatrix} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{k,i}^{x,-} - \underline{\hat{\boldsymbol{x}}}_k^{-} \\ \boldsymbol{\mathcal{X}}_{l,i}^{x} - \underline{\hat{\boldsymbol{x}}}_l \\ \vdots \\ \boldsymbol{\mathcal{X}}_{l+Md,i}^{x} - \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix} \cdot \left[\boldsymbol{\mathcal{Y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right]^T \tag{3.46}$$

The non-delayed cross-covariance $\boldsymbol{P}_{\hat{x}_k^e \hat{y}_k}$ is expressed as,

$$\boldsymbol{P}_{\hat{x}_k^e \hat{y}_k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left(\boldsymbol{\mathcal{X}}_{k,i}^{x,e,-} - \underline{\hat{\boldsymbol{x}}}_k^{e-}\right) \left(\boldsymbol{\mathcal{Y}}_k - \underline{\hat{\boldsymbol{y}}}_k\right)^T$$

Unfolding $\boldsymbol{\mathcal{X}}_k^{x,e,-}$ and $\underline{\hat{\boldsymbol{x}}}_k^{e-}$ using Equations 3.36 and 3.38, i.e.,

$$\begin{bmatrix} \boldsymbol{P}_{\hat{x}_k \hat{y}_k} \\ \boldsymbol{P}_{\hat{x}_l \hat{y}_k} \\ \vdots \\ \boldsymbol{P}_{\hat{x}_{l+Md} \hat{y}_k} \end{bmatrix} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{k,i}^{x,-} - \underline{\hat{\boldsymbol{x}}}_k^{-} \\ \boldsymbol{\mathcal{X}}_{l,i}^{x} - \underline{\hat{\boldsymbol{x}}}_l \\ \vdots \\ \boldsymbol{\mathcal{X}}_{l+Md,i}^{x} - \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix} \cdot \left[\boldsymbol{\mathcal{Y}}_l - \underline{\hat{\boldsymbol{y}}}_k\right]^T \tag{3.47}$$

With the delayed estimated observation's covariance $\boldsymbol{P}_{\hat{y}_{l\to k}}$ and delayed cross-covariance $\boldsymbol{P}_{\hat{x}_k^e \hat{y}_{l\to k}}$ defined in Equations 3.43 and 3.46 the definition of the Kalman gain $\boldsymbol{K}$ for the delayed case in Equation 3.31 at the start of this section comes back in view. Following the definition of the KF-framework, the extended Kalman gain $\boldsymbol{K}^e$, meaning the set of Kalman gains for every estimated state of the system for the given observation, is defined in the delayed case as,

$$\boldsymbol{K}_{l\to}^e = \boldsymbol{P}_{\hat{x}_k^e \hat{y}_{l\to k}} \cdot \boldsymbol{P}_{\hat{y}_{l\to k}}^{-1}$$

Here, the subscript $l\to$ on the extended Kalman gain $\boldsymbol{K}_{l\to}^e$ indicating the time-step from which the correction is based on. Using Equation 3.46 to expand $\boldsymbol{P}_{\hat{x}_k^e \hat{y}_{l\to k}}$, i.e.,

$$\begin{bmatrix} \boldsymbol{K}_{l\to k} \\ \boldsymbol{K}_{l\to l} \\ \vdots \\ \boldsymbol{K}_{l\to l+Md} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k \hat{y}_{l\to k}} \\ \boldsymbol{P}_{\hat{x}_l \hat{y}_{l\to k}} \\ \vdots \\ \boldsymbol{P}_{\hat{x}_{l+Md} \hat{y}_{l\to k}} \end{bmatrix} \cdot \boldsymbol{P}_{\hat{y}_{l\to k}}^{-1} \tag{3.48}$$

It is now clear that the Kalman gain $\boldsymbol{K}_{l\to k}$ Equation 3.31 is actually a part of the extended Kalman gain $\boldsymbol{K}_{l\to}^e$.

Employing similar tactics, the extended Kalman gain $\boldsymbol{K}_{k\to}^e$, meant for correcting the extended estimated state $\underline{\hat{\boldsymbol{x}}}_k^e$ based on the non-delayed observation $\underline{\boldsymbol{y}}_k$, is defined as,

$$\boldsymbol{K}_{k\to}^e = \boldsymbol{P}_{\hat{x}_k^e \hat{y}_k} \cdot \boldsymbol{P}_{\hat{y}_k}^{-1}$$

Using Equation 3.47 to expand $\boldsymbol{P}_{\hat{x}_k^e \hat{y}_k}$, i.e.,

$$\begin{bmatrix} \boldsymbol{K}_{k\to k} \\ \boldsymbol{K}_{k\to l} \\ \vdots \\ \boldsymbol{K}_{k\to l+Md} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k \hat{y}_k} \\ \boldsymbol{P}_{\hat{x}_l \hat{y}_k} \\ \vdots \\ \boldsymbol{P}_{\hat{x}_{l+Md} \hat{y}_k} \end{bmatrix} \cdot \boldsymbol{P}_{\hat{y}_k}^{-1} \tag{3.49}$$

Once the extended Kalman gain $\boldsymbol{K}^e$ is computed, the predicted extended state $\underline{\hat{\boldsymbol{x}}}_k^{e,-}$ and its covariance $\boldsymbol{P}_{\hat{x}_k^e}^-$ are corrected using the same principles as in the original UKF Equations 3.27 and 3.28. In the delayed case the correction of the predicted extended state $\underline{\hat{\boldsymbol{x}}}_k^{e,-}$ is defined as,

$$\underline{\hat{\boldsymbol{x}}}_k^{e} = \underline{\hat{\boldsymbol{x}}}_k^{e,-} + \boldsymbol{K}_{l\to}^{e}\left(\underline{\boldsymbol{y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right)$$

Expanding $\underline{\hat{\boldsymbol{x}}}_k^{e,-}$ and $\boldsymbol{K}_{l\to}^{e}$ using Equations 3.38 and 3.48 respectively, i.e.,

$$\begin{bmatrix} \underline{\hat{\boldsymbol{x}}}_k \\ \underline{\hat{\boldsymbol{x}}}_l^+ \\ \vdots \\ \underline{\hat{\boldsymbol{x}}}_{l+Md}^+ \end{bmatrix} = \begin{bmatrix} \underline{\hat{\boldsymbol{x}}}_k^- \\ \underline{\hat{\boldsymbol{x}}}_l \\ \vdots \\ \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{l\to k} \\ \boldsymbol{K}_{l\to l} \\ \vdots \\ \boldsymbol{K}_{l\to l+Md} \end{bmatrix} \cdot \left[\underline{\boldsymbol{y}}_{l\to k} - \underline{\hat{\boldsymbol{y}}}_{l\to k}\right] \tag{3.50}$$

Likewise, the non-delayed correction of the predicted extended state $\underline{\hat{\boldsymbol{x}}}_k^{e,-}$ is defined as,

$$\underline{\hat{\boldsymbol{x}}}_k^{e} = \underline{\hat{\boldsymbol{x}}}_k^{e,-} + \boldsymbol{K}_{k\to}^{e}\left(\underline{\boldsymbol{y}}_{k} - \underline{\hat{\boldsymbol{y}}}_{k}\right)$$

Expanding $\underline{\hat{\boldsymbol{x}}}_k^{e,-}$ and $\boldsymbol{K}_{k\to}^{e}$ using Equations 3.38 and 3.49 respectively, i.e.,

$$\begin{bmatrix} \underline{\hat{\boldsymbol{x}}}_k \\ \underline{\hat{\boldsymbol{x}}}_l^+ \\ \vdots \\ \underline{\hat{\boldsymbol{x}}}_{l+Md}^+ \end{bmatrix} = \begin{bmatrix} \underline{\hat{\boldsymbol{x}}}_k^- \\ \underline{\hat{\boldsymbol{x}}}_l \\ \vdots \\ \underline{\hat{\boldsymbol{x}}}_{l+Md} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{k\to k} \\ \boldsymbol{K}_{k\to l} \\ \vdots \\ \boldsymbol{K}_{k\to l+Md} \end{bmatrix} \cdot \left[\underline{\boldsymbol{y}}_{k} - \underline{\hat{\boldsymbol{y}}}_{k}\right] \tag{3.51}$$

Last, the extended estimated state's covariance $\boldsymbol{P}_{\hat{x}_k^e}$ is corrected using the appropriate Kalman gain. In similar fashion to Equation 3.28, the delayed can be expressed as,

$$\boldsymbol{P}_{\hat{x}_k^e} = \boldsymbol{P}_{\hat{x}_k^e}^- - \boldsymbol{K}_{l\to}^e \boldsymbol{P}_{\hat{y}_{l\to k}} \boldsymbol{K}_{l\to}^{e^T}$$

Expanding $\boldsymbol{P}_{\hat{x}_k^e}^-$ and $\boldsymbol{K}_{l\to}^{e}$ using Equations 3.36 and 3.48 respectively, i.e.,

$$= \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k}^- & \boldsymbol{P}_{\hat{x}_k\hat{x}_l}^- & \cdots & \boldsymbol{P}_{\hat{x}_k\hat{x}_{l+Md}}^- \\ \boldsymbol{P}_{\hat{x}_l\hat{x}_k}^- & \boldsymbol{P}_{\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_l\hat{x}_{l+Md}} \\ \vdots & & \ddots & \\ \boldsymbol{P}_{\hat{x}_{l+Md}\hat{x}_k}^- & \boldsymbol{P}_{\hat{x}_{l+Md}\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_{l+Md}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{K}_{l\to k} \\ \boldsymbol{K}_{l\to l} \\ \vdots \\ \boldsymbol{K}_{l\to l+Md} \end{bmatrix} \boldsymbol{P}_{\hat{y}_{l\to k}} \left[\boldsymbol{K}_{l\to k}^T \quad \boldsymbol{K}_{l\to l}^T \quad \cdots \quad \boldsymbol{K}_{l\to l+Md}^T\right] \tag{3.52}$$

$$= \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k} & \boldsymbol{P}_{\hat{x}_k\hat{x}_l^+} & \cdots & \boldsymbol{P}_{\hat{x}_k\hat{x}_{l+Md}^+} \\ \boldsymbol{P}_{\hat{x}_l^+\hat{x}_k} & \boldsymbol{P}_{\hat{x}_l^+} & & \boldsymbol{P}_{\hat{x}_l^+\hat{x}_{l+Md}^+} \\ \vdots & & \ddots & \\ \boldsymbol{P}_{\hat{x}_{l+Md^+}\hat{x}_k} & \boldsymbol{P}_{\hat{x}_{l+Md}^+\hat{x}_l^+} & & \boldsymbol{P}_{\hat{x}_{l+Md}^+} \end{bmatrix} \tag{3.53}$$

Likewise, for the non-delayed correction of the extended estimated state's covariance $\boldsymbol{P}_{\hat{x}_k^e}$, i.e.,

$$\boldsymbol{P}_{\hat{x}_k^e} = \boldsymbol{P}_{\hat{x}_k^e}^- - \boldsymbol{K}_{k\to}^e \boldsymbol{P}_{\hat{y}_{l\to k}} \boldsymbol{K}_{k\to}^{e^T}$$

Expanding $\boldsymbol{P}_{\hat{x}_k^e}^-$ and $\boldsymbol{K}_{l\to}^{e}$ using Equations 3.36 and 3.49 respectively, i.e.,

$$= \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k}^- & \boldsymbol{P}_{\hat{x}_k\hat{x}_l}^- & \cdots & \boldsymbol{P}_{\hat{x}_k\hat{x}_{l+Md}}^- \\ \boldsymbol{P}_{\hat{x}_l\hat{x}_k}^- & \boldsymbol{P}_{\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_l\hat{x}_{l+Md}} \\ \vdots & & \ddots & \\ \boldsymbol{P}_{\hat{x}_{l+Md}\hat{x}_k}^- & \boldsymbol{P}_{\hat{x}_{l+Md}\hat{x}_l} & & \boldsymbol{P}_{\hat{x}_{l+Md}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{K}_{k\to k} \\ \boldsymbol{K}_{k\to l} \\ \vdots \\ \boldsymbol{K}_{k\to l+Md} \end{bmatrix} \boldsymbol{P}_{\hat{y}_{l\to k}} \left[\boldsymbol{K}_{k\to k}^T \quad \boldsymbol{K}_{k\to l}^T \quad \cdots \quad \boldsymbol{K}_{k\to l+Md}^T\right] \tag{3.54}$$

$$= \begin{bmatrix} \boldsymbol{P}_{\hat{x}_k} & \boldsymbol{P}_{\hat{x}_k\hat{x}_l^+} & \cdots & \boldsymbol{P}_{\hat{x}_k\hat{x}_{l+Md}^+} \\ \boldsymbol{P}_{\hat{x}_l^+\hat{x}_k} & \boldsymbol{P}_{\hat{x}_l^+} & & \boldsymbol{P}_{\hat{x}_l^+\hat{x}_{l+Md}^+} \\ \vdots & & \ddots & \\ \boldsymbol{P}_{\hat{x}_{l+Md^+}\hat{x}_k} & \boldsymbol{P}_{\hat{x}_{l+Md}^+\hat{x}_l^+} & & \boldsymbol{P}_{\hat{x}_{l+Md}^+} \end{bmatrix} \tag{3.55}$$

This concludes the correction phase and the UKF implementation of the SDSF. One of the foreseeable downsides of this algorithm extension is the increased computational cost, as the matrices grow in size due to the copying of the estimated state.

4

HMD & Stewart Platform Kinematics

In preparation of the discrete-time system required by the Unscented Kalman Filter (UKF), presented in Chapter 3, the continuous-time system equations are first established in this chapter. The kinematics of the Virtual Reality (VR)/Augmented Reality (AR) system and a Stewart platform are presented individually first. Following, both these individual kinematics, the VR/AR system and Stewart platform kinematics, are joined in the next chapter, Chapter 5, into various configurations.

However before the individual kinematics are established, some general assumptions are listed in Section 4.1, simplifying the kinematics. Next, a short section about the quaternion attitude representation, used throughout this Thesis, is presented in Section 4.2. Initially, the individual kinematics of both systems are presented in their configuration with Inertial Measurement Unit (IMU). Due to the fact that in both systems the IMU measurements serve as input and the observations are based on comparable states the process model and state of both individual kinematic models are alike. The general input $\underline{\boldsymbol{u}}$, state $\underline{\boldsymbol{x}}$, and process model f of both systems are presented in Section 4.3 and purposefully expressed in certain reference frames to aid joining and extending both kinematic models in the next chapter, Chapter 5. After presenting the general process model f, the observation model of the Stewart platform itself is constructed in Section 4.4. Last, an explanation of the VR/AR system is given and the VR/AR system observation is deduced, both are found in Section 4.5. Hereby, both VR/AR system and Stewart platform initial individual system equations are completed.

4.1. Assumptions

The following assumptions will simplify the mathematical models used throughout this Thesis while only having a small effect on the accuracy of the kinematic models:

- **Rigid body of constant mass**
  It is assumed that the geometry and the mass of the systems under consideration do not change during standard use-cases.
- **Flat Earth**
  Due to the fact that the systems under consideration can only travel a short distance from their fixed origin, the Earth's curvature influence is negligible.
- **Non-rotating Earth**
  The systems under consideration can travel only a short distance from their fixed origin and have low velocities. Neglecting the rotation of the Earth ignores two accelerations, i.e. coriolis acceleration and centripetal acceleration. The coriolis acceleration is small as the velocity relative to the Earth of the systems under consideration is small. The centripetal acceleration is small and can be more importantly be assumed constant as the systems under considerations can only move small distances around a fixed origin.
- **Constant gravity** As the systems under consideration only travel a short distance from their fixed origin, this assumption is valid.

4.2. Attitude

In this section, a brief introduction will be given to the attitude representation used throughout this Thesis. The attitude will be represented by the Euler-Rodrigues quaternion formulation. The Euler-Rodrigues quaternions, from now on just called quaternions, are a computationally superior method when compared to the more intuitive Euler angle formulation [51].

The quaternion attitude representation of a rotation of one reference frame to another will be described mathematically in Equation 4.6 [71, 61, 51]. In this appendix, the mathematical relation between the quaternion and Euler angle attitude representation is also presented.

A quaternion attitude $\underline{\boldsymbol{e}}$ with respect to (w.r.t.) to the inertial reference frame $\mathbb{E}_I$ can be noted as:

$$\underline{\boldsymbol{e}} = \begin{pmatrix} e_0 & e_x & e_y & e_z \end{pmatrix} \begin{Bmatrix} 1 \\ \mathbb{E}_I \end{Bmatrix} \tag{4.1}$$

With,

$$\|\underline{\boldsymbol{e}}\| = \sqrt{e_0^2 + e_x^2 + e_y^2 + e_z^2} = 1 \tag{4.2}$$

Where e_0 is the quaternion scalar term and e_x, e_y, and e_z form the vector components of the quaternion in the inertial reference frame $\mathbb{E}_I$ and in order to represent an attitude with a quaternion it is important that the norm of the quaternion is equal to one [71, 51].

The relationship between two reference frames, an example reference frame $\mathbb{E}_Z$ and the initial reference frame $\mathbb{E}_I$, is defined as the transformation $\boldsymbol{T}_{ZI}$, i.e.,

$$\{\mathbb{E}_Z\} = \boldsymbol{T}_{ZI} \{\mathbb{E}_I\} \tag{4.3}$$

Where the transformation $\boldsymbol{T}_{ZI}$ is capable of transforming points and vectors from the inertial $\mathbb{E}_I$ to the example reference frame $\mathbb{E}_Z$. The transformation $\boldsymbol{T}_{ZI}$ can be expressed as an orthogonal matrix, meaning that the inverse is equal to the transposed. In the case of the current example this means mathematically that,

$$\boldsymbol{T}_{ZI}\boldsymbol{T}_{ZI}^{-1} = \boldsymbol{T}_{ZI}\boldsymbol{T}_{ZI}^{T} = I \rightarrow \{\mathbb{E}_I\} = \boldsymbol{T}_{ZI}^{T} \{\mathbb{E}_Z\} \tag{4.4}$$

Moreover, if in this example the quaternion $\underline{\boldsymbol{e}}$ is defined as the rotation from the inertial $\mathbb{E}_I$ to the example reference frame $\mathbb{E}_Z$, the transformation matrix $\boldsymbol{T}_{ZI}$ can be written with in terms of the quaternion $\underline{\boldsymbol{e}}$ as,

$$\boldsymbol{T}_{ZI} = \begin{bmatrix} e_0^2 + e_x^2 - e_y^2 - e_z^2 & 2\left(e_x e_y + e_0 e_z\right) & 2\left(e_x e_z - e_0 e_y\right) \\ 2\left(e_x e_y - e_0 e_z\right) & e_0^2 - e_x^2 + e_y^2 - e_z^2 & 2\left(e_y e_z + e_0 e_x\right) \\ 2\left(e_x e_z + e_0 e_y\right) & 2\left(e_y e_z - e_0 e_x\right) & e_0^2 - e_x^2 - e_y^2 + e_z^2 \end{bmatrix} \tag{4.5}$$

And can be further simplified using Equation 4.2 into,

$$= 2 \begin{bmatrix} 0.5 - e_y^2 - e_z^2 & e_x e_y + e_0 e_z & e_x e_z - e_0 e_y \\ e_x e_y - e_0 e_z & 0.5 - e_x^2 - e_z^2 & e_y e_z + e_0 e_x \\ e_x e_z + e_0 e_y & e_y e_z - e_0 e_x & 0.5 - e_x^2 - e_y^2 \end{bmatrix} \tag{4.6}$$

4.3. General Kinematics

The VR/AR system state $\underline{\boldsymbol{x}}_H$ and input $\underline{\boldsymbol{u}}_H$ and the Stewart platform state $\underline{\boldsymbol{x}}_P$ and input $\underline{\boldsymbol{u}}_P$ are similar in design in this configuration. The reason for this resemblance is because both the Stewart platform and the VR/AR system under consideration have an IMU attached to their moving, non-inertial, reference frame $\mathbb{E}_\square$ measuring specific force $\underline{\boldsymbol{f}}_{\square_m}$ and the angular rate $\underline{\boldsymbol{\omega}}_{\square_m}$. In addition, both systems other sensor observes the position $\underline{\boldsymbol{c}}_\square$ and attitude $\underline{\boldsymbol{e}}_\square$ of the moving, non-inertial, reference frame $\mathbb{E}_\square$. Based on these two statements, an overview is depicted of the general system in Figure 4.1.

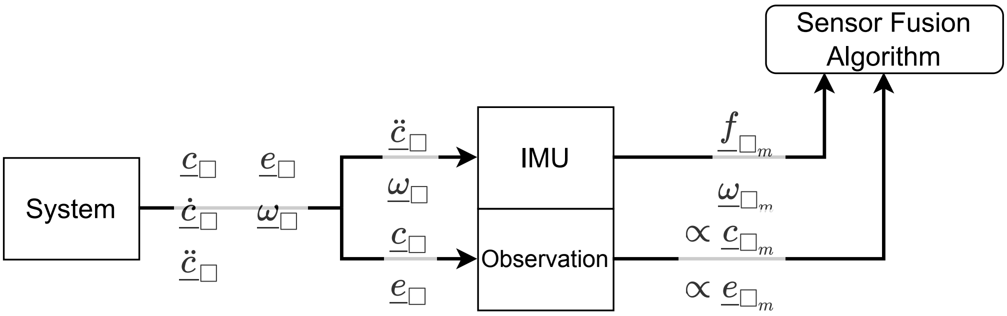

Sensor Fusion Algorithm
System
$\underline{\boldsymbol{c}}_\square$
$\underline{\dot{\boldsymbol{c}}}_\square$
$\underline{\ddot{\boldsymbol{c}}}_\square$
$\underline{\boldsymbol{e}}_\square$
$\underline{\boldsymbol{\omega}}_\square$
IMU
Observation
$\underline{\boldsymbol{f}}_{\square_m}$
$\underline{\boldsymbol{\omega}}_{\square_m}$
$\propto \underline{\boldsymbol{c}}_{\square_m}$
$\propto \underline{\boldsymbol{e}}_{\square_m}$

Figure 4.1: The general system's sensor layout and variable overview.

Moreover, by purposefully defining both, VR/AR system and Stewart platform, states' position $\underline{\boldsymbol{c}}_\square$, velocity $\underline{\dot{\boldsymbol{c}}}_\square$, acceleration $\underline{\ddot{\boldsymbol{c}}}_\square$, attitude $\underline{\boldsymbol{e}}_\square$, and angular rate $\underline{\boldsymbol{\omega}}_\square$ w.r.t. the inertial reference frame $\mathbb{E}_I$, i.e. independent of one another, the joining and extensibility of the achieved models in the next chapter is eased.

The reference frame $\mathbb{E}_\square$ will serve as a placeholder for the non-inertial reference frames of the systems themselves, $\mathbb{E}_H$ for the VR/AR system and $\mathbb{E}_P$ for the Stewart platform. A more in depth look into the respective systems and proof for these previous statements can be found in Sections 4.4 and 4.5. Moreover, any other variable in this section with a □ subscript is a placeholder for their respective system. With P as subscript to indicate Stewart platform and H to indicate the VR/AR system.

In Section 4.3.1, the input is further discussed and designed. Followed by a definition of the general state, in Section 4.3.2 based on the input and the general observation of the system. Next, the general process model is achieved in Section 4.3.3.

4.3.1. Input

Stated in Section 2.2.1 and shown in Figure 4.1, the Micro-Electro-Mechanical Systems (MEMS) based IMU under consideration measures the specific force $\underline{\boldsymbol{f}}_{\square_m}$ and the angular rate $\underline{\boldsymbol{\omega}}_{\square_m}$ of a system, where m indicates it is a measured quantity. The input vector $\underline{\boldsymbol{u}}_\square$ will be defined according to those measurements of the IMU, ie

$$\underline{\boldsymbol{u}}_\square = \begin{Bmatrix} \underline{\boldsymbol{f}}_{\square_m} \\ \underline{\boldsymbol{\omega}}_{\square_m} \end{Bmatrix} \tag{4.7}$$

However, IMU measurements are not perfect, any IMU raw measurement is corrupted by various factors, as described in Section 2.2.1. The factors with the largest influence on the IMU measurements are the bias $\underline{\boldsymbol{\lambda}}$ and Gaussian white noise $\underline{\boldsymbol{n}}$ [73, 21], resulting in a decomposition of both measured inputs, $\underline{\boldsymbol{f}}_{\square_m}$ and $\underline{\boldsymbol{\omega}}_{\square_m}$, as:

$$\underline{\boldsymbol{f}}_{\square_m} = \underline{\boldsymbol{f}}_\square + \underline{\boldsymbol{n}}_{\boldsymbol{f}_\square} + \underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_\square} \tag{4.8}$$

$$\underline{\boldsymbol{\omega}}_{\square_m} = \underline{\boldsymbol{\omega}}_\square + \underline{\boldsymbol{n}}_{\boldsymbol{\omega}_\square} + \underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_\square} \tag{4.9}$$

Where $\underline{\boldsymbol{f}}_\square$ and $\underline{\boldsymbol{\omega}}_\square$ indicate the specific force and rotation rate respectively of their system. Later in this chapter, these decompositions, in Equations 4.8 and 4.9, will help to express the change of the state over time $\underline{\dot{\boldsymbol{x}}}_\square$ in terms of the input $\underline{\boldsymbol{u}}_\square$ and indicate the necessary vectors in the state $\underline{\boldsymbol{x}}_\square$.

Specific force $\underline{\boldsymbol{f}}_\square$ is defined as the accumulation of all the non-field forces per unit mass. After the assumption of a **Flat Earth** and a **Non-rotating Earth**, explained in Section 4.1, specific force $\underline{\boldsymbol{f}}_\square$ can be defined as the accumulation of all the non-gravitational forces and expressed in terms of gravity force $\underline{\boldsymbol{g}}$ and the acceleration $\underline{\ddot{c}}$ of the system in the inertial reference frame $\mathbb{E}_I$. The specific force can thus be described in vector notation as:

$$\underline{\boldsymbol{f}} = \underline{\ddot{c}} - \underline{\boldsymbol{g}} \tag{4.10}$$

The **measured specific force** $\underline{\boldsymbol{f}}_{\square_m}$, part of the input $\underline{\boldsymbol{u}}_\square$ in Equation 4.7, is defined in the moving reference frame $\mathbb{E}_\square$. This is due to the fact that the IMU is assumed fixed to the center of the moving, non-inertial, reference frame $\mathbb{E}_\square$ and can, for all intents and purposes, be assumed aligned. Mitigations for failing either assumptions can be found in Chapter 8. Expressing the specific force $\underline{\boldsymbol{f}}_{\square_m}$ of the input $\underline{\boldsymbol{u}}_\square$ in the three orthogonal x, y, and z components of the moving reference frame $\mathbb{E}_\square$ results in:

$$\underline{\boldsymbol{f}}_{\square_m} = \begin{pmatrix} f_{\ddot{x}_{\square_m}} & f_{\ddot{y}_{\square_m}} & f_{\ddot{z}_{\square_m}} \end{pmatrix} \{\mathbb{E}_\square\} \tag{4.11}$$

Furthermore, using the decomposition of Equation 4.8 the previous equation can be expanded in terms of moving reference frame $\mathbb{E}_\square$:

$$\begin{Bmatrix} f_{\ddot{x}_{\square_m}} \\ f_{\ddot{y}_{\square_m}} \\ f_{\ddot{z}_{\square_m}} \end{Bmatrix} = \begin{Bmatrix} f_{\ddot{x}_\square} + n_{\ddot{x}_\square} + \lambda_{\ddot{x}_\square} \\ f_{\ddot{y}_\square} + n_{\ddot{y}_\square} + \lambda_{\ddot{y}_\square} \\ f_{\ddot{z}_\square} + n_{\ddot{z}_\square} + \lambda_{\ddot{z}_\square} \end{Bmatrix} \tag{4.12}$$

Likewise, the **measured angular rate** $\underline{\boldsymbol{\omega}}_{\square_m}$, part of the input $\underline{\boldsymbol{u}}_\square$ in Equation 4.7, when expressed in the moving reference frame $\mathbb{E}_\square$:

$$\underline{\boldsymbol{\omega}}_{\square_m} = \begin{pmatrix} p_{\square_m} & q_{\square_m} & r_{\square_m} \end{pmatrix} \{\mathbb{E}_\square\} \tag{4.13}$$

The previous equation can be worded, using the decomposition in Equation 4.9, in terms of moving reference frame $\mathbb{E}_\square$:

$$\begin{Bmatrix} p_{\square_m} \\ q_{\square_m} \\ r_{\square_m} \end{Bmatrix} = \begin{Bmatrix} p_\square + n_{p_\square} + \lambda_{p_\square} \\ q_\square + n_{q_\square} + \lambda_{q_\square} \\ r_\square + n_{r_\square} + \lambda_{r_\square} \end{Bmatrix} \tag{4.14}$$

Process Noise

The process noise random variables $\underline{\boldsymbol{\nu}}_\square$ are defined as the collection of Gaussian noises present in the IMU. Apart from the Gaussian white noise $\underline{\boldsymbol{n}}$ vector, in case of an IMU called the angle/velocity random walk, a second Gaussian noise vector is present. This vector is called rate/acceleration random walk $\underline{\boldsymbol{\mu}}$ and indicates the wandering of the bias $\underline{\boldsymbol{\lambda}}$ over time [73, 21]. Thus, the process noise random variables $\underline{\boldsymbol{\nu}}_\square$ can be written, using information explained here and from Equations 4.12 and 4.14, in terms of the moving reference frame $\mathbb{E}_\square$ as:

$$\underline{\boldsymbol{\nu}}_\square = \begin{Bmatrix} \underline{\boldsymbol{n}}_{\boldsymbol{f}_\square} \\ \underline{\boldsymbol{\mu}}_{\boldsymbol{f}_\square} \\ \underline{\boldsymbol{n}}_{\boldsymbol{\omega}_\square} \\ \underline{\boldsymbol{\mu}}_{\boldsymbol{\omega}_\square} \end{Bmatrix} = \begin{pmatrix} n_{\ddot{x}_\square} & n_{\ddot{y}_\square} & n_{\ddot{z}_\square} \\ \mu_{\ddot{x}_\square} & \mu_{\ddot{y}_\square} & \mu_{\ddot{z}_\square} \\ n_{p_\square} & n_{q_\square} & n_{r_\square} \\ \mu_{p_\square} & \mu_{q_\square} & \mu_{r_\square} \end{pmatrix} \{\mathbb{E}_\square\} \tag{4.15}$$

4.3.2. State

As stated in the introduction of Section 4.3 and shown in Figure 4.1, the non-IMU sensors of the systems measure the position $\underline{\boldsymbol{c}}_\square$ and attitude $\underline{\boldsymbol{e}}_\square$ directly or indirectly. This will be further explained in the sections of their respective systems, Section 4.4 for the Stewart platform and Section 4.5 for the VR/AR system.

In order to connect the input $\underline{\boldsymbol{u}}_\square$, consisting of the specific force $\underline{\boldsymbol{f}}_\square$ and the angular rate $\underline{\boldsymbol{\omega}}_\square$, with the observation $\underline{\boldsymbol{y}}$ of the position and the attitude the state $\underline{\boldsymbol{x}}$ includes the position $\underline{\boldsymbol{c}}_\square$, velocity $\underline{\dot{\boldsymbol{c}}}_\square$, attitude $\underline{\boldsymbol{e}}_\square$, and the IMU biases $\underline{\boldsymbol{\lambda}}_\square$, i.e.,

$$\underline{\boldsymbol{x}}_\square = \begin{Bmatrix} \underline{\boldsymbol{c}}_\square \\ \underline{\dot{\boldsymbol{c}}}_\square \\ \underline{\boldsymbol{e}}_\square \\ \underline{\boldsymbol{\lambda}}_\square \end{Bmatrix} \tag{4.16}$$

The **position** coordinate vector $\underline{\boldsymbol{c}}_\square$ of the center point of the moving, non-inertial, frame $\mathbb{E}_\square$ is expressed as the vector from the center point of inertial reference frame $\mathbb{E}_I$ to itself expressed in terms of the inertial reference frame $\mathbb{E}_I$ as:

$$\underline{\boldsymbol{c}}_\square = \begin{pmatrix} x_\square & y_\square & z_\square \end{pmatrix} \{\mathbb{E}_I\} \tag{4.17}$$

The **velocity** vector $\underline{\dot{\boldsymbol{c}}}_\square$ of the center point of the moving, non-inertial, frame $\mathbb{E}_\square$, in Equation 4.16, is defined as the change in position $\underline{\dot{\boldsymbol{c}}}_\square$ over time and expressed in the $\mathbb{E}_I$ reference frame, the same reference frame the position was defined in, i.e.,

$$\underline{\dot{\boldsymbol{c}}}_\square = \begin{pmatrix}\dot{x}_\square & \dot{y}_\square & \dot{z}_\square\end{pmatrix}\{\mathbb{E}_I\} + \begin{pmatrix}x_\square & y_\square & z_\square\end{pmatrix}\{\dot{\mathbb{E}}_I\} \tag{4.18}$$

The inertial reference frame $\mathbb{E}_I$ does not change over time by definition, i.e., $\{\dot{\mathbb{E}}_I\} = 0$ and thus Equation 4.18 can be reduced to,

$$= \begin{pmatrix}\dot{x}_\square & \dot{y}_\square & \dot{z}_\square\end{pmatrix}\{\mathbb{E}_I\} \tag{4.19}$$

Where $\dot{x}_\square$, $\dot{y}_\square$, and $\dot{z}_\square$ are the variables defined as the changes over time of the position $\underline{\boldsymbol{c}}_\square$ individual components, $x_\square$, $y_\square$, and $z_\square$ respectively.

The quaternion **attitude** $\underline{\boldsymbol{e}}_\square$, in Equation 4.16, is defined as the rotation from the inertial reference frame $\mathbb{E}_I$ to the moving, non-inertial, frame $\mathbb{E}_\square$ and, through the use of Equation 4.1, can be written as:

$$\underline{\boldsymbol{e}}_\square = \begin{pmatrix}e_{0_\square} & e_{x_\square} & e_{y_\square} & e_{z_\square}\end{pmatrix}\begin{Bmatrix}1\\ \mathbb{E}_I\end{Bmatrix} \tag{4.20}$$

Where $e_{0_\square}$ is the scalar term and $e_{x_\square}$, $e_{y_\square}$, and $e_{z_\square}$ are the rotational vector components expressed in the inertial frame $\mathbb{E}_I$. The transformation matrix $\boldsymbol{T}_{\square I}$ can be constructed using the quaternion attitude $\underline{\boldsymbol{e}}_\square$, and using Equation 4.6 and can be used to transform vectors from the inertial frame $\mathbb{E}_I$ to the moving frame $\mathbb{E}_\square$, i.e.,

$$\{\mathbb{E}_\square\} = \boldsymbol{T}_{\square I}\{\mathbb{E}_I\} \tag{4.21}$$

Last, the **biases** $\underline{\boldsymbol{\lambda}}_\square$ present in both inputs, namely measured specific force $\underline{\boldsymbol{f}}_{\square_m}$ and measured rotation rate $\underline{\boldsymbol{\omega}}_{\square_m}$, can be expanded in terms of the moving reference frame $\mathbb{E}_\square$, as previously done separately in Equations 4.12 and 4.14, i.e.,

$$\underline{\boldsymbol{\lambda}}_\square = \begin{Bmatrix}\underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_\square}\\ \underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_\square}\end{Bmatrix} = \begin{pmatrix}\lambda_{\ddot{x}_\square} & \lambda_{\ddot{y}_\square} & \lambda_{\ddot{z}_\square}\\ \lambda_{p_\square} & \lambda_{q_\square} & \lambda_{r_\square}\end{pmatrix}\{\mathbb{E}_\square\} \tag{4.22}$$

4.3.3. Process Model

In this section, the continuous-time process model $\mathrm{f}_\square$ will be inferred from the state $\underline{\boldsymbol{x}}_\square$ and input $\underline{\boldsymbol{u}}_\square$. In order to achieve this, the time derivate of the state $\underline{\dot{\boldsymbol{x}}}_\square$ is first computed in terms of the state $\underline{\boldsymbol{x}}_\square$, input $\underline{\boldsymbol{u}}_\square$, and process noise $\underline{\boldsymbol{\nu}}_\square$. i.e.,

$$\underline{\dot{\boldsymbol{x}}}_\square(t) = \mathrm{f}_\square\left(\underline{\boldsymbol{x}}_\square(t), \underline{\boldsymbol{u}}_\square(t), \underline{\boldsymbol{\nu}}_\square(t)\right) \tag{4.23}$$

$$\underline{\boldsymbol{y}}_\square(t) = \mathrm{h}_\square\left(\underline{\boldsymbol{x}}_\square(t), \underline{\boldsymbol{o}}_\square(t)\right) \tag{4.24}$$

By writing the process model $\mathrm{f}_\square$ as such, allows for the propagation of the process noise random variables $\underline{\boldsymbol{\nu}}_\square$ in said model. The inclusion of the process noise $\underline{\boldsymbol{\nu}}_\square$ in the continuous-time process model $\mathrm{f}_\square$ is not a coincidence and is in preparation for the process model's discrete-time counterpart F. The UKF algorithm, as presented in Chapter 3, was capable of propagating the expected process noise random variables $\underline{\boldsymbol{\nu}}$ through the process model F, as per Equation 3.3.

The same line of reasoning can be followed for the inclusion of the observation noise $\underline{\boldsymbol{o}}$ as an input to the continuous-time observation model $\mathrm{h}_\square$ and is presented in Equation 4.24 for the sake of completeness. However, the individual observation models $\mathrm{h}_\square$ will be explained in Sections 4.4.2 and 4.5.2 of the Stewart platform and Head-Mounted Display (HMD) respectively.

State Time Derivatives

Taking the time derivative of the state, defined in Section 4.3.2 in Equation 4.16, can be written as

$$\underline{\dot{\boldsymbol{x}}}_\square = \begin{Bmatrix}\underline{\dot{\boldsymbol{c}}}_\square\\ \underline{\ddot{\boldsymbol{c}}}_\square\\ \underline{\dot{\boldsymbol{e}}}_\square\\ \underline{\dot{\boldsymbol{\lambda}}}_\square\end{Bmatrix} \tag{4.25}$$

First, the time derivate of the **position** $\underline{\dot{\boldsymbol{c}}}_\square$ of the system is equal to the systems velocity $\underline{\dot{\boldsymbol{c}}}_\square$. Both position $\underline{\boldsymbol{c}}_\square$ and velocity $\underline{\dot{\boldsymbol{c}}}_\square$ were already defined in the inertial reference frame $\mathbb{E}_I$, see Equations 4.17 and 4.19. This makes expressing the time derivate of the position $\underline{\dot{\boldsymbol{c}}}_\square$ in terms of the systems's state $\underline{\boldsymbol{x}}_\square$, more precisely the system's velocity $\underline{\dot{\boldsymbol{c}}}_\square$, trivial, i.e.,

$$\underline{\dot{\boldsymbol{c}}}_\square = \underline{\dot{\boldsymbol{c}}}_\square = \begin{pmatrix}\dot{x}_\square & \dot{y}_\square & \dot{z}_\square\end{pmatrix}\{\mathbb{E}_I\} \tag{4.26}$$

Second, the time derivate of the **velocity** $\dot{\underline{\dot{\boldsymbol{c}}}}_\square$ of the system is equal to the acceleration $\underline{\ddot{\boldsymbol{c}}}_\square$ of said system. Given that the velocity was also defined in the inertial frame $\mathbb{E}_I$, in Equation 4.19, the acceleration $\underline{\ddot{\boldsymbol{c}}}_\square$ of the system can be expressed as,

$$\underline{\ddot{\boldsymbol{c}}}_\square = \begin{pmatrix}\ddot{x}_\square & \ddot{y}_\square & \ddot{z}_\square\end{pmatrix}\{\mathbb{E}_I\} + \begin{pmatrix}\dot{x}_\square & \dot{y}_\square & \dot{z}_\square\end{pmatrix}\left\{\dot{\mathbb{E}}_I\right\} \tag{4.27}$$

Again, the inertial reference frame $\mathbb{E}_I$ does not change over time by definition, i.e., $\left\{\dot{\mathbb{E}}_I\right\} = 0$ and thus Equation 4.27 can be reduced to,

$$= \begin{pmatrix}\ddot{x}_\square & \ddot{y}_\square & \ddot{z}_\square\end{pmatrix}\{\mathbb{E}_I\} \tag{4.28}$$

Here, the selection of the inertial reference frame $\mathbb{E}_I$ over the moving reference frame $\mathbb{E}_\square$ to express the velocity $\underline{\dot{\boldsymbol{c}}}_\square$ in results in a slight reduction in computational cost. This is due to the influence a rotating reference frame has on any positional time derivative.

When discussing the design of the input in Section 4.3.1 it was concluded that due to the assumptions **Flat Earth** and a **Non-rotating Earth**, from Section 4.1, the specific force $\underline{\boldsymbol{f}}_\square$ was expressed in Equation 4.10 in terms of gravity $\underline{\boldsymbol{g}}$ and the system's acceleration $\underline{\ddot{\boldsymbol{c}}}_\square$ in the inertial reference frame, i.e.,

$$\underline{\ddot{\boldsymbol{c}}}_\square = \underline{\boldsymbol{f}}_\square + \underline{\boldsymbol{g}}$$

Expanding the specific force $\underline{\boldsymbol{f}}_\square$ using Equation 4.8,

$$= \underline{\boldsymbol{f}}_{\square_m} - \underline{\boldsymbol{\lambda}}_\square - \underline{\boldsymbol{n}}_\square + \underline{\boldsymbol{g}}$$

Expressing the vectors in their previously defined reference frames using Equations 4.11 and 4.28, i.e.,

$$\begin{pmatrix}\ddot{x}_\square & \ddot{y}_\square & \ddot{z}_\square\end{pmatrix}\{\mathbb{E}_I\} = \begin{pmatrix}f_{\ddot{x}_\square} & f_{\ddot{y}_\square} & f_{\ddot{z}_\square}\end{pmatrix}\{\mathbb{E}_\square\} + \begin{pmatrix}0 & 0 & g\end{pmatrix}\{\mathbb{E}_I\} \tag{4.29}$$

Using Equation 4.21 to state all components in terms of the inertial reference frame $\mathbb{E}_I$ results in:

$$\begin{pmatrix}\ddot{x}_\square & \ddot{y}_\square & \ddot{z}_\square\end{pmatrix} = \begin{pmatrix}f_{\ddot{x}_\square} & f_{\ddot{y}_\square} & f_{\ddot{z}_\square}\end{pmatrix}\boldsymbol{T}_{\square I} + \begin{pmatrix}0 & 0 & g\end{pmatrix}$$

Finally, taking the transpose of the previous equation to achieve a column notation to aid clarity,

$$\begin{Bmatrix}\ddot{x}_\square \\ \ddot{y}_\square \\ \ddot{z}_\square\end{Bmatrix} = \boldsymbol{T}_{\square I}^T \begin{Bmatrix}f_{\ddot{x}_{\square_m}} - \lambda_{\ddot{x}_\square} - n_{\ddot{x}_\square} \\ f_{\ddot{y}_{\square_m}} - \lambda_{\ddot{y}_\square} - n_{\ddot{y}_\square} \\ f_{\ddot{z}_{\square_m}} - \lambda_{\ddot{z}_\square} - n_{\ddot{z}_\square}\end{Bmatrix} + \begin{Bmatrix}0 \\ 0 \\ g\end{Bmatrix} \tag{4.30}$$

Concluding that the time derivative of the velocity $\underline{\ddot{\boldsymbol{c}}}_\square$ can be written in terms of the state $\underline{\boldsymbol{x}}_\square$, input $\underline{\boldsymbol{u}}_\square$, and process noise $\underline{\boldsymbol{\nu}}_\square$.

Third, the time derivate of the quaternion **attitude** $\underline{\dot{\boldsymbol{e}}}_\square$ of a system is based on the angular rate $\underline{\boldsymbol{\omega}}_\square$ of said system. It was found in [51], that the change of the quaternion attitude over time with the rotation rate $\underline{\boldsymbol{\omega}}_\square$ aligned with the moving reference frame $\mathbb{E}_\square$ could be expressed as:

$$\underline{\dot{\boldsymbol{e}}}_\square = \frac{1}{2}\underline{\boldsymbol{e}}_\square \underset{L}{\otimes} \underline{\boldsymbol{\omega}}_\square \tag{4.31}$$

Where $\underset{L}{\otimes}$ represents the quaternion product which respects the left-to-right rotation sequence [71, 51, 61]. Equation 4.31 can be expanded into its scalar and vector components:

$$\begin{Bmatrix}\dot{e}_{0_\square} \\ \dot{e}_{x_\square} \\ \dot{e}_{y_\square} \\ \dot{e}_{z_\square}\end{Bmatrix} = \frac{1}{2}\begin{bmatrix}e_{0_\square} & -e_{x_\square} & -e_{y_\square} & -e_{z_\square} \\ e_{x_\square} & e_{0_\square} & -e_{z_\square} & e_{y_\square} \\ e_{y_\square} & e_{z_\square} & e_{0_\square} & -e_{x_\square} \\ e_{z_\square} & -e_{y_\square} & e_{x_\square} & e_{0_\square}\end{bmatrix}\begin{Bmatrix}0 \\ p_\square \\ q_\square \\ r_\square\end{Bmatrix} \tag{4.32}$$

The quaternion representation of the rotation rate $\underline{\boldsymbol{\omega}}_\square$ defined in the previous equation has a zero scalar value. The zero scalar value of the rotational rate $\underline{\boldsymbol{\omega}}_\square$ allows the removal of the first column of the quaternion rotation matrix. Furthermore, expanding the measured angular rate $\underline{\boldsymbol{\omega}}_{\square_m}$ in Equation 4.14, enables the previous equation, Equation 4.32, to be expressed fully in terms of the state $\underline{\boldsymbol{x}}_\square$, input $\underline{\boldsymbol{x}}_\square$, and process noise $\underline{\boldsymbol{\nu}}_\square$ as:

$$\begin{Bmatrix} \dot{e}_{0_\square} \\ \dot{e}_{x_\square} \\ \dot{e}_{y_\square} \\ \dot{e}_{z_\square} \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} -e_{x_\square} & -e_{y_\square} & -e_{z_\square} \\ e_{0_\square} & -e_{z_\square} & e_{y_\square} \\ e_{z_\square} & e_{0_\square} & -e_{x_\square} \\ -e_{y_\square} & e_{x_\square} & e_{0_\square} \end{bmatrix} \begin{Bmatrix} p_{\square_m} - \lambda_{p_\square} - n_{p_\square} \\ q_{\square_m} - \lambda_{q_\square} - n_{q_\square} \\ r_{\square_m} - \lambda_{r_\square} - n_{r_\square} \end{Bmatrix} \tag{4.33}$$

Next, the time derivative of the **biases** $\underline{\dot{\boldsymbol{\lambda}}}_\square$, of both the specific force $\underline{\boldsymbol{f}}_{\square_m}$ and angular rate $\underline{\boldsymbol{\omega}}_{\square_m}$ as measured by the IMU, equates to the rate/acceleration random walk $\underline{\boldsymbol{\mu}}$ as mentioned in Sections 4.3.1 and 2.2.1, i.e.,

$$\underline{\dot{\boldsymbol{\lambda}}}_\square = \begin{Bmatrix} \underline{\dot{\boldsymbol{\lambda}}}_{\boldsymbol{f}_\square} \\ \underline{\dot{\boldsymbol{\lambda}}}_{\boldsymbol{\omega}_\square} \end{Bmatrix} = \begin{pmatrix} \dot{\lambda}_{\ddot{x}_\square} & \dot{\lambda}_{\ddot{y}_\square} & \dot{\lambda}_{\ddot{z}_\square} \\ \dot{\lambda}_{p_\square} & \dot{\lambda}_{q_\square} & \dot{\lambda}_{r_\square} \end{pmatrix} \{\mathbb{E}_\square\} = \begin{pmatrix} \mu_{\ddot{x}_\square} & \mu_{\ddot{y}_\square} & \mu_{\ddot{z}_\square} \\ \mu_{p_\square} & \mu_{q_\square} & \mu_{r_\square} \end{pmatrix} \{\mathbb{E}_\square\} = \begin{Bmatrix} \underline{\boldsymbol{\mu}}_{\boldsymbol{f}_\square} \\ \underline{\boldsymbol{\mu}}_{\boldsymbol{\omega}_\square} \end{Bmatrix} = \underline{\boldsymbol{\mu}}_\square \tag{4.34}$$

Last, expanding the state time derivate and the process model, of Equation 4.23, with Equations 4.26, 4.28, 4.33, and 4.34 results in the overview:

$$\begin{gathered} \underline{\dot{\boldsymbol{x}}}_\square(t) = \mathrm{f}_\square\left(\underline{\boldsymbol{x}}_\square(t), \underline{\boldsymbol{u}}_\square(t), \underline{\boldsymbol{\nu}}_\square(t)\right) \\ \begin{Bmatrix} \dot{x}_\square \\ \dot{y}_\square \\ \dot{z}_\square \\ \ddot{x}_\square \\ \ddot{y}_\square \\ \ddot{z}_\square \\ \dot{e}_{0_\square} \\ \dot{e}_{x_\square} \\ \dot{e}_{y_\square} \\ \dot{e}_{z_\square} \\ \dot{\lambda}_{\ddot{x}_\square} \\ \dot{\lambda}_{\ddot{y}_\square} \\ \dot{\lambda}_{\ddot{z}_\square} \\ \dot{\lambda}_{p_\square} \\ \dot{\lambda}_{q_\square} \\ \dot{\lambda}_{r_\square} \end{Bmatrix} = \begin{Bmatrix} \dot{x}_\square \\ \dot{y}_\square \\ \dot{z}_\square \\ \boldsymbol{T}_{\square I}^T \begin{Bmatrix} f_{\ddot{x}_{\square_m}} - \lambda_{\ddot{x}_\square} - n_{\ddot{x}_\square} \\ f_{\ddot{y}_{\square_m}} - \lambda_{\ddot{y}_\square} - n_{\ddot{y}_\square} \\ f_{\ddot{z}_{\square_m}} - \lambda_{\ddot{z}_\square} - n_{\ddot{z}_\square} \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ g \end{Bmatrix} \\ \frac{1}{2} \begin{bmatrix} -e_{x_\square} & -e_{y_\square} & -e_{z_\square} \\ e_{0_\square} & -e_{z_\square} & e_{y_\square} \\ e_{z_\square} & e_{0_\square} & -e_{x_\square} \\ -e_{y_\square} & e_{x_\square} & e_{0_\square} \end{bmatrix} \begin{Bmatrix} p_{\square_m} - \lambda_{p_\square} - n_{p_\square} \\ q_{\square_m} - \lambda_{q_\square} - n_{q_\square} \\ r_{\square_m} - \lambda_{r_\square} - n_{r_\square} \end{Bmatrix} \\ \mu_{\ddot{x}_\square} \\ \mu_{\ddot{y}_\square} \\ \mu_{\ddot{z}_\square} \\ \mu_{p_\square} \\ \mu_{q_\square} \\ \mu_{r_\square} \end{Bmatrix} \end{gathered} \tag{4.35}$$

4.4. Stewart Platform Kinematics

A Stewart platform has six degrees of freedom and is formed by two connected rigid frames [1], as in Figure 4.2. These two frames are connected by six cylindrical actuators with Hooke joints at each end, the six dotted lines in Figure 4.2. Hooke joints provide 2 degrees of freedom.

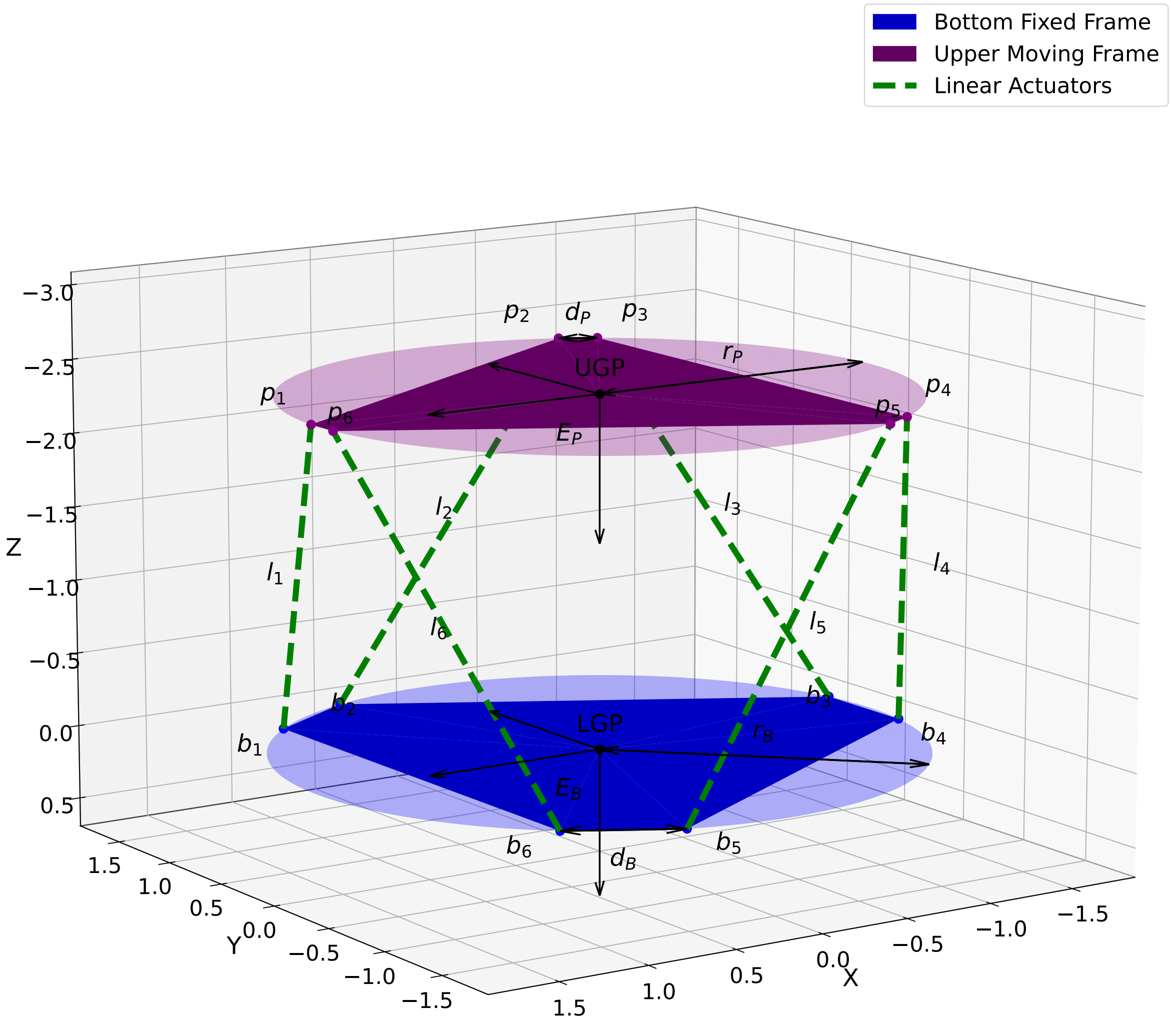


Figure 4.2: Geometric representation of a Stewart platform.

The lower frame, also called the base frame and blue in Figure 4.2, is fixed to the ground and has three pair of Hooke joints, six in total, a constant radius r_B from the center point of the frame. Each pair is spaced $120°$ on the frame. The upper frame is connected to the lower frame with also three pair of Hooke joints placed on a circular path a radius r_P, connecting six cylindrical actuators to the Hooke joints. The upper frame connection points are also spaced $120°$ from each other. Each Hooke joint connected to the closest Hooke joint on the other frame via a cylindrical actuator. Each Hooke joint of a pair on a frame is connected to a different pair on the other frame, thus creating a $60°$ offset in resting position from one frame to the next. The spacing between each Hooke joint within a pair is described by d_B and d_P for the base frame and upper frame respectively.

Each frame of the Stewart platform also defines a reference frame. $\mathbb{E}_B$ is attached to the lower fixed frame with the origin the center of the circle on which the pairs of Hooke joints where placed. Moreover, the origin is referred to as Lower Gimbal Point (LGP). Following assumptions **Flat Earth** and **Non-rotating Earth**, reference frame $\mathbb{E}_B$ is assumed to be the same as the inertial reference frame $\mathbb{E}_I$.

The reference frame attached to the upper moving frame is denoted as $\mathbb{E}_P$. Its origin is also placed in the center of the circle described by the placement of the Hooke joint pairs on the upper frame. Moreover, the origin is referred to as Upper Gimbal Point (UGP).

For the Stewart platform individual kinematics, the sensor layout will be the same as mentioned in [43] and discussed in Sections 2.3.1. The general sensor layout and variable data flow presented in Figure 4.1 can be specialized into Figure 4.3. The general kinematics are compatible due to the fact that the IMU, attached in the UGP and aligned to the upper moving reference frame $\mathbb{E}_P$, measures the specific force $\underline{\boldsymbol{f}}_{P_m}$ and the angular rate $\underline{\boldsymbol{\omega}}_{P_m}$, while also having linear encoders on each cylindrical actuator of the Stewart platform, measuring the length of said actuators $\boldsymbol{l}_m$. In turn the length of the actuators is expressed in terms of the position $\underline{\boldsymbol{c}}_P$ and attitude $\underline{\boldsymbol{e}}_P$ in the upcoming Section 4.4.2. The compatibility with the general kinematics is also highlighted in the similarity of Figure 4.3 to Figure 4.1.

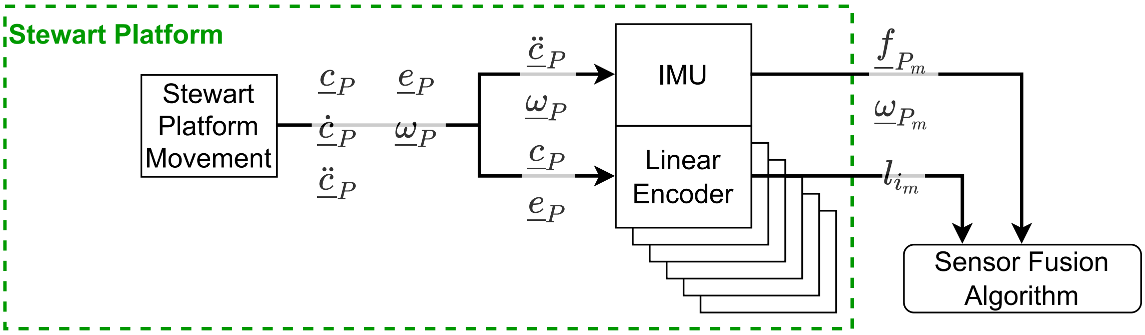


Figure 4.3: The SRS system's sensor layout and variable overview.

4.4.1. Process Model

Following the definitions of the reference frames and the sensors of the Stewart platform, the state $\underline{\boldsymbol{x}}_P$, the input $\underline{\boldsymbol{u}}_P$, the process noise $\underline{\boldsymbol{\nu}}_P$, and process model f_P can be extracted from Section 4.3 by replacing the placeholder $\square$ by the Stewart platform indicator P. The observation $\underline{\boldsymbol{y}}_P$ and compatible observation model h_P are to be determined in the next section. The time-continuous observation model h_P will be based on the work done by Miletović et al. [43].

First, the **input** vector $\underline{\boldsymbol{u}}_P$ of the Stewart platform process model are the measurements of the IMU attached to the upper moving frame $\mathbb{E}_P$ and assumed at the center of the UGP. This is in line with Section 4.3.1, and thus defines not only the input $\underline{\boldsymbol{u}}_P$ but also the process noise $\underline{\boldsymbol{\nu}}_P$ via Equations 4.7 and 4.15.

Second, the **state** vector $\underline{\boldsymbol{x}}_P$ of the Stewart platform is defined in the same manner as in Equation 4.16 and resembles the state of the upper moving platform. Whereby, the **position** $\underline{\boldsymbol{c}}_P$ tracks the center of the upper moving platform, i.e. UGP, in reference to the LGP and defined in the inertial reference frame $\mathbb{E}_I$, alike to Equation 4.17. Following, the **velocity** $\underline{\dot{\boldsymbol{c}}}_P$ of the UGP is defined as the change in said position $\underline{\boldsymbol{c}}_P$ over time, again comparable to Equation 4.19. Moreover, the quaternion **attitude** $\underline{\boldsymbol{e}}_P$ is defined as the rotation from the lower fixed frame $\mathbb{E}_B$, equal to the inertial reference frame $\mathbb{E}_I$, to the upper moving frame $\mathbb{E}_P$ and can be written in the same way as Equation 4.20. Ensuing the definitions of the input $\underline{\boldsymbol{u}}_P$ and the process noise $\underline{\boldsymbol{\nu}}_P$, the **biases** $\underline{\boldsymbol{\lambda}}_P$ can similarly be defined as in Equation 4.22.

Last, having defined the relevant vectors, the process model f_P is constructed in the same manner as performed in Section 4.3.3 in Equation 4.35.

4.4.2. Observation Model

In this section, the continuous-time observation model h_P of the Stewart platform and subsequent observation vector $\underline{\boldsymbol{y}}_P$ are defined and discussed. The general design of the continuous-time observation model h_P was already setup in Section 4.3.3 in Equation 4.24. Analogous to the continuous-time process model case, the observation noise random variables $\underline{\boldsymbol{o}}_P$ are included in the continuous-time observation model h_P for the same reasons. Namely, the propagation of said noises through the observation model as the UKF algorithm presented in Chapter 3 can take advantage of this fact.

Let the observation vector $\underline{\boldsymbol{y}}_P$ of the Stewart platform be defined by the six lengths l_{i_m} measured by the linear encoders of hydraulic actuators, i.e.,

$$\underline{\boldsymbol{y}}_P = \boldsymbol{l}_m = \left\{ l_{i_m} \right\} \qquad \forall i = 1, \dots, 6 \tag{4.36}$$

As discussed in Section 6.2.1, each linear encoder measurement l_{i_m} is corrupted additively by a noise value n_{l_i}, i.e.,

$$l_{i_m} = l_i + n_{l_i} \tag{4.37}$$

Allowing Equation 4.36 to be rewritten in terms of hydraulic actuator lengths l_i and related noise terms n_{l_i} as:

$$\underline{\boldsymbol{y}}_P = \boldsymbol{l}_m = \left\{ l_i + n_{l_i} \right\} \quad \forall i = 1, \dots, 6 \tag{4.38}$$

Following the definition of the observation vector $\underline{\boldsymbol{y}}_P$ in Equation 4.38 the observation noise random variables can be defined as:

$$\underline{\boldsymbol{o}}_P = \left\{ n_{l_i} \right\} \quad \forall i = 1, \dots, 6 \tag{4.39}$$

Given the definitions of the position $\underline{\boldsymbol{c}}_P$ and attitude $\underline{\boldsymbol{e}}_P$ of the upper moving frame, explained in Section 4.4.1, it is possible now to define the output of the six linear encoders l. In order to express the length of each of the hydraulic actuators, a coordinate vector $\underline{\boldsymbol{l}}_i$ expressing the hydraulic actuators in space is first created. The norm of said vector $\|\underline{\boldsymbol{l}}_i\|$ will result in the length of respective hydraulic actuator l_i.

Each of the hydraulic actuator position $\underline{\boldsymbol{l}}_i$ can be described as the difference between the vectors $\underline{\boldsymbol{b}}_i$ and $\underline{\boldsymbol{p}}_i$, defined in turn from the LGP and UGP respectively to each of their frame six joint in the lower fixed frame $\mathbb{E}_B$ and upper moving frame $\mathbb{E}_P$ respectively [43, 1], while adding the UGP coordinate vector $\underline{\boldsymbol{c}}_P$, i.e.,

$$\underline{\boldsymbol{l}}_i = \underline{\boldsymbol{c}}_P + \underline{\boldsymbol{p}}_i - \underline{\boldsymbol{b}}_i \quad \forall i = 1, \dots, 6 \tag{4.40}$$

With the joint coordinates for the lower $\underline{\boldsymbol{b}}_i$ and upper frame $\underline{\boldsymbol{b}}_i$ defined as,

$$\left.\begin{aligned} \underline{\boldsymbol{b}}_i &= \begin{pmatrix} b_{x_i} & b_{y_i} & b_{z_i} \end{pmatrix} \{\mathbb{E}_B\} \\ \underline{\boldsymbol{p}}_i &= \begin{pmatrix} p_{x_i} & p_{y_i} & p_{z_i} \end{pmatrix} \{\mathbb{E}_P\} \end{aligned}\right\} \quad \forall i = 1, \dots, 6$$

In the case of the SIMONA Research Simulator (SRS), the z-component in the respective local frame of both the upper and lower joint coordinates are equal to zero, reducing the previous equation to:

$$\left.\begin{aligned} \underline{\boldsymbol{b}}_i &= \begin{pmatrix} b_{x_i} & b_{y_i} & 0 \end{pmatrix} \{\mathbb{E}_B\} \\ \underline{\boldsymbol{p}}_i &= \begin{pmatrix} p_{x_i} & p_{y_i} & 0 \end{pmatrix} \{\mathbb{E}_P\} \end{aligned}\right\} \quad \forall i = 1, \dots, 6 \tag{4.41}$$

The local joint coordinates are further detailed in Section 4.4.2.1 in terms of geometric variables from Figure 4.2.

The resulting vectors $\underline{\boldsymbol{l}}_i$ written in the inertial frame $\mathbb{E}_I$, equal to the lower fixed frame $\mathbb{E}_B$, with the help of Equation 4.6,

$$\underline{\boldsymbol{l}}_i = \begin{pmatrix} x_P & y_P & z_P \end{pmatrix} \{\mathbb{E}_B\} + \begin{pmatrix} p_{x_i} & p_{y_i} & 0 \end{pmatrix} \{\mathbb{E}_P\} - \begin{pmatrix} b_{x_i} & b_{y_i} & 0 \end{pmatrix} \{\mathbb{E}_B\} \quad \forall i = 1, \dots, 6$$

Expressing all components in terms of the inertial reference frame $\mathbb{E}_I$ using Equation 4.21,

$$= \left(\begin{pmatrix} x_P & y_P & z_P \end{pmatrix} + \begin{pmatrix} p_{x_i} & p_{y_i} & 0 \end{pmatrix} \boldsymbol{T}_{PI} - \begin{pmatrix} b_{x_i} & b_{y_i} & 0 \end{pmatrix} \right) \{\mathbb{E}_I\} \quad \forall i = 1, \dots, 6 \tag{4.42}$$

Next, the norm can be taken of each of the six vector statements describing the position of the six hydraulic actuator $\underline{\boldsymbol{l}}_i$ (for all $i = 1, \dots, 6$) in order to achieve the length of each l_i, i.e.,

$$l_i = \|\underline{\boldsymbol{l}}_i\| = \left\|\begin{pmatrix} x_P & y_P & z_P \end{pmatrix} + \begin{pmatrix} p_{x_i} & p_{y_i} & 0 \end{pmatrix} \boldsymbol{T}_{PI} - \begin{pmatrix} b_{x_i} & b_{y_i} & 0 \end{pmatrix}\right\| \quad \forall i = 1, \dots, 6$$

Taking the transpose to achieve the column notation, i.e.,

$$= \left\| \begin{Bmatrix} x_P \\ y_P \\ z_P \end{Bmatrix} + \boldsymbol{T}_{PI}^T \begin{Bmatrix} p_{x_i} \\ p_{y_i} \\ 0 \end{Bmatrix} - \begin{Bmatrix} b_{x_i} \\ b_{y_i} \\ 0 \end{Bmatrix} \right\| \quad \forall i = 1, \dots, 6$$

Expanding the transformation matrix $\boldsymbol{T}_{PI}^T$ using Equation 4.6 and solving the matrix multiplication results in,

$$= \left\| \begin{matrix} x_P + 2\left(p_{x_i}\left(0.5 - e_{y_P}^2 - e_{z_P}^2\right) + p_{y_i}\left(e_{x_P}e_{y_P} - e_{0_P}e_{z_P}\right)\right) - b_{x_i} \\ y_P + 2\left(p_{x_i}\left(e_{x_P}e_{y_P} + e_{0_P}e_{z_P}\right) + p_{y_i}\left(0.5 - e_{x_P}^2 - e_{z_P}^2\right)\right) - b_{y_i} \\ z_P + 2\left(p_{x_i}\left(e_{x_P}e_{z_P} - e_{0_P}e_{y_P}\right) + p_{y_i}\left(e_{y_P}e_{z_P} + e_{0_P}e_{x_P}\right)\right) \end{matrix} \right\| \forall i = 1, \dots, 6 \tag{4.43}$$

Last, the time-continuous observation model of the Stewart platform, a general version presented in Equation 4.24, can be expanded by the use of Equations 4.38 and 4.43 into:

$$\left.\begin{aligned} \underline{\boldsymbol{y}}_P(t) &= \mathrm{h}_P\left(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{o}}_P(t)\right) \\ &= l_i + n_{l_i} \\ &= \left\|\begin{pmatrix} x_P & y_P & z_P \end{pmatrix} + \begin{pmatrix} p_{x_i} & p_{y_i} & 0 \end{pmatrix} \boldsymbol{T}_{PI} - \begin{pmatrix} b_{x_i} & b_{y_i} & 0 \end{pmatrix}\right\| + n_{l_i} \\ &= \left\| \begin{matrix} x_P + 2\left(p_{x_i}\left(0.5 - e_{y_P}^2 - e_{z_P}^2\right) + p_{y_i}\left(e_{x_P}e_{y_P} - e_{0_P}e_{z_P}\right)\right) - b_{x_i} \\ y_P + 2\left(p_{x_i}\left(e_{x_P}e_{y_P} + e_{0_P}e_{z_P}\right) + p_{y_i}\left(0.5 - e_{x_P}^2 - e_{z_P}^2\right)\right) - b_{y_i} \\ z_P + 2\left(p_{x_i}\left(e_{x_P}e_{z_P} - e_{0_P}e_{y_P}\right) + p_{y_i}\left(e_{y_P}e_{z_P} + e_{0_P}e_{x_P}\right)\right) \end{matrix} \right\| + n_{l_i} \end{aligned}\right\} \forall i = 1, \dots, 6 \tag{4.44}$$

Joint Coordinates

In order to express the length of the actuator in terms of the state $\underline{\boldsymbol{x}}_P$, the joint coordinates are required. Pointed out in [43] and from the information in Figure 4.2 the radii, r_B and r_P, and the distance between two consecutive joints, d_B and d_P, completely define their respective frame's geometry.

In order to write the position coordinates b_{x_i}, b_{y_i}, p_{x_i}, and p_{y_i} in terms of the geometric distances r_B, r_P, r_B, and r_P, first the angles θ_B and θ_P are defined as the angle between two joints of a joint pair of the lower and upper frame respectively, i.e.,

$$\begin{aligned} \theta_B &= 2 \arcsin\left(\frac{d_B}{2r_B}\right) \\ \theta_P &= 2 \arcsin\left(\frac{d_P}{2r_P}\right) \end{aligned} \tag{4.45}$$

Followed by the angles ξ_B and ξ_P, which are defined as the angle between two consecutive joints not part of the same pair of the lower and upper frame respectively, i.e.,

$$\begin{aligned} \xi_B &= \frac{2\pi - 3\theta_B}{3} = \frac{2}{3}\pi - \theta_B \\ \xi_P &= \frac{2\pi - 3\theta_P}{3} = \frac{2}{3}\pi - \theta_P \end{aligned} \tag{4.46}$$

Using the definitions of θ_B, θ_P, ξ_B, and ξ_P it is possible to write the coordinates for the lower static base as:

$$b_{x_i} = \begin{cases} r_B \cos\left(\frac{i}{2}\xi_B + \frac{i-1}{2}\theta_B\right) \\ r_B \cos\left(\frac{i-1}{2}\xi_B + \frac{i}{2}\theta_B\right) \end{cases} \qquad b_{y_i} = \begin{cases} r_B \sin\left(\frac{i}{2}\xi_B + \frac{i-1}{2}\theta_B\right) \\ r_B \sin\left(\frac{i-1}{2}\xi_B + \frac{i}{2}\theta_B\right) \end{cases} \quad \forall i = \begin{cases} 1, 3, 5 \\ 2, 4, 6 \end{cases} \tag{4.47}$$

And for the upper moving base as:

$$p_{x_i} = \begin{cases} r_P \cos\left(\frac{i-1}{2}\xi_P + \frac{i}{2}\theta_P\right) \\ r_P \cos\left(\frac{i}{2}\xi_P + \frac{i-1}{2}\theta_P\right) \end{cases} \qquad p_{y_i} = \begin{cases} r_P \sin\left(\frac{i-1}{2}\xi_P + \frac{i}{2}\theta_P\right) \\ r_P \sin\left(\frac{i}{2}\xi_P + \frac{i-1}{2}\theta_P\right) \end{cases} \quad \forall i = \begin{cases} 1, 3, 5 \\ 2, 4, 6 \end{cases} \tag{4.48}$$

4.5. General VR/AR System Kinematics

Mentioned in Section 2.2, the VR/AR systems under consideration are systems with a HMD that contains its own IMU and employs a form of visual outside-in tracking. If inside-out visual tracking is required the observation model h, at least, should be modified.

The HMD has its own right-handed reference frame $\mathbb{E}_H$, shown in Figure 4.4, which is located at the center between the retinas [34] and fixed to the HMD. From Figure 4.4 it is clear that the x-axis points along the display view, the y-axis points from the right side of the HMD and the z-axis points down.

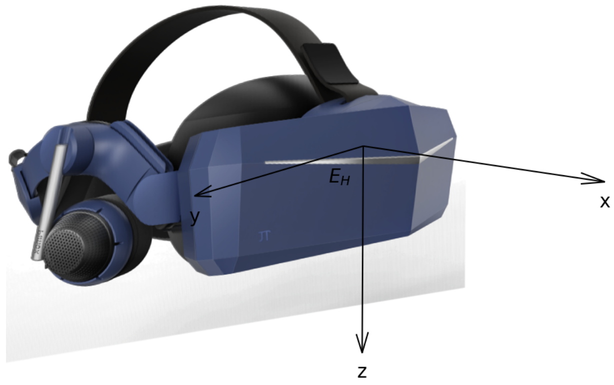


Figure 4.4: Reference frame $\mathbb{E}_H$ attached to the HMD.

Including assumptions **Flat Earth** and **Non-rotating Earth**, it is assumed the visual tracking reference frame is fixed w.r.t. to the environment it is attached to. The center point of the visual tracking reference frame is initialized at the starting position in space of the HMD. The visual tracking reference frame is assumed aligned to the reference frame it is attached to.

Following from the literature review in Section 2.2, the information flow of an individual VR/AR system is schematized in Figure 4.5 by relaying on Figure 4.1 of the general kinematics. Here, the state $\underline{\boldsymbol{x}}_H$ is observed by two types of sensors, an IMU and visual tracking. The IMU that is attached to the HMD between the retinas of the user, i.e. the center of the HMD reference frame $\mathbb{E}_H$, measures the specific force $\underline{\boldsymbol{f}}_{H_m}$ and angular rate $\underline{\omega}_{H_m}$ in the $\mathbb{E}_H$ reference frame from the HMD's acceleration $\underline{\ddot{c}}_H$ and angular rate $\underline{\omega}_H$ w.r.t. the inertial reference frame $\mathbb{E}_I$. Also, the visual tracking results in the relative measured position $\underline{c}_{H_m}$ and attitude $\underline{e}_{H_m}$ of the HMD while observing the HMD's pose. Based on the measurements, the pose, $\underline{\hat{c}}_H$ and $\underline{\hat{e}}_H$, can be estimated by the sensor fusion algorithm, in this case the UKF.

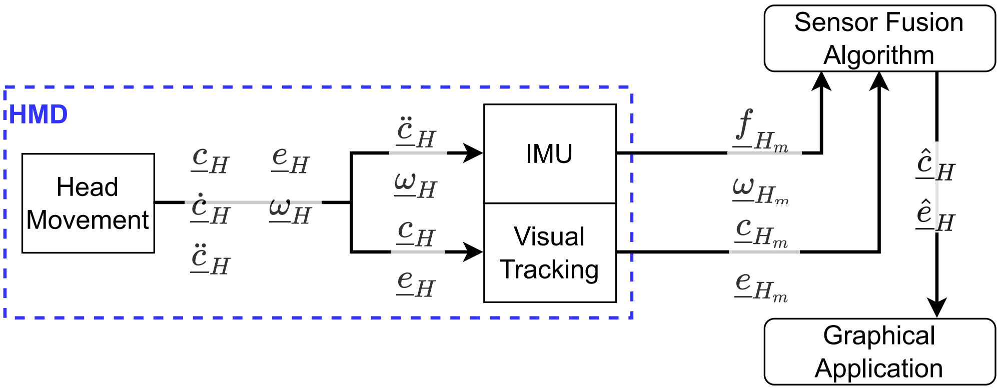


Figure 4.5: The information flow in an individual VR/AR system.

Most often the visual tracking results in an indirect measurement of the pose. In this case however, the visual tracking sensors under consideration measures the pose, $\underline{\boldsymbol{c}}_{H_m}$ and $\underline{\boldsymbol{e}}_{H_m}$, directly, the impact of this is further discussed in Section 4.5.2.

The IMU attached to the moving non-inertial frame $\mathbb{E}_H$, and a sensor observing the position $\underline{\boldsymbol{c}}_H$ and attitude $\underline{\boldsymbol{e}}_H$ of the moving, non-inertial, reference frame $\mathbb{E}_H$ validates both requirements. This allows the adaption of the shared state $\underline{\boldsymbol{x}}_\square$, the input $\underline{\boldsymbol{u}}_\square$, process noise $\underline{\boldsymbol{\nu}}_\square$, and compatible process model $\mathrm{f}_\square$ from Section 4.3.

The kinematics presented here will change slightly when the system is placed in the enclosed motion platform. Due to the independent design of the state $\underline{\boldsymbol{x}}_H$, nor the process model f nor the state $\underline{\boldsymbol{x}}_H$ will have to be changed in the next chapter. Only the observation model will need to be altered due to the user's and the visual tracking system's reference frame changing from the inertial $\mathbb{E}_I$ to the upper moving reference frame $\mathbb{E}_P$. However, the concept of the observation model, explained in this section, will stay the same.

4.5.1. Process Model

The adaption of the state $\underline{\boldsymbol{x}}_H$, the input $\underline{\boldsymbol{u}}_H$, the process noise $\underline{\boldsymbol{\nu}}_H$, and process model f_H is performed by replacing the placeholder $\square$ by the HMD indicator H throughout Section 4.3.

First, the **input** vector $\underline{\boldsymbol{u}}_H$ of the HMD process model are the measurements of the IMU attached to the HMD reference frame $\mathbb{E}_H$ and the center located between the retinas of the user.

Second, the **state** vector $\underline{\boldsymbol{x}}_H$ of the HMD is defined similarly as Equation 4.16. Whereby, the **position** $\underline{\boldsymbol{c}}_H$ of the HMD center point is expressed as the vector from the initialization position to the HMD center point itself, similar to Equation 4.17. Following, the **velocity** $\underline{\dot{\boldsymbol{c}}}_H$ of the point between the retinas is defined as the change in said position $\underline{\boldsymbol{c}}_H$ over time and also expressed in the inertial reference frame $\mathbb{E}_I$, again comparable to Equation 4.19. Moreover, the quaternion **attitude** $\underline{\boldsymbol{e}}_H$ is defined as the rotation from the inertial reference frame $\mathbb{E}_I$ to the moving reference frame $\mathbb{E}_H$ attached to the HMD, comparable to Equation 4.20. The **biases** $\underline{\boldsymbol{\lambda}}_H$ can be defined using Equation 4.22.

Last, having defined the relevant vectors the process model f_H is constructed in the same manner as performed in Section 4.3.3 in Equation 4.35.

4.5.2. Observation Model

The visual tracking sensor under consideration is modeled as simply measuring the position $\underline{\boldsymbol{c}}_{H_m}$ and attitude $\underline{\boldsymbol{e}}_{H_m}$ of the HMD directly. The **observation** vector $\underline{\boldsymbol{y}}_H$ can be noted as:

$$\underline{\boldsymbol{y}}_H = \begin{Bmatrix} \underline{\boldsymbol{c}}_{H_m} \\ \underline{\boldsymbol{e}}_{H_m} \end{Bmatrix} \tag{4.49}$$

As a side note for this section, the reason for this simplistic observation model is due to the fact that the visual tracking system under consideration has a proprietary abstraction layer that prevents the reading of the raw values. The abstraction layer, rather, computes the measured position $\underline{\boldsymbol{c}}_{H_m}$ and attitude $\underline{\boldsymbol{e}}_{H_m}$ of the HMD. This also causes for the noise model to be no longer Gaussian. More about the abstraction layer and the validity of the assumptions in this section can be found when the actual sensor is presented in Section 6.1.2 and in the experiment proposal in Chapter 8.

Measurements are never perfect, the corruption of the observation of the actual state by noise $\underline{\boldsymbol{n}}$ assumed Gaussian is used to model the imperfections. Even though it is expected that the abstraction layer of the visual tracking at hand already runs an estimation algorithm, noise variables need to be added. The addition of said noise variables is necessary for the integration of said measurements in any Kalman Filter (KF). A discussion about the validity of the Gaussian noise model and the consequences surrounding these assumption can be found in the presentation of the actual sensor in Section 6.1.2 and in the experiment proposal in Chapter 8. In this model, the Gaussian noise $\underline{\boldsymbol{n}}$ is added linearly to both the position $\underline{\boldsymbol{c}}_H$ and the attitude $\underline{\boldsymbol{e}}_H$, i.e.,

$$\begin{Bmatrix} \underline{\boldsymbol{c}}_{H_m} \\ \underline{\boldsymbol{e}}_{H_m} \end{Bmatrix} = \begin{Bmatrix} \underline{\boldsymbol{c}}_H + \underline{\boldsymbol{n}}_{\boldsymbol{c}_H} \\ \underline{\boldsymbol{e}}_H + \underline{\boldsymbol{n}}_{\boldsymbol{e}_H} \end{Bmatrix} \tag{4.50}$$

The equation allows the observation noise collection $\underline{\boldsymbol{o}}_H$ to be expressed as:

$$\underline{\boldsymbol{o}}_H = \begin{Bmatrix} \underline{\boldsymbol{n}}_{\boldsymbol{c}_H} \\ \underline{\boldsymbol{n}}_{\boldsymbol{e}_H} \end{Bmatrix} \tag{4.51}$$

In this individual case, the visual tracking measures the state $\underline{\boldsymbol{x}}_H$ of the HMD in the inertial reference frame $\mathbb{E}_I$ as it is attached to the room. Moreover, the software of the visual tracking sensor under consideration presents the attitude in Euler angles, in three sequential orthogonal rotations around the z-axis, y-axis, and x-axis, also called the yaw ψ_H, pitch θ_H, and roll ϕ_H formulation. Since both the position $\underline{\boldsymbol{c}}_H$ and the attitude $\underline{\boldsymbol{e}}_H$ were already defined in the inertial reference frame $\mathbb{E}_I$, see Equations 4.17 and 4.20; using the decomposition of the measurement terms in Equation 4.51; and the quaternion to Euler angle transformation found in [51], the observation model h_H of the VR/AR system can be written as:

$$\begin{aligned}
\underline{\boldsymbol{y}}_H(t) &= \mathrm{h}_H\left(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{o}}_H(t)\right) \\
&= \begin{Bmatrix} \underline{\boldsymbol{c}}_H + \underline{\boldsymbol{n}}_{\boldsymbol{c}_H} \\ \underline{\boldsymbol{e}}_H + \underline{\boldsymbol{n}}_{\boldsymbol{e}_H} \end{Bmatrix} \\
&= \begin{Bmatrix} x_H + n_{x_H} \\ y_H + n_{y_H} \\ z_H + n_{z_H} \\ \psi_H + n_{\psi_H} \\ \theta_H + n_{\theta_H} \\ \phi_H + n_{\phi_H} \end{Bmatrix} \\
&= \begin{Bmatrix} x_H + n_{x_H} \\ y_H + n_{y_H} \\ z_H + n_{z_H} \\ \text{arctan}\left(\left(e_{0_H} e_{x_H} + e_{y_H} + e_{z_H}\right), \left(0.5 - e_{x_H} - e_{y_H}\right)\right) + n_{\psi_H} \\ \text{arcsin}\left(2\left(e_{0_H} e_{y_H} - e_{x_H} e_{z_H}\right)\right) + n_{\theta_H} \\ \text{arctan}\left(\left(e_{0_H} e_{z_H} + e_{x_H} + e_{y_H}\right), \left(0.5 - e_{y_H} - e_{z_H}\right)\right) + n_{\phi_H} \end{Bmatrix}
\end{aligned} \tag{4.52}$$

Where arctan in the previous equation should be performed with the two-argument four-quadrant inverse tangent.

5

Configurations and their Joined Kinematics

In the previous chapter, Chapter 4, the individual kinematic models of both the Virtual Reality (VR)/Augmented Reality (AR) system and the Stewart platform were presented and discussed. In this chapter the individual kinematic models are combined to resemble a VR/AR system in an enclosed Stewart platform. The Unscented Kalman Filter (UKF), explained in Chapter 3, will be used to estimate the proper pose, position and attitude, of the Head-Mounted Display (HMD) with respect to (w.r.t.) the Stewart platform, in the next chapter, Chapter 7, in a model based simulation.

Integrating a VR/AR system in an enclosed Stewart platform is in essence a change in reference frame for the VR/AR system from the inertial $\mathbb{E}_I$ to the upper moving reference frame $\mathbb{E}_P$. Meaning that the pose from which to render the image for the user should be the pose of the HMD w.r.t. the Upper Gimbal Point (UGP) as seen from the upper moving frame $\mathbb{E}_P$ of the Stewart platform, i.e. the vehicle-fixed pose. This new vehicle-fixed pose is denoted as $\underline{\boldsymbol{c}}_{HP}$ and $\underline{\boldsymbol{e}}_{HP}$ for the position and attitude respectively.

However, the Inertial Measurement Unit (IMU) of the HMD will pick up both acceleration $\ddot{\underline{\boldsymbol{c}}}_P$ and angular rate $\underline{\boldsymbol{\omega}}_P$, of the Stewart platform alongside the vehicle-fixed acceleration $\ddot{\underline{\boldsymbol{c}}}_{HP}$ and angular rate $\dot{\underline{\boldsymbol{\omega}}}_{HP}$ of the user's HMD, indicated by the red dashed arrow in Figure 5.1. Even though the visual tracking, now attached to the upper moving reference frame $\mathbb{E}_P$, already measures the vehicle-fixed pose, $\underline{\boldsymbol{c}}_{HP_m}$ and $\underline{\boldsymbol{e}}_{HP_m}$, a standard sensor fusion algorithm deployed for the pose estimation of a nominal HMD has no notion of the influence of the Stewart platform on the HMD's IMU, resulting in a distorted pose. In turn, distorting the image shown to the user, rendering this VR/AR system setup unusable.

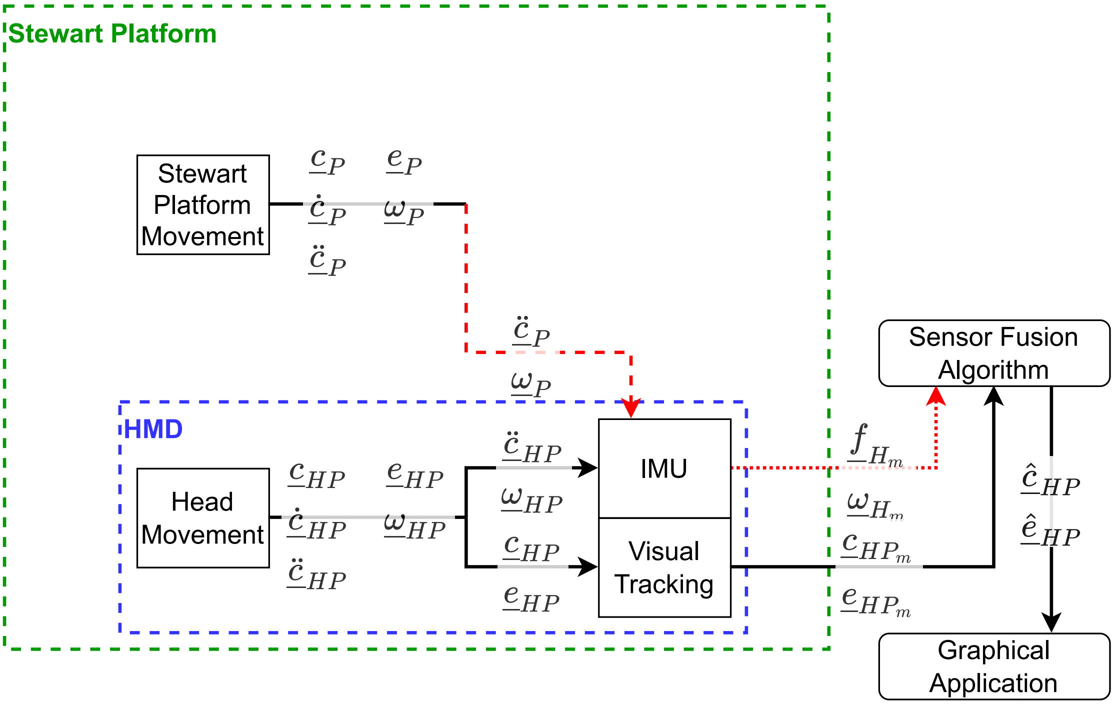


Figure 5.1: The result of placing a VR/AR system in an enclosed Stewart platform. The red dashed and dotted arrows indicating the IMU of the HMD picking up the motion of the Stewart platform

In order to fix the distorted pose three configurations alongside a baseline will be presented and the kinematics discussed that are capable in combination with the UKF to estimate the vehicle-fixed pose, $\underline{\hat{c}}_{HP}$ and $\underline{\hat{e}}_{HP}$, of the HMD w.r.t. the UGP of a Stewart platform. A baseline configuration, Configuration 0, is presented in Section 5.1 and uses only the visual tracking to which the other three configurations can be compared to. Followed by Configuration 2, in Section 5.2, which enables the IMU on the HMD for more precise and lower latency pose estimation compared to the baseline, Configuration 0. Moreover, Configuration 1 also estimates the pose, position $\underline{\hat{c}}_P$ and attitude $\underline{\hat{e}}_P$, of the Stewart platform using only the measured lengths of its hydraulic actuators. Next, Configuration 2, explained in Section 5.3, extends Configuration 1 by enabling the use of the IMU attached to the Stewart platform. The aim of Configuration 2 is to investigate the influence of an improved accuracy pose estimation of the Stewart platform on the pose estimation of the HMD. Last, Configuration 3, will try something novel, namely the use of software based setpoints of the Stewart platform to predict the pose of the Stewart platform as a replacement of the IMU in Configuration 2. The software based setpoints were explained in Section 2.3.3.

5.1. Configuration 0 --- Baseline

The visual tracking, when attached to the moving Stewart platform, already measures the proper pose of the HMD as stated in the introduction of this chapter. By disabling the IMU of the HMD the problem highlighted in Figure 5.1 is removed. However, this also removes the fast pose updates the IMU can provide. This configuration is the minimal sensor configuration to achieve vehicle-fixed pose estimations of a VR/AR system within an enclosed motion platform and will serve as the baseline to which the other configurations in this chapter can be compared to.

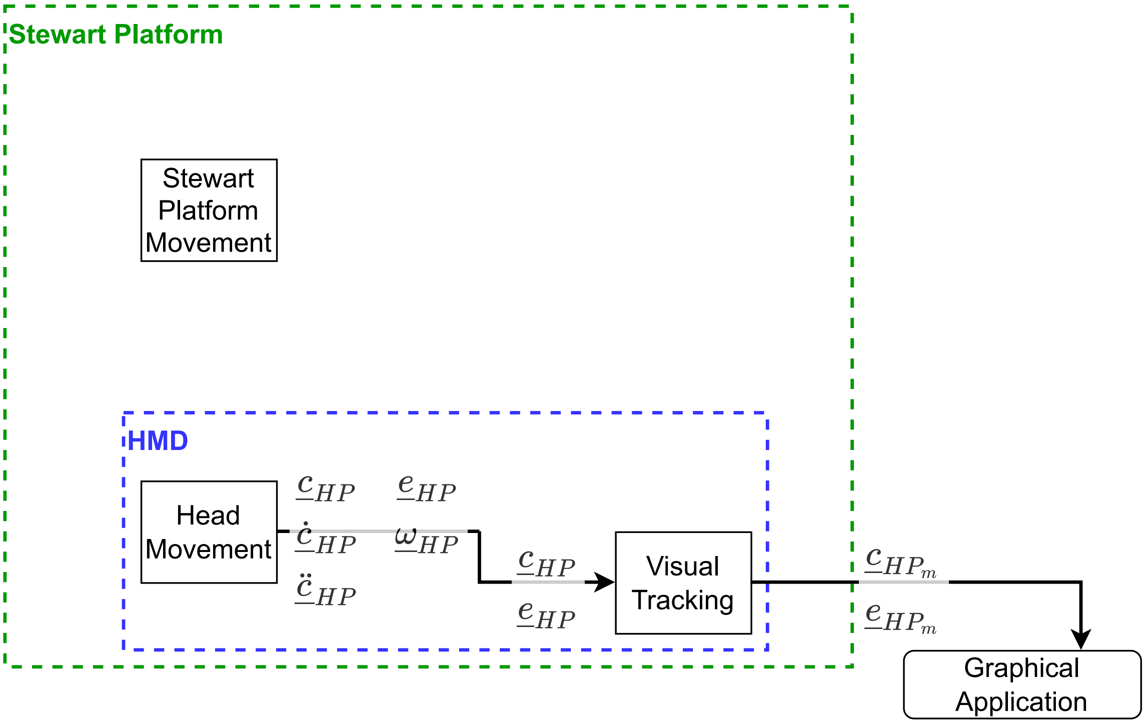


Figure 5.2: Configuration 1: A VR/AR system in an enclosed Stewart platform, with only measuring the pose of the HMD.

5.1.1. Kinematics

The visual tracking, under considerations, has a proprietary abstraction layer that prevents the reading of the raw underlying measurement values, as stated in Section 4.5.2. The abstraction layer already measures and estimates the vehicle-fixed position $\underline{\boldsymbol{c}}_{HP_m}$ and attitude $\underline{\boldsymbol{e}}_{HP_m}$ directly, i.e.,

$$\underline{\boldsymbol{y}}_{HP} = \begin{bmatrix} \underline{\boldsymbol{c}}_{HP_m} \\ \underline{\boldsymbol{e}}_{HP_m} \end{bmatrix} \tag{5.1}$$

Note again the new subscript HP that indicates a variable of the HMD w.r.t. the upper moving reference frame $\mathbb{E}_P$.

No process model is used in this case, as it is expected that the abstraction layer already runs a pose estimation algorithm on the raw measurements of the sensor. Applying a Kalman Filter (KF) on top of another pose estimation algorithm without the addition of other sensors information will not improve the outcome [31]. Thus, the measurements are taken as is.

5.2. Configuration 1 --- IMU on HMD

Configuration 1, depicted in Figure 5.3, enables the use of the IMU on the HMD in order to enable improved accuracy pose estimation by including the Stewart platform in the kinematics of the sensor fusion algorithm. Due to the design of the state, only the observation model of the VR/AR system presented in Section 4.5 is altered in Section 5.2.2 to resemble the aforementioned change in reference frame of the VR/AR system. The change of the Stewart platform process model f_P, presented in Section 4.4.1, is due to the removal of the IMU attached to the Stewart platform, invalidating the previously used general kinematics and is altered in Section 5.2.1.

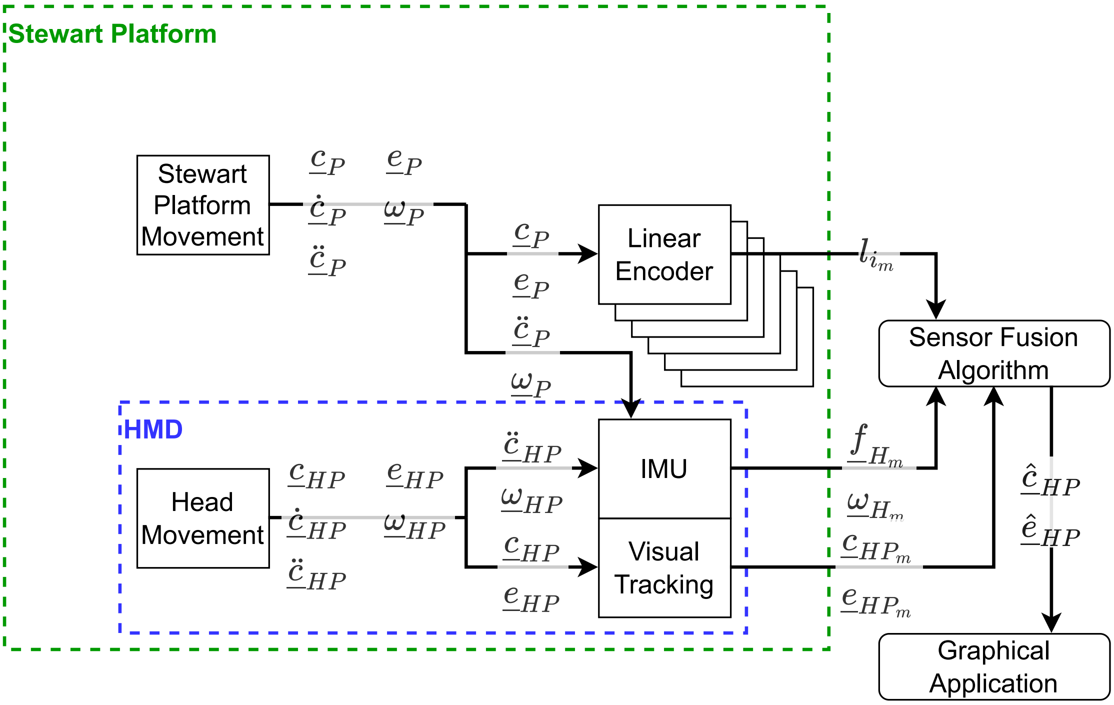

Stewart Platform
Stewart Platform Movement
$\underline{c}_P$ $\underline{e}_P$
$\dot{\underline{c}}_P$ $\underline{\omega}_P$
$\ddot{\underline{c}}_P$
$\underline{c}_P$
$\underline{e}_P$
$\ddot{\underline{c}}_P$
$\underline{\omega}_P$
Linear Encoder
l_{i_m}
Sensor Fusion Algorithm
HMD
Head Movement
$\underline{c}_{HP}$ $\underline{e}_{HP}$
$\dot{\underline{c}}_{HP}$ $\underline{\omega}_{HP}$
$\ddot{\underline{c}}_{HP}$
$\ddot{\underline{c}}_{HP}$
$\underline{\omega}_{HP}$
$\underline{c}_{HP}$
$\underline{e}_{HP}$
IMU
Visual Tracking
$\underline{f}_{H_m}$
$\underline{\omega}_{H_m}$
$\underline{c}_{HP_m}$
$\underline{e}_{HP_m}$
$\hat{\underline{c}}_{HP}$
$\hat{\underline{e}}_{HP}$
Graphical Application

Figure 5.3: Configuration 1: A VR/AR system with an IMU in an enclosed Stewart platform. The kinematics are adapted by also estimating the Stewart platform pose.

5.2.1. Stewart Platform Process Model Alteration

The change in the state $\dot{\underline{x}}_P$ of the Stewart platform can be seen as Gaussian noise on said system and can be included as part of the process noise $\underline{\nu}_P$ [31], especially when no other measurements exist to express the change in the state $\dot{\underline{x}}_P$. The acceleration $\ddot{\underline{c}}_P$ and the angular acceleration $\dot{\underline{\omega}}_P$ of the Stewart platform are modeled as Gaussian noise, leading to a second-order system [31]. Expressing the acceleration $\underline{\nu}_{\ddot{c}_P}$ and angular acceleration noise $\underline{\nu}_{\dot{\omega}_P}$ in the inertial $\mathbb{E}_I$ and upper moving frame $\mathbb{E}_P$ respectively, the process noise $\underline{\nu}_P$ is redefined as:

$$\underline{\nu}_P = \begin{bmatrix} \underline{\nu}_{\ddot{c}_P} \\ \underline{\nu}_{\dot{\omega}_P} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \nu_{\ddot{x}_P} & \nu_{\ddot{y}_P} & \nu_{\ddot{z}_P} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \nu_{\dot{p}_P} & \nu_{\dot{q}_P} & \nu_{\dot{r}_P} \end{pmatrix} \{\mathbb{E}_P\} \end{bmatrix} \tag{5.2}$$

The proportionality of the acceleration $\underline{\nu}_{\ddot{c}_P}$ and the angular acceleration noise $\underline{\nu}_{\dot{\omega}_P}$ is investigated further in Section 7.5.

In this case, the redefined process noise $\underline{\nu}_P$ can be seen as a replacement for the removed IMU. One that measures angular acceleration $\dot{\underline{\omega}}_{P_m}$ instead of rate $\underline{\omega}_{P_m}$ and acceleration $\ddot{\underline{c}}_{P_m}$ in the inertial reference frame $\mathbb{E}_I$ instead of specific force $\underline{f}_{P_m}$ in the upper moving reference frame $\mathbb{E}_P$. Moreover, the redefined process noise $\underline{\nu}_P$ can be seen as a replacement of the removed IMU that measures zero constantly and thus has large noise values for $\underline{\nu}_{\ddot{c}_P}$ and $\underline{\nu}_{\dot{\omega}_P}$ to make up for it.

The modified input causes the state $\underline{x}_P$, previously presented in Equation 4.16, to change in order to incorporate the angular acceleration $\dot{\underline{\omega}}_P$ later in the process model f_P by adding the angular rate $\underline{\omega}_P$.

However, the bias terms $\underline{\boldsymbol{\lambda}}_P$ found in Equation 4.16, all belonging to the IMU can be removed, resulting in the redefined state $\underline{\boldsymbol{x}}_P$ of the Stewart platform, i.e.,

$$\underline{\boldsymbol{x}}_P = \begin{bmatrix} \underline{\boldsymbol{c}}_P \\ \underline{\dot{\boldsymbol{c}}}_P \\ \underline{\boldsymbol{e}}_P \\ \underline{\boldsymbol{\omega}}_P \end{bmatrix} \tag{5.3}$$

The angular rate $\underline{\boldsymbol{\omega}}_P$ part of the state $\underline{\boldsymbol{x}}_P$ in Equation 5.3 is defined similar to the measurements of the IMU, i.e. defined in the upper moving frame $\mathbb{E}_P$, i.e.,

$$\underline{\boldsymbol{\omega}}_P = \begin{pmatrix} p_P & q_P & r_P \end{pmatrix} \{\mathbb{E}_P\} \tag{5.4}$$

To create this configuration's process model f_P the derivative of the state $\underline{\dot{\boldsymbol{x}}}_P$ is taken. With the redefinition of the angular rate $\underline{\boldsymbol{\omega}}_P$ in Equation 5.4, the derivative of the quaternion attitude $\underline{\dot{\boldsymbol{e}}}_P$ can be simplified, by removing the IMU related variables, from Equation 4.33 to:

$$\underline{\dot{\boldsymbol{e}}}_P = \begin{Bmatrix} \dot{e}_{0_P} \\ \dot{e}_{x_P} \\ \dot{e}_{y_P} \\ \dot{e}_{z_P} \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} -e_{x_P} & -e_{y_P} & -e_{z_P} \\ e_{0_P} & -e_{z_P} & e_{y_P} \\ e_{z_P} & e_{0_P} & -e_{x_P} \\ -e_{y_P} & e_{x_P} & e_{0_P} \end{bmatrix} \begin{Bmatrix} p_P \\ q_P \\ r_P \end{Bmatrix} \tag{5.5}$$

The change of the angular rate $\underline{\dot{\boldsymbol{\omega}}}_P$ over time, also called the angular acceleration, is expressed by the time derivative of Equation 5.4, i.e.,

$$\underline{\dot{\boldsymbol{\omega}}}_P = \begin{pmatrix} \dot{p}_P & \dot{q}_P & \dot{r}_P \end{pmatrix} \{\mathbb{E}_P\} + \begin{pmatrix} p_P & q_P & r_P \end{pmatrix} \{\dot{\mathbb{E}}_P\} \tag{5.6}$$

Here, the $\{\dot{\mathbb{E}}_P\}$ is a column notation of the time derivative of the unit vectors of the upper moving reference frame $\mathbb{E}_P$. The direction change of the upper moving reference frame $\mathbb{E}_P$ is due to the angular rate of the object in question and can be expressed as:

$$= \begin{pmatrix} \dot{p}_P & \dot{q}_P & \dot{r}_P \end{pmatrix} \{\mathbb{E}_P\} + \begin{pmatrix} p_P & q_P & r_P \end{pmatrix} [\boldsymbol{\Omega}_P] \{\mathbb{E}_P\}$$

The $[\boldsymbol{\Omega}_P]$ is called the angular operator and is a skew-symmetrical matrix composed of the angular rate $\underline{\boldsymbol{\omega}}_P$. A matrix composition to express the vector in a rotating reference frame by taking the cross-product of a vector and the rotation itself, i.e.,

$$= \begin{pmatrix} \dot{p}_P & \dot{q}_P & \dot{r}_P \end{pmatrix} \{\mathbb{E}_P\} + \begin{pmatrix} p_P & q_P & r_P \end{pmatrix} \begin{bmatrix} 0 & r_P & -q_P \\ -r_P & 0 & p_P \\ q_P & -p_P & 0 \end{bmatrix} \{\mathbb{E}_P\}$$

The vector, expressed in the $\mathbb{E}_P$ frame, of the cross product also being the rotation on said moving non-inertial frame is equal to zero, i.e.,

$$= \begin{pmatrix} \dot{p}_P & \dot{q}_P & \dot{r}_P \end{pmatrix} \{\mathbb{E}_P\} \tag{5.7}$$

The process model f_P can be written by taking the process model $\mathrm{f}_\square$ in Equation 4.35 and applying the changes presented above in Equations 5.2 to 5.7. This results in:

$$\begin{gathered} \underline{\dot{\boldsymbol{x}}}_P(t) = \mathrm{f}_P\left(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{\nu}}_P(t)\right) \\ \begin{Bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \\ \ddot{x}_P \\ \ddot{y}_P \\ \ddot{z}_P \\ \dot{e}_{0_P} \\ \dot{e}_{x_P} \\ \dot{e}_{y_P} \\ \dot{e}_{z_P} \\ \dot{p}_P \\ \dot{q}_P \\ \dot{r}_P \end{Bmatrix} = \begin{Bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \\ \nu_{\ddot{x}_P} \\ \nu_{\ddot{y}_P} \\ \nu_{\ddot{z}_P} \\ \frac{1}{2} \begin{bmatrix} -e_{x_P} & -e_{y_P} & -e_{z_P} \\ e_{0_P} & -e_{z_P} & e_{y_P} \\ e_{z_P} & e_{0_P} & -e_{x_P} \\ -e_{y_P} & e_{x_P} & e_{0_P} \end{bmatrix} \begin{Bmatrix} p_P \\ q_P \\ r_P \end{Bmatrix} \\ \nu_{\dot{p}_P} \\ \nu_{\dot{q}_P} \\ \nu_{\dot{r}_P} \end{Bmatrix} \end{gathered} \tag{5.8}$$

Note that there is no external input to this equation, here, only process noise $\underline{\boldsymbol{\nu}}_P$ is assumed accountable for the systems movement.

The Stewart platform observation model h_P, defined in Section 4.4.2 in Equation 4.44 is left unaltered as nor the position $\underline{\boldsymbol{c}}_P$ nor the attitude $\underline{\boldsymbol{e}}_P$ was redefined.

5.2.2. VR/AR Observation Model Alteration

The HMD process model f_H as found in Section 4.5 is also left unaltered. This is true as the HMD's IMU still measures the specific force $\underline{\boldsymbol{f}}_H$ and angular rate $\underline{\boldsymbol{\omega}}_H$ of the HMD frame $\mathbb{E}_H$ w.r.t. the inertial frame $\mathbb{E}_I$. However, the observation h_H needs alteration as the visual tracking is no longer attached to the inertial frame but to the upper moving frame $\mathbb{E}_P$ of the Stewart platform. The position $\underline{\boldsymbol{c}}_{HP_m}$ and attitude $\underline{\boldsymbol{e}}_{HP_m}$ measured by the visual tracking of the HMD are in this case w.r.t. to the upper moving frame $\mathbb{E}_P$ of the Stewart platform. Both HMD and Stewart platform state's position and attitude are expressed w.r.t. the inertial frame $\mathbb{E}_I$. Thus, both HMD and Stewart platform poses need to be deducted to express the position $\underline{\boldsymbol{c}}_{HP_m}$ and attitude $\underline{\boldsymbol{e}}_{HP_m}$ measured by the visual tracking. This leads to a definition of the observation model h_{HP} of the VR/AR system as:

$$\underline{\boldsymbol{y}}_{HP} = \mathrm{h}_{HP}\left(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{o}}_H(t)\right) \tag{5.9}$$

$$\begin{bmatrix} \underline{\boldsymbol{c}}_{HP_m} \\ \underline{\boldsymbol{e}}_{HP_m} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{c}}_{HP} + \underline{\boldsymbol{n}}_{\boldsymbol{c}_H} \\ \underline{\boldsymbol{e}}_{HP} + \underline{\boldsymbol{n}}_{\boldsymbol{e}_H} \end{bmatrix} \tag{5.10}$$

Here, in Equation 5.10, the observation noise $\underline{\boldsymbol{o}}_H$ existing out of the positional $\underline{\boldsymbol{n}}_{\boldsymbol{c}_H}$ and attitude $\underline{\boldsymbol{n}}_{\boldsymbol{e}_H}$ Gaussian noise vectors, are left unaltered since their inception in Section 4.5.2. As of note, this observation h_{HP} is the only time the relative motion is relevant and both systems actively interact in the kinematic models.

First, the position is defined from the center of the upper moving frame $\mathbb{E}_P$ of the Stewart platform, the UGP, to between the users retinas, the center of the HMD frame of reference $\mathbb{E}_H$. This can be expressed as the difference between the position of the HMD $\underline{\boldsymbol{c}}_H$ and the Stewart platform $\underline{\boldsymbol{c}}_P$, i.e.,

$$\underline{\boldsymbol{c}}_{HP} = \underline{\boldsymbol{c}}_H - \underline{\boldsymbol{c}}_P \tag{5.11}$$

Expressing $\underline{\boldsymbol{c}}_{HP}$ in the upper moving frame $\mathbb{E}_P$, while using Equation 4.17 to express both positions, $\underline{\boldsymbol{c}}_H$ and $\underline{\boldsymbol{c}}_P$, in the inertial frame $\mathbb{E}_I$ Equation 5.11 can be expanded to:

$$\begin{pmatrix} x_{HP} & y_{HP} & z_{HP} \end{pmatrix} \{\mathbb{E}_P\} = \begin{pmatrix} x_H - x_P & y_H - y_P & z_H - z_P \end{pmatrix} \{\mathbb{E}_I\}$$

Continuing, the right side of the previous equation is transformed, using Equation 4.21, to the upper moving frame $\mathbb{E}_P$ to concur with the measurement, i.e.,

$$= \begin{pmatrix} x_H - x_P & y_H - y_P & z_H - z_P \end{pmatrix} \boldsymbol{T}_{PI}^T \{\mathbb{E}_P\} \tag{5.12}$$

The attitude $\underline{\boldsymbol{e}}_{HP}$ follows a similar method as in Section 4.5.2. The Euler-angles, ψ_{HP}, θ_{HP}, and ϕ_{HP}, in which the visual tracking under considerations expresses itself are defined as the attitude from the upper moving frame $\mathbb{E}_P$ to the reference frame $\mathbb{E}_H$ attached to the HMD. Using the same quaternion to Euler-angle transformation as in Equation 4.52 and found in [51], the Euler-angles can be written as:

$$\begin{Bmatrix} \psi_{HP} \\ \theta_{HP} \\ \phi_{HP} \end{Bmatrix} = \begin{Bmatrix} \arctan\left(\left(e_{0_{HP}} e_{x_{HP}} + e_{y_{HP}} + e_{z_{HP}}\right), \left(0.5 - e_{x_{HP}} - e_{y_{HP}}\right)\right) \\ \arcsin\left(2\left(e_{0_{HP}} e_{y_{HP}} - e_{x_{HP}} e_{z_{HP}}\right)\right) \\ \arctan\left(\left(e_{0_{HP}} e_{z_{HP}} + e_{x_{HP}} + e_{y_{HP}}\right), \left(0.5 - e_{y_{HP}} - e_{z_{HP}}\right)\right) \end{Bmatrix} \tag{5.13}$$

Where the arctan function of the previous equation, Equation 5.13, should be performed with the two-argument four-quadrant inverse tangent.

The quaternion representation of the attitude $\underline{\boldsymbol{e}}_{HP}$ used in Equation 5.13 is defined as the attitude of the HMD attached frame $\mathbb{E}_H$ from the upper moving frame $\mathbb{E}_P$, i.e.,

$$\underline{\boldsymbol{e}}_{HP} = \begin{pmatrix} e_{0_{HP}} & e_{x_{HP}} & e_{y_{HP}} & e_{z_{HP}} \end{pmatrix} \begin{Bmatrix} 1 \\ \mathbb{E}_P \end{Bmatrix} \tag{5.14}$$

The computation of said quaternion $\underline{\boldsymbol{e}}_{HP}$ is actually quite similar to its positional counterpart $\underline{\boldsymbol{c}}_{HP}$. Meaning, the Stewart platform attitude $\underline{\boldsymbol{e}}_{P}$ is subtracted from the HMD attitude $\underline{\boldsymbol{e}}_{H}$. Only, the quaternion counterpart of a subtraction is a counter rotation. The counter rotation of a quaternion is indicated by $\sim$ and is performed by switching the signs of only the vector part of the quaternion, i.e.,

$$\begin{aligned}\sim\underline{\boldsymbol{e}}_{HP} &= \begin{pmatrix} e_{0_{HP}} & e_{x_{HP}} & e_{y_{HP}} & e_{z_{HP}} \end{pmatrix} \begin{Bmatrix} 1 \\ -\mathbb{E}_P \end{Bmatrix} \\ &= \begin{pmatrix} e_{0_{HP}} & -e_{x_{HP}} & -e_{y_{HP}} & -e_{z_{HP}} \end{pmatrix} \begin{Bmatrix} 1 \\ \mathbb{E}_P \end{Bmatrix} \end{aligned} \tag{5.15}$$

The counter rotation of the Stewart platform $\sim\underline{\boldsymbol{e}}_{P}$ is 'added' to the HMD attitude $\underline{\boldsymbol{e}}_{H}$ via quaternion multiplication $\underset{L}{\otimes}$, previously used in Equation 4.31, i.e.,

$$\underline{\boldsymbol{e}}_{HP} = \underline{\boldsymbol{e}}_{H} \underset{L}{\otimes} \left(\sim\underline{\boldsymbol{e}}_{P}\right) \tag{5.16}$$

$$\begin{aligned}\begin{Bmatrix} e_{0_{HP}} \\ e_{x_{HP}} \\ e_{y_{HP}} \\ e_{z_{HP}} \end{Bmatrix} &= \begin{bmatrix} e_{0_H} & -e_{x_H} & -e_{y_H} & -e_{z_H} \\ e_{x_H} & e_{0_H} & -e_{z_H} & e_{y_H} \\ e_{y_H} & e_{z_H} & e_{0_H} & -e_{x_H} \\ e_{z_H} & -e_{y_H} & e_{x_H} & e_{0_H} \end{bmatrix} \begin{Bmatrix} e_{0_P} \\ -e_{x_P} \\ -e_{y_P} \\ -e_{z_P} \end{Bmatrix} \\ &= \begin{Bmatrix} e_{0_H}e_{0_P} + e_{x_H}e_{x_P} + e_{y_H}e_{y_P} + e_{z_H}e_{z_P} \\ e_{x_H}e_{0_P} - e_{0_H}e_{x_P} + e_{z_H}e_{y_P} - e_{y_H}e_{z_P} \\ e_{y_H}e_{0_P} - e_{z_H}e_{x_P} - e_{0_H}e_{y_P} + e_{x_H}e_{z_P} \\ e_{z_H}e_{0_P} + e_{y_H}e_{x_P} - e_{x_H}e_{y_P} - e_{0_H}e_{z_P} \end{Bmatrix} \end{aligned} \tag{5.17}$$

The observation model h_{HP} of the enclosed HMD in a Stewart platform can be constructed by combining: the assumed influence of the noise variables, $\underline{\boldsymbol{n}}_{\boldsymbol{c}_H}$ and $\underline{\boldsymbol{n}}_{\boldsymbol{e}_H}$, found in Equation 4.52; the expression for the position $\underline{\boldsymbol{c}}_{HP}$ of the HMD to the UGP, stated in Equation 5.12; and the transformation of the quaternion representation to the Euler-angles representation of the attitude $\underline{\boldsymbol{e}}_{HP}$ in Equation 5.13:

$$\begin{aligned}\underline{\boldsymbol{y}}_{HP}(t) &= \mathrm{h}_{HP}\left(\underline{\boldsymbol{x}}_{H}(t), \underline{\boldsymbol{x}}_{P}(t), \underline{\boldsymbol{o}}_{H}(t)\right) \\ \begin{Bmatrix} \underline{\boldsymbol{c}}_{HP} + \underline{\boldsymbol{n}}_{\boldsymbol{c}_H} \\ \underline{\boldsymbol{e}}_{HP} + \underline{\boldsymbol{n}}_{\boldsymbol{e}_H} \end{Bmatrix} &= \begin{Bmatrix} x_{HP} \\ y_{HP} \\ z_{HP} \\ \psi_{HP} \\ \theta_{HP} \\ \phi_{HP} \end{Bmatrix} + \begin{Bmatrix} n_{x_H} \\ n_{y_H} \\ n_{z_H} \\ n_{\psi_H} \\ n_{\theta_H} \\ n_{\phi_H} \end{Bmatrix} \\ &= \begin{Bmatrix} \boldsymbol{T}_{PI} \begin{Bmatrix} x_H - x_P \\ y_H - y_P \\ z_H - z_P \end{Bmatrix} \\ \text{arctan}\left(\left(e_{0_{HP}}e_{x_{HP}} + e_{y_{HP}} + e_{z_{HP}}\right), \left(0.5 - e_{x_{HP}} - e_{y_{HP}}\right)\right) \\ \text{arcsin}\left(2\left(e_{0_{HP}}e_{y_{HP}} - e_{x_{HP}}e_{z_{HP}}\right)\right) \\ \text{arctan}\left(\left(e_{0_{HP}}e_{z_{HP}} + e_{x_{HP}} + e_{y_{HP}}\right), \left(0.5 - e_{y_{HP}} - e_{z_{HP}}\right)\right) \end{Bmatrix} + \begin{Bmatrix} n_{x_H} \\ n_{y_H} \\ n_{z_H} \\ n_{\psi_H} \\ n_{\theta_H} \\ n_{\phi_H} \end{Bmatrix} \end{aligned} \tag{5.18}$$

5.3. Configuration 2 --- IMU Attached to Stewart platform

The influence of the precision of the estimated pose of the Stewart platform on the moving frame-fixed pose estimation of the HMD is looked into by Configuration 2. This configuration is similar to Configuration 1 but has an IMU attached to the upper moving frame of the Stewart platform as depicted in Figure 5.4.

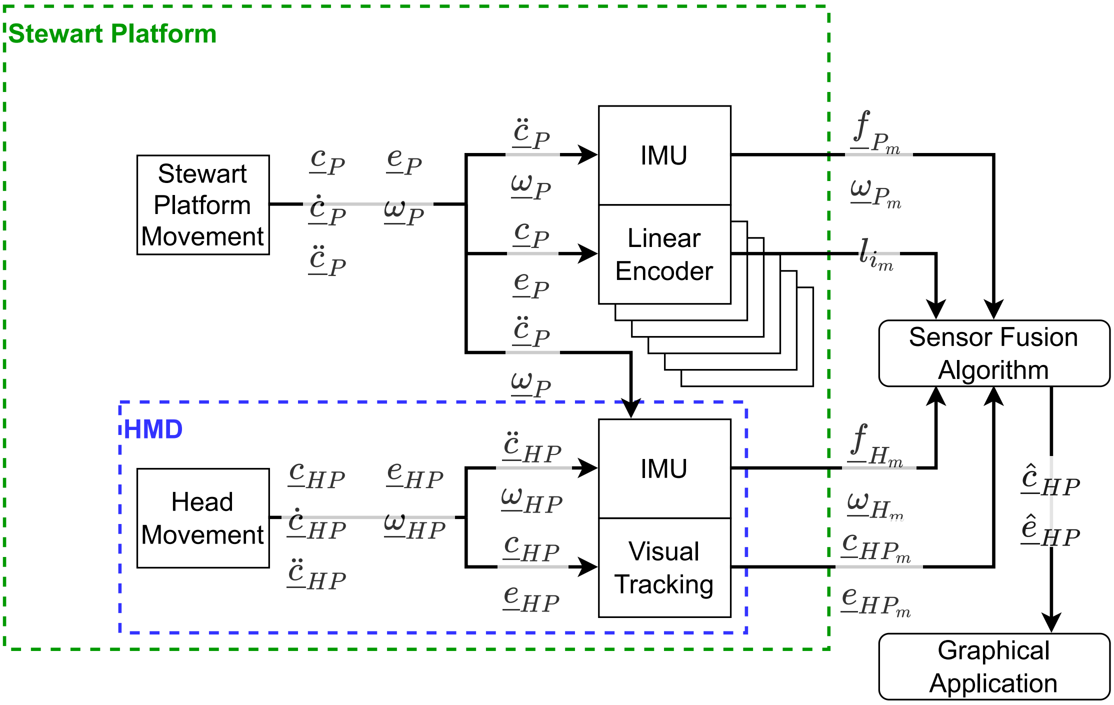


Figure 5.4: Configuration 2: Configuration 1, depicted in Figure 5.3, with an IMU attached to the upper moving frame of the Stewart platform.

The fact that the Stewart platform again has an IMU attached to its non-inertial reference frame $\mathbb{E}_\square$ means that it can use the general kinematics in this configuration. Following, the state $\underline{\boldsymbol{x}}_P$ and the process model f_P was defined in Section 4.3.3. Moreover, the observation model h_P of the Stewart platform is the same as in Equation 4.44.

Nothing changes to the VR/AR system so nothing changes to the kinematic model. The state $\underline{\boldsymbol{x}}_H$, the process model f_H are as developed in Section 4.3.3 The observation model h_{HP} of the HMD in the enclosed Stewart platform was discussed in Section 5.2.2 and detailed in Equation 5.18.

The resulting kinematic model has again two seemingly independent states, the Stewart platform $\underline{\boldsymbol{x}}_P$ and HMD $\underline{\boldsymbol{x}}_H$ states. However, in the VR/AR system observation h_{HP} the systems interact direct mathematically and the relative motion between both systems becomes relevant.

5.4. Configuration 3 --- SRS with Setpoints

Contrary to Configuration 2, Configuration 3 uses the setpoints $\underline{s}$ of the Stewart platform motion system to predict the future pose of the Stewart platform itself, indicated in Figure 5.5.

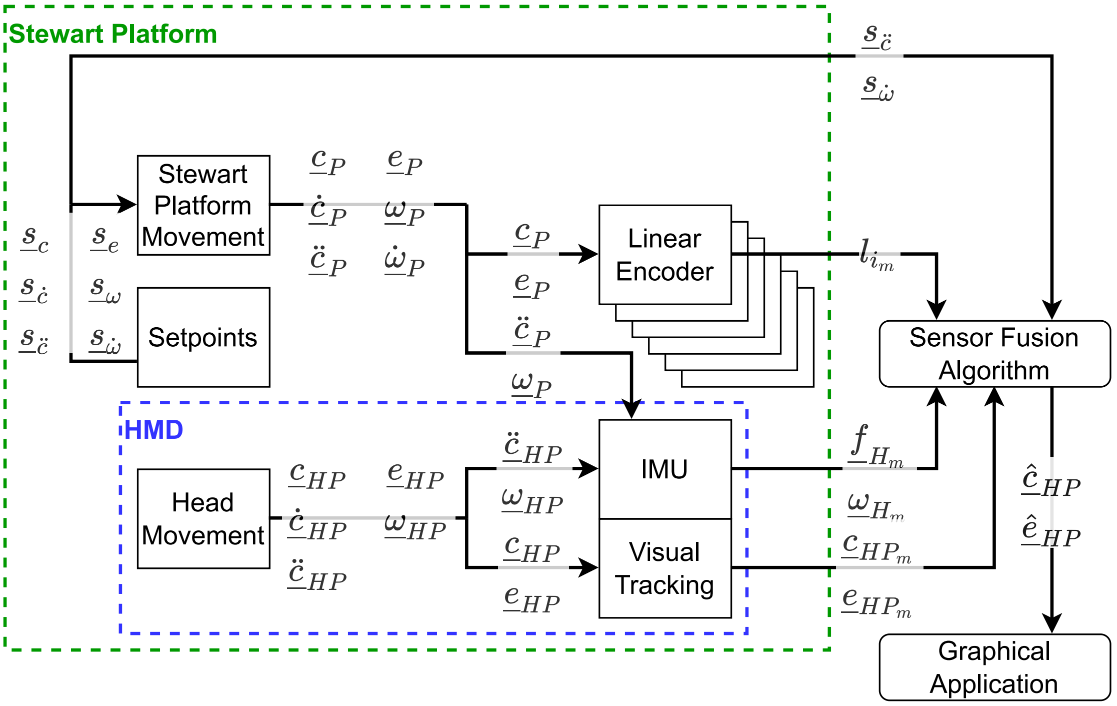

Stewart Platform
$\underline{s}_{\ddot{c}}$
$\underline{s}_{\dot{\omega}}$
Stewart Platform Movement
$\underline{c}_P$ $\underline{e}_P$
$\underline{\dot{c}}_P$ $\underline{\omega}_P$
$\underline{\ddot{c}}_P$ $\underline{\dot{\omega}}_P$
$\underline{s}_c$ $\underline{s}_e$
$\underline{s}_{\dot{c}}$ $\underline{s}_\omega$
$\underline{s}_{\ddot{c}}$ $\underline{s}_{\dot{\omega}}$
Setpoints
$\underline{c}_P$
$\underline{e}_P$
$\underline{\ddot{c}}_P$
$\underline{\omega}_P$
Linear Encoder
$\underline{l}_{i_m}$
Sensor Fusion Algorithm
HMD
Head Movement
$\underline{c}_{HP}$ $\underline{e}_{HP}$
$\underline{\dot{c}}_{HP}$ $\underline{\omega}_{HP}$
$\underline{\ddot{c}}_{HP}$
$\underline{\ddot{c}}_{HP}$
$\underline{\omega}_{HP}$
$\underline{c}_{HP}$
$\underline{e}_{HP}$
IMU
Visual Tracking
$\underline{f}_{H_m}$
$\underline{\omega}_{H_m}$
$\underline{c}_{HP_m}$
$\underline{e}_{HP_m}$
$\underline{\hat{c}}_{HP}$
$\underline{\hat{e}}_{HP}$
Graphical Application

Figure 5.5: Configuration 3: similar to Configuration 1, shown in Figure 5.3, with the setpoints used to help predict the pose of the Stewart platform.

Also depicted in Figure 5.5, are the full set of setpoints available, namely the position $\underline{s}_c$, velocity $\underline{s}_{\dot{c}}$, acceleration $\underline{s}_{\ddot{c}}$, attitude $\underline{s}_e$, angular rate $\underline{s}_\omega$, and angular acceleration $\underline{s}_{\dot{\omega}}$. Of which only the acceleration positional $\underline{s}_{\ddot{c}}$ and the angular acceleration positional $\underline{s}_{\dot{\omega}}$ are only used due to their higher order relation to elements of the state.

This configuration investigates if software based setpoints of the Stewart platform can be used to replace the IMU as a sensor and the possible impact it would have on the proper pose of the HMD enclosed in said Stewart platform. The added information should improve the pose estimation of the Stewart platform when compared to Configuration 1. When compared to Configuration 2, it is the question how Gaussian and accurate the setpoints based process model and the added variables are when compared to the IMU process model and variables.

5.4.1. Joined Kinematics

Similar to Configurations 1 and 2, the HMD process model f_H, the HMD observation model h_{HP}, and the Stewart platform observation model h_P can be found in Equations 4.35, 5.18, and 4.44 respectively.

Koekebakker states in [29] that a motion system with *'an inner loop feedback and feedback linearising control result in first order response of the system, from desired to actual accelerations.'* Following, a motion system actual acceleration and angular acceleration can be described by the transfer function $\mathrm{G}(s)$ as:

$$\mathrm{G}(s) = \frac{\mathrm{Y}(s)}{\mathrm{U}(s)} = \frac{1}{\tau s + 1} = \frac{1/\tau}{s + 1/\tau} \tag{5.19}$$

Where, the input $\mathrm{U}(s)$ comprises of the setpoints describing the desired second derivative of the system pose, the output $\mathrm{Y}(s)$ expresses the actual acceleration of the system, both are based on the complex variable s, and a time lag τ that indicates the time the system takes to reach 63 % of the final step input.

When applied to the second derivative setpoints $\underline{\boldsymbol{s}}$, namely the desired acceleration $\underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}}$ and desired angular acceleration $\underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}}$, should result in the actual second derivatives, the acceleration $\ddot{\underline{\boldsymbol{c}}}_P$ and angular acceleration $\dot{\underline{\boldsymbol{\omega}}}_P$ of the Stewart platform. To integrate the actual acceleration $\ddot{\underline{\boldsymbol{c}}}_P$ and angular acceleration $\dot{\underline{\boldsymbol{\omega}}}_P$ into the process model f_P, the transfer function $\mathrm{G}(s)$, in Equation 5.19, is expressed in a state-space form, i.e.,

$$\dddot{\underline{\boldsymbol{c}}}_P = \frac{1}{\underline{\boldsymbol{\tau}}_{\ddot{\boldsymbol{c}}}} \left(\underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}} - \ddot{\underline{\boldsymbol{c}}}_P \right) \tag{5.20}$$

$$\ddot{\underline{\boldsymbol{\omega}}}_P = \frac{1}{\underline{\boldsymbol{\tau}}_{\dot{\boldsymbol{\omega}}}} \left(\underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}} - \dot{\underline{\boldsymbol{\omega}}}_P \right) \tag{5.21}$$

These two equations, Equations 5.20 and 5.21, indicate that the base state $\underline{\boldsymbol{x}}_{\square}$, presented in Equation 4.16, should be extended with the acceleration $\underline{\boldsymbol{\omega}}_P$, angular rate $\underline{\boldsymbol{\omega}}_P$, angular acceleration $\dot{\underline{\boldsymbol{\omega}}}_P$, acceleration lag $\underline{\boldsymbol{\tau}}_{\ddot{\boldsymbol{c}}}$, and angular acceleration lag $\underline{\boldsymbol{\tau}}_{\dot{\boldsymbol{\omega}}}$, while removing the IMU related variables, i.e.,

$$\underline{\boldsymbol{x}}_P = \begin{bmatrix} \underline{\boldsymbol{c}}_P \\ \dot{\underline{\boldsymbol{c}}}_P \\ \ddot{\underline{\boldsymbol{c}}}_P \\ \underline{\boldsymbol{e}}_P \\ \underline{\boldsymbol{\omega}}_P \\ \dot{\underline{\boldsymbol{\omega}}}_P \\ \underline{\boldsymbol{\tau}}_{\ddot{\boldsymbol{c}}} \\ \underline{\boldsymbol{\tau}}_{\dot{\boldsymbol{\omega}}} \end{bmatrix} \tag{5.22}$$

Also, the input $\underline{\boldsymbol{u}}_P$ is defined as the setpoints $\underline{\boldsymbol{s}}$, both the desired acceleration $\underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}}$ and desired angular acceleration $\underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}}$ to replace the IMU related parts, as was the case in Equation 4.7, i.e.,

$$\underline{\boldsymbol{u}}_P = \begin{bmatrix} \underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}} \\ \underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}} \end{bmatrix} \tag{5.23}$$

No changes are made to the reference frames in which the position $\underline{\boldsymbol{c}}_P$, velocity $\dot{\underline{\boldsymbol{c}}}_P$, and attitude $\underline{\boldsymbol{e}}_P$ were defined for the base state $\underline{\boldsymbol{x}}_{\square}$ in Section 4.3.2 in Equations 4.17, 4.19, and 4.20 respectively. Also, the angular rate $\underline{\boldsymbol{\omega}}_P$ is defined in the same manner as in Equation 5.4.

The same trend continuous for the derivatives of the previously named variables. The change in position $\dot{\underline{\boldsymbol{c}}}_P$, velocity $\ddot{\underline{\boldsymbol{c}}}_P$, attitude $\dot{\underline{\boldsymbol{e}}}_P$, and angular rate $\dot{\underline{\boldsymbol{\omega}}}_P$ were defined in Equations 4.26, 4.28, 5.5, and 5.7 respectively.

Following, the acceleration lag $\underline{\boldsymbol{\tau}}_{\ddot{\boldsymbol{c}}}$ and acceleration lag $\underline{\boldsymbol{\tau}}_{\dot{\boldsymbol{\omega}}}$, can be expressed in the inertial $\mathbb{E}_I$ and upper moving frame $\mathbb{E}_P$ respectively. Assuming also that the acceleration setpoint $\underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}}$ and angular acceleration setpoint $\underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}}$ are defined in the same frame of reference respectively results in:

$$\begin{bmatrix} \underline{\boldsymbol{\tau}}_{\ddot{\boldsymbol{c}}} \\ \underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}} \end{bmatrix} = \begin{pmatrix} \tau_{\ddot{x}} & \tau_{\ddot{y}} & \tau_{\ddot{z}} \\ s_{\ddot{x}} & s_{\ddot{y}} & s_{\ddot{z}} \end{pmatrix} \{\mathbb{E}_I\} \tag{5.24}$$

$$\begin{bmatrix} \underline{\boldsymbol{\tau}}_{\dot{\boldsymbol{\omega}}} \\ \underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}} \end{bmatrix} = \begin{pmatrix} \tau_{\dot{p}} & \tau_{\dot{q}} & \tau_{\dot{r}} \\ s_{\dot{p}} & s_{\dot{q}} & s_{\dot{r}} \end{pmatrix} \{\mathbb{E}_P\} \tag{5.25}$$

Given these last two equations, Equations 5.24 and 5.25, Equations 5.20 and 5.21 can be fully expressed in the inertial $\mathbb{E}_I$ and upper moving frame $\mathbb{E}_P$ and no transformation matrices are necessary, i.e.,

$$\begin{Bmatrix} \dddot{x}_P \\ \dddot{y}_P \\ \dddot{z}_P \\ \ddot{p}_P \\ \ddot{q}_P \\ \ddot{r}_P \end{Bmatrix} = \begin{Bmatrix} -\ddot{x}_P/\tau_{\ddot{x}} + s_{\ddot{x}}/\tau_{\ddot{x}} \\ -\ddot{y}_P/\tau_{\ddot{y}} + s_{\ddot{y}}/\tau_{\ddot{y}} \\ -\ddot{z}_P/\tau_{\ddot{z}} + s_{\ddot{z}}/\tau_{\ddot{z}} \\ -\dot{p}_P/\tau_{\dot{p}} + s_{\dot{p}}/\tau_{\dot{p}} \\ -\dot{q}_P/\tau_{\dot{q}} + s_{\dot{q}}/\tau_{\dot{q}} \\ -\dot{r}_P/\tau_{\dot{r}} + s_{\dot{r}}/\tau_{\dot{r}} \end{Bmatrix} \tag{5.26}$$

Next, the process noise vector $\underline{\boldsymbol{\nu}}_P$ is created. In order for the UKF to estimate the lag $\underline{\boldsymbol{\tau}}$ influencing the setpoints in Equation 5.26, a variance needs to be created on the lag $\underline{\boldsymbol{\tau}}$. This is achieved by defining a change of the lag $\dot{\underline{\boldsymbol{\tau}}}$ of the system over time as the noise parameter $\underline{\boldsymbol{\nu}}_{\boldsymbol{\tau}}$. This noise parameter $\underline{\boldsymbol{\nu}}_{\boldsymbol{\tau}}$ fulfills a similar function as the noise variable $\underline{\boldsymbol{\mu}}$ to the $\underline{\boldsymbol{\lambda}}$ of an IMU in Equation 4.34. Whereby the variance on the lag $\underline{\boldsymbol{\tau}}$ allows the UKF to change the lag $\underline{\boldsymbol{\tau}}$ based on the measurements of the rest of the system.

$$\underline{\boldsymbol{\nu}}_P = \begin{bmatrix} \underline{\boldsymbol{\nu}}_{\ddot{c}} \\ \underline{\boldsymbol{\nu}}_{\dot{\omega}} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \nu_{\dot{\tau}_{\ddot{x}}} & \nu_{\dot{\tau}_{\ddot{y}}} & \nu_{\dot{\tau}_{\ddot{z}}} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \nu_{\dot{\tau}_{\dot{p}}} & \nu_{\dot{\tau}_{\dot{q}}} & \nu_{\dot{\tau}_{\dot{r}}} \end{pmatrix} \{\mathbb{E}_P\} \end{bmatrix} \tag{5.27}$$

In principle the noise vector $\underline{\boldsymbol{\nu}}_P$ is complete. However, when this process model is applied to the UKF the behavior of would not be optimal. As the only introduction of uncertainty to the state would be present on the lag $\underline{\boldsymbol{\tau}}$, which has to be small, i.e. $\times 10^{-3}$, relative to the lag itself as to not let the UKF make large changes to the expected lag values. This would results in an almost zero state covariance $\boldsymbol{P}_{\hat{x}}$ of the UKF in a couple of iterations when in the presents of an accurate observation sensors, such as the actuator length measurement devices present on the SIMONA Research Simulator (SRS). Moreover, to account for the integration errors, discussed in Section 7.2.4, extra variance on the states needs to be created. The variance is introduced by two additional Gaussian noise sources, an acceleration variance $\underline{\boldsymbol{\nu}}_{\ddot{\boldsymbol{c}}}$ and an angular acceleration $\underline{\boldsymbol{\nu}}_{\dot{\boldsymbol{\omega}}}$, akin to the Stewart platform in Configuration 1 in Section 5.2.1. The acceleration $\underline{\boldsymbol{\nu}}_{\ddot{\boldsymbol{c}}}$ and angular acceleration $\underline{\boldsymbol{\nu}}_{\dot{\boldsymbol{\omega}}}$ noises are added to the noise vector $\underline{\boldsymbol{\nu}}_P$, i.e.,

$$\underline{\boldsymbol{\nu}}_P = \begin{bmatrix} \underline{\boldsymbol{\nu}}_{\ddot{c}} \\ \underline{\boldsymbol{\nu}}_{\dot{\omega}} \\ \underline{\boldsymbol{\nu}}_{\ddot{\boldsymbol{c}}} \\ \underline{\boldsymbol{\nu}}_{\dot{\boldsymbol{\omega}}} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \nu_{\dot{\tau}_{\ddot{x}}} & \nu_{\dot{\tau}_{\ddot{y}}} & \nu_{\dot{\tau}_{\ddot{z}}} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \nu_{\dot{\tau}_{\dot{p}}} & \nu_{\dot{\tau}_{\dot{q}}} & \nu_{\dot{\tau}_{\dot{r}}} \end{pmatrix} \{\mathbb{E}_P\} \\ \begin{pmatrix} \nu_{\ddot{x}_P} & \nu_{\ddot{y}_P} & \nu_{\ddot{z}_P} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \nu_{\dot{p}_P} & \nu_{\dot{q}_P} & \nu_{\dot{r}_P} \end{pmatrix} \{\mathbb{E}_P\} \end{bmatrix} \tag{5.28}$$

The continuous-time process model f_P of the Stewart platform can now be constructed. The combination of Equations 4.35, 4.28, 5.26, 5.5, 5.7, and 5.28 accumulates in:

$$\begin{gathered} \dot{\underline{\boldsymbol{x}}}_P(t) = \mathrm{f}_P\left(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{\nu}}_P(t)\right) \\ \begin{Bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \\ \ddot{x}_P \\ \ddot{y}_P \\ \ddot{z}_P \\ \dddot{x}_P \\ \dddot{y}_P \\ \dddot{z}_P \\ \dot{e}_{0_P} \\ \dot{e}_{x_P} \\ \dot{e}_{y_P} \\ \dot{e}_{z_P} \\ \dot{p}_P \\ \dot{q}_P \\ \dot{r}_P \\ \ddot{p}_P \\ \ddot{q}_P \\ \ddot{r}_P \\ \dot{\tau}_{\ddot{x}} \\ \dot{\tau}_{\ddot{y}} \\ \dot{\tau}_{\ddot{z}} \\ \dot{\tau}_{\dot{p}} \\ \dot{\tau}_{\dot{q}} \\ \dot{\tau}_{\dot{r}} \end{Bmatrix} = \begin{Bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \\ \ddot{x}_P + \nu_{\ddot{x}_P} \\ \ddot{y}_P + \nu_{\ddot{y}_P} \\ \ddot{z}_P + \nu_{\ddot{z}_P} \\ -\ddot{x}_P/\tau_{\ddot{x}} + s_{\ddot{x}}/\tau_{\ddot{x}} \\ -\ddot{y}_P/\tau_{\ddot{y}} + s_{\ddot{y}}/\tau_{\ddot{y}} \\ -\ddot{z}_P/\tau_{\ddot{z}} + s_{\ddot{z}}/\tau_{\ddot{z}} \\ \frac{1}{2}\begin{bmatrix} -e_{x_P} & -e_{y_P} & -e_{z_P} \\ e_{0_P} & -e_{z_P} & e_{y_P} \\ e_{z_P} & e_{0_P} & -e_{x_P} \\ -e_{y_P} & e_{x_P} & e_{0_P} \end{bmatrix} \begin{Bmatrix} p_P \\ q_P \\ r_P \end{Bmatrix} \\ \dot{p}_P + \nu_{\dot{p}_P} \\ \dot{q}_P + \nu_{\dot{q}_P} \\ \dot{r}_P + \nu_{\dot{r}_P} \\ -\dot{p}_P/\tau_{\dot{p}} + s_{\dot{p}}/\tau_{\dot{p}} \\ -\dot{q}_P/\tau_{\dot{q}} + s_{\dot{q}}/\tau_{\dot{q}} \\ -\dot{r}_P/\tau_{\dot{r}} + s_{\dot{r}}/\tau_{\dot{r}} \\ \nu_{\dot{\tau}_{\ddot{x}}} \\ \nu_{\dot{\tau}_{\ddot{y}}} \\ \nu_{\dot{\tau}_{\ddot{z}}} \\ \nu_{\dot{\tau}_{\dot{p}}} \\ \nu_{\dot{\tau}_{\dot{q}}} \\ \nu_{\dot{\tau}_{\dot{r}}} \end{Bmatrix} \end{gathered} \tag{5.29}$$

6

Sensor Selection and Parametrization

In the previous chapter, Chapter 5, joined process and observation models were created, linking input to state and state to observation of an enclosed Virtual Reality (VR)/Augmented Reality (AR) system on a moving Stewart platform. These models contain the $\underline{\boldsymbol{\nu}}$ and observation noise $\underline{\boldsymbol{o}}$ vectors representing the zero-mean Gaussian, i.e. $\aleph\left(0, ...\right)$, influence on their respective models in the form of their variances as $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices through Equation 3.16 in the Unscented Kalman Filter (UKF).

In this chapter, the variances of the various noise sources are gathered and estimated with measurement data where possible. Also, the time-lags $\underline{\boldsymbol{\tau}}$, used for the integration of the setpoints for the Stewart platform pose as per Section 5.4.1, are estimated in this chapter. The values found in this chapter are the starting point for the analysis in the next chapter, Chapter 7.

First, Section 6.1 describes the selected VR system in short. Followed by noise profiles for the Inertial Measurement Unit (IMU) of the Head-Mounted Display (HMD) as well as the visual tracking sensor in Sections 6.1.1 and 6.1.2 respectively. Next, the SIMONA Research Simulator (SRS), the Stewart platform to be modelled, will be presented in short in Section 6.2. The focus of the following sections, Sections 6.2.1 and 6.2.2, will be on its sensors, namely the absolute linear encoders measuring the hydraulic actuator lengths and the IMU attached to the upper moving platform of the SRS. The time-lag $\underline{\boldsymbol{\tau}}$ on the highest order software-based setpoints $\underline{\boldsymbol{s}}$ to pose, consisting of position and attitude, is measured and discussed in Section 6.2.3.

6.1. Virtual Reality system

The VR system to be simulated and analyzed is presented in this section. As stated in Section 2.2, a six-Degree-of-Freedom (DoF) VR system comprises out of a HMD, with an IMU attached, and absolute tracking, mostly in the form of visual tracking. In Section 6.1.1, the Pimax 8k-x [52], shown in Figure 6.1, is proposed and initial noise parameters of its IMU are presented. Followed by Section 6.1.2, where the TrackIR 5 [69], depicted in Figure 6.2, is proposed mainly due to its capability of working in a moving environment. Moreover, due to a lack of in-depth information on the TrackIR 5 noise profile, one is constructed in this section based on stationary measurements.

Figure 6.1: Pimax 8k-x [52]

Figure 6.2: TrackIR 5 [69]

6.1.1. Head-Mounted Display

The Pimax 8k-x, shown in Figure 6.1, was chosen based on its availability and design coherency with other SteamVR devices. The compatibility with SteamVR devices makes it possible to read out the raw IMU data by making use of the open-source software tools like libsurvive [35].

In the rest of this section, the velocity/angle random walk and the acceleration/rate random walk of the IMU onboard the Pimax 8k-x are estimated based on stationary measurements and compared to the manufacturer data sheet. The variance on these two IMU noise sources, the velocity/angle random walk and the acceleration/rate random walk, are required as they are part of the kinematics. Thanks to Equations 3.16 and 3.19 the influence of the IMU noise is propagated through the process model in Equation 4.35 to the next predicted state $\underline{\hat{\boldsymbol{x}}}_k^-$ and its covariance $\boldsymbol{P}_{\hat{x}_k^-}$ following Equations 3.20 and 3.21.

The Pimax 8k-x's IMU is positioned between the displays and at the same height as retinas of the user. This IMU is a MPU-6500 as depicted in the breakdown in [53]. The IMU, a Micro-Electro-Mechanical Systems (MEMS), includes an accelerometer and a gyroscope and runs up to 1000 Hz. The data-sheet of the MPU-6500 [44] states that the accelerometer has a velocity random walk $N_{\boldsymbol{f}_H}$ of 3×10^2 µg/$\sqrt{\text{Hz}}$, or $\sim 3 \times 10^{-3}$ m/s^2/$\sqrt{\text{Hz}}$, on all three axis. Meanwhile, the gyroscope has an angle random walk coefficient $N_{\boldsymbol{\omega}_H}$ on all three axis of 1×10^{-2} °/s/$\sqrt{\text{Hz}}$, or $\sim 2 \times 10^{-4}$ rad/s/$\sqrt{\text{Hz}}$. Transforming both numbers to discrete standard deviations found in actual measurements both the angle and velocity random walk are to be multiplied by $\sqrt{1000}\,\sqrt{\text{Hz}}$ in this case. This results in a standard deviation for the accelerometer of 9×10^{-2} m/s^2 and 6×10^{-3} rad/s for the gyroscope. Both accelerometer $K_{\boldsymbol{f}_H}$ and rate random walk coefficients $K_{\boldsymbol{\omega}_H}$, Power Spectral Density (PSD) values, representing the rate at which the biases change over time, are not present on the data-sheet of MPU-6500 [44].

By applying a Overlapping Allan Deviation (OAD), a standard procedure for creating noise profiles for IMU [15, 60, 73, 27], on stationary measurements of the MPU-6500 with a duration of 20 min, the velocity $N_{\boldsymbol{f}_H}$ and angle random walk $N_{\boldsymbol{\omega}_H}$ presented on the data-sheet can be verified. The OAD, explained in Appendix A, is a time-domain-analysis technique capable of determining both the deviation as well as the source of the noise. The OAD is a superior adaptation of the Allan Deviation (AD) by exploring the available data in different sized 'overlapping' bins [32].

In Figure 6.3, the AD of the stationary measurements of the accelerometer of all three axis are plotted on a log-log scale. The white noise coefficient $N_{\boldsymbol{f}_H}$ can be found at the green dot, found graphically at the intersection of a fitting sloped line, with a slope of $-1/2$, and an averaging time $m\tau_0$ of one. Here, the white noise coefficients $N_{\boldsymbol{f}_H}$ found were between a third and a half of the values found on the data-sheet. These values discrete counterpart, extracted by multiplying by $\sqrt{1000}\,\sqrt{\text{Hz}}$, and representing the standard deviation on the measurements are given in Table 6.1.

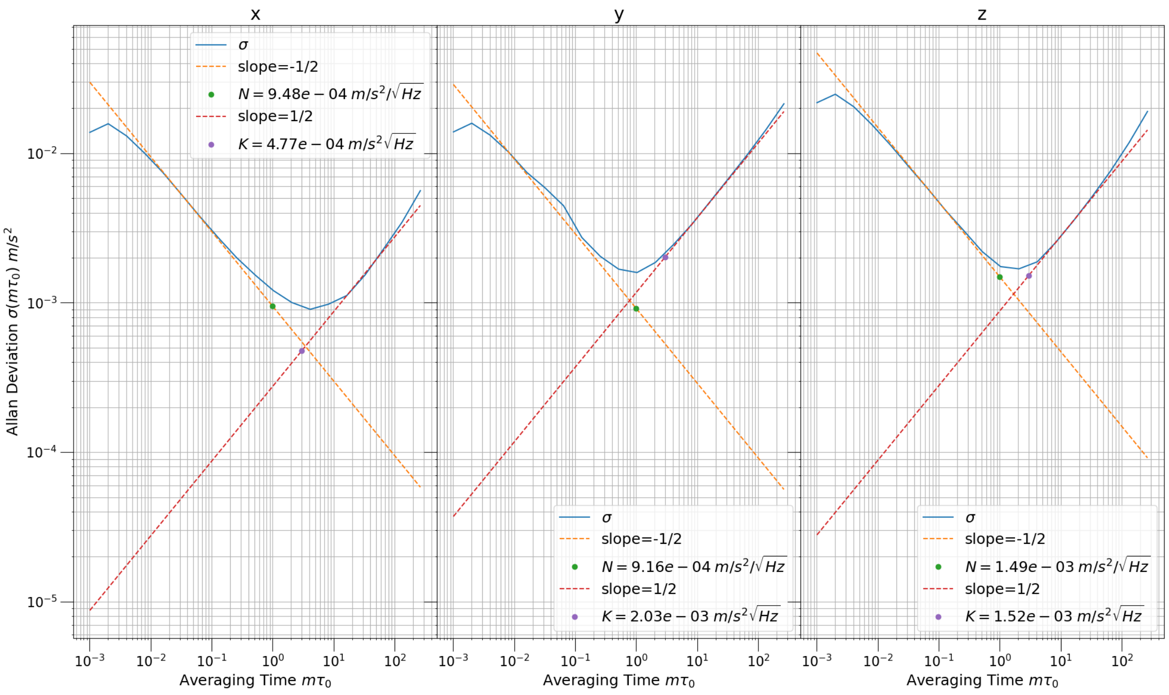


Figure 6.3: Overlapping Allan Deviation method applied to stationary measurement of the MPU-6500 accelerometer found in the Pimax 8k-x.

Moreover, the accelerometer random walk coefficients $K_{\boldsymbol{f}_H}$ are also found using this method. This coefficient $K_{\boldsymbol{f}_H}$ indicated as the purple dot in Figure 6.3, is found graphically at the intersection of a straight line, fitted to the AD with a slope of $1/2$, and at an averaging time $m\tau_0$ of three. The standard deviation on the change of the bias of the IMU is retrieved by multiplying the acceleration random walk coefficients $K_{\boldsymbol{f}_H}$ by the square-root of the IMU operational frequency, i.e. $K_{\boldsymbol{f}_H} \cdot \sqrt{1000}$, in this case. The final values are also presented in Table 6.1.

In Figure 6.4, the AD of the stationary measurements of the gyroscope of all three axis are plotted on a log-log scale. The white noise coefficient $N_{\boldsymbol{\omega}_H}$ is found in the same manner as before, by graphically finding the intersection of a line, with a slope of $-1/2$ and fitted to the AD, and an averaging time $m\tau_0$ of one. Here, the white noise coefficients $N_{\boldsymbol{f}_H}$ found are around half of the values presented in the datasheet of the IMU. However, the rate random walk coefficients $K_{\boldsymbol{\omega}_H}$ are not found as the measurement duration was not long enough. In Figure 6.4, the averaging time is not long enough for the OAD to show a clear upward trent with a slope of a half, indicating the measurement duration was too short.

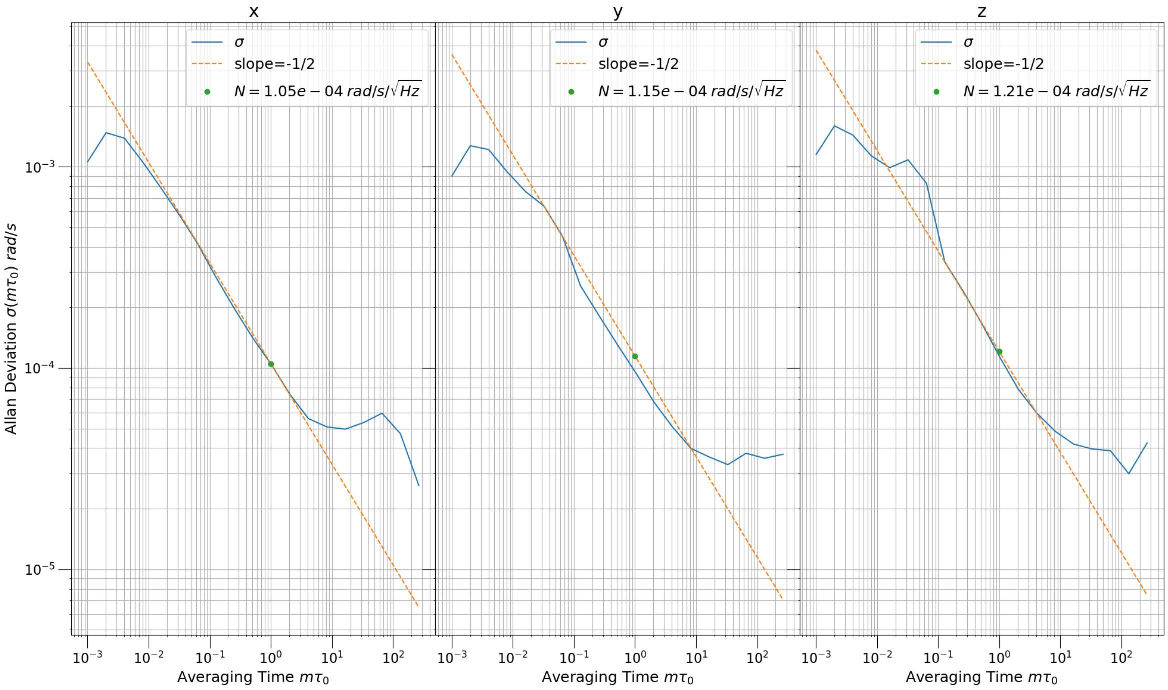


Figure 6.4: Overlapping Allan Deviation method applied to stationary measurement of MPU-6500 gyroscope found in Pimax 8k-x.

In [47], the gyroscope of a MPU-6500 IMU was found to have an angle random walk of 1.42×10^{-4}, 1.08×10^{-4}, and 1.12×10^{-4} rad/s/$\sqrt{\text{Hz}}$ around the x, y, and z-axis respectively. Close to the values found in Figure 6.4. However, in this paper the measurement was long enough to get the gyroscope's rate random walk coefficient μ_{ω_H} of 5.66×10^{-6}, 1.09×10^{-5}, and 1.42×10^{-5} rad/s^2/$\sqrt{\text{Hz}}$ were measured around the x, y, and z-axis respectively in the aforementioned paper. Until a longer measurement is performed, the values mention in this paper for rate random walk are used in the simulation in Chapter 7.

Table 6.1: Sensor noise profile overview on the Pimax 8k-x HMD, the MPU-6500

Frequency	Sensor	Velocity/Angle Random Walk	Acceleration/Rate Random Walk
1000 Hz	Accelerometer	$\sigma_{f_{\ddot{x}_H}}$: 3.0×10^{-2} m/s^2	$\sigma_{f_{\mu_{\ddot{x}_H}}}$: 1.5×10^{-2} m/s^3
		$\sigma_{f_{\ddot{y}_H}}$: 2.9×10^{-2} m/s^2	$\sigma_{f_{\mu_{\ddot{y}_H}}}$: 6.4×10^{-2} m/s^3
		$\sigma_{f_{\ddot{z}_H}}$: 4.7×10^{-2} m/s^2	$\sigma_{f_{\mu_{\ddot{z}_H}}}$: 4.8×10^{-2} m/s^3
	Gyroscope	$\sigma_{\omega_{p_H}}$: 3.3×10^{-3} rad/s	$\sigma_{\omega_{\mu_{p_H}}}$: 1.8×10^{-4} rad/s^2
		$\sigma_{\omega_{p_H}}$: 3.6×10^{-3} rad/s	$\sigma_{\omega_{\mu_{q_H}}}$: 3.4×10^{-4} rad/s^2
		$\sigma_{\omega_{p_H}}$: 3.8×10^{-3} rad/s	$\sigma_{\omega_{\mu_{r_H}}}$: 4.5×10^{-4} rad/s^2

The impact of the accuracy of the IMU of the HMD on the proper pose will be investigated in the initial analysis in Section 7.4. Also, the impact of assuming the wrong value, with a factor of ten, will be investigated in Section 7.4. Moreover, both the IMU noise values will be further validated in the experiment proposed in Chapter 8.

6.1.2. Visual Tracking

The visual tracking nominally employed by SteamVR devices, both Base Station version 1.0 and 2.0, are not designed to be used in a moving environment. The spinning Infrared (IR) emitter, inside these Base Stations, are sensitive to sudden movement and designed to stop working if any motion is detected by their build-in accelerometer. More on the working of the Base Stations is found in Section 2.2.2.1. Therefore, the TrackIR 5 visual tracking system was chosen, the sensor itself depicted in Figure 6.2. This device is capable of withstanding the movement of the SRS, was available, and its small size and single sensor makes it easy to integrate into the cramped interior of the SRS. For details on the SRS itself see Section 2.3.1.

The TrackIR 5 is a IR transceiver, emitting and receiving the IR light in the same sensor. The IR light is refracted by a passive TrackClip, presented in Figure 6.5. The light is refracted on the three reflectors seen in the figure below and attached to the user's head. It is assumed that the known shape lets the TrackIR 5 proprietary algorithm workout the position and attitude of the TrackClip at an update frequency of 120 Hz.

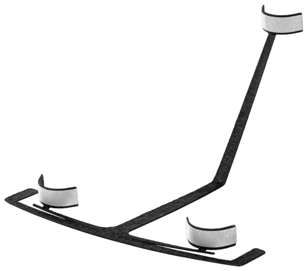


Figure 6.5: TrackIR 5

No accuracy information was found on the TrackIR 5 sensor and proprietary software combination. However, in order to model the TrackIR 5 as a sensor in a Kalman-Filter framework a Gaussian noise profiles is necessary. For this reason, stationary measurements were made with the IR light emitter clamped in place facing TrackIR 5 sensor 50 cm away for approximately five minutes. The result of these measurements are shown in Figure 6.6 where some drift on all measurements are present. This is likely a result of the proprietary software running the TrackIR 5, but more experiments are needed to understand the accuracy of this sensor properly, see Chapter 8.

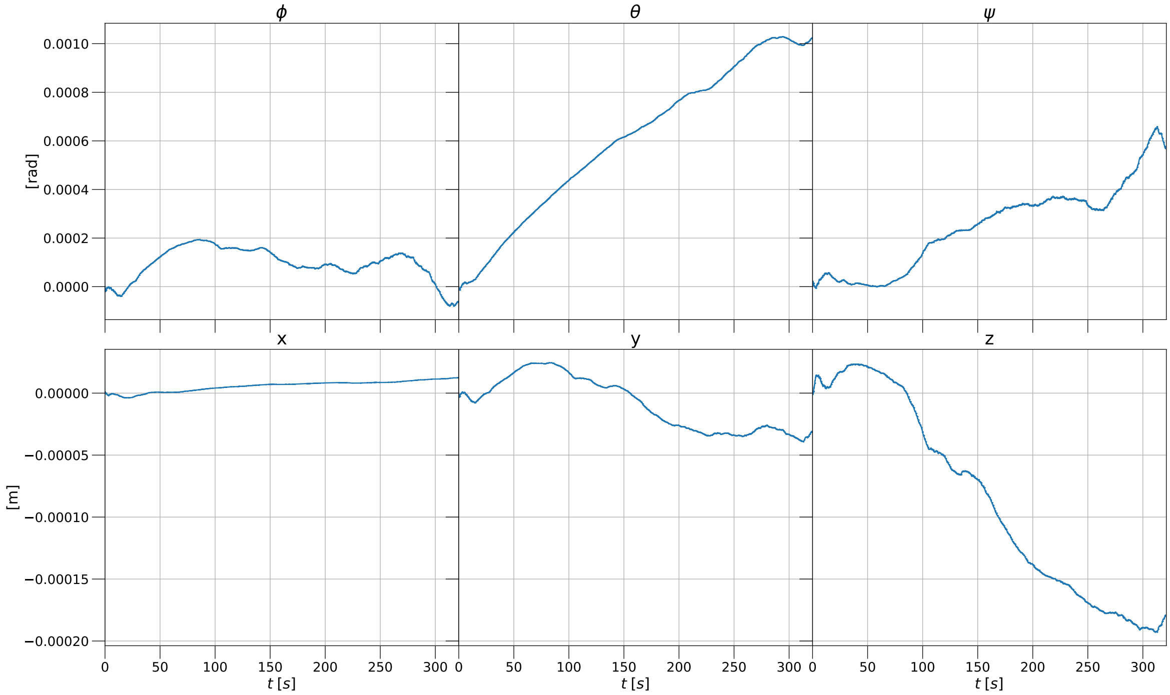


Figure 6.6: Stationary measurements with TrackIR 5.

In order to model noise parameters on the stationary measurement the slow drift is removed from the data. A high-pass filter was applied in the form of a Butterworth filter [8]. After analyzing the stationary measurement in the frequency domain using the Fast Fourier Transform, the cut-off frequency was set to 0.1 Hz in the Butterworth filter. The resulting data is presented in Figure 6.7

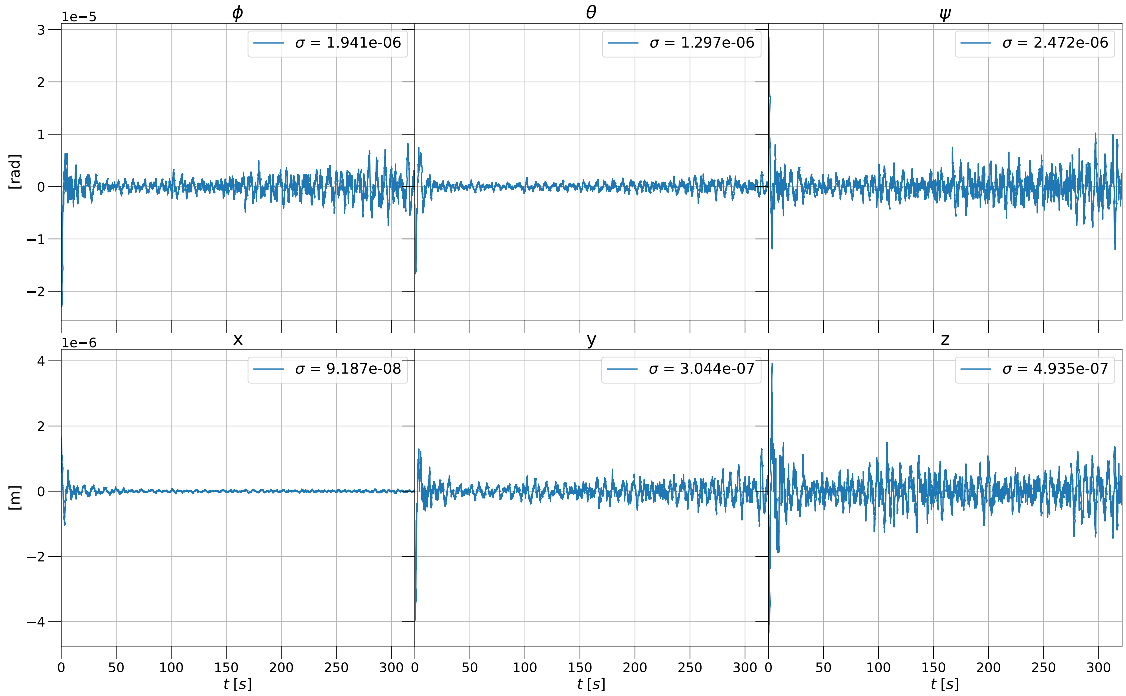


Figure 6.7: Stationary measurements with TrackIR 5 high-pass filtered with cut-off frequency 0.1 Hz.

The resulting filtered standard deviation compiled in Table 6.2 are small. This makes sense as the the original measurements performed stationary and filtered to remove the drift present. Meaning that the standard deviation values should be taken with caution. The proprietary algorithm of the TrackIR 5 could be optimized for stationary measurements, as this would be a recurring scenario in its use case. However, it is unlikely that in a real use case the error would be this small. For this reason it is proposed to increase the computed standard deviation in Table 6.2 by a factor of 1000, still resulting in an accurate sensor model when compared to the sensor results of purpose build HTC Vive for example [7]. Moreover, the factor 1000 is probably on the small side as it is suspected that the dynamic accuracy is less than its static accuracy.

Table 6.2: TrackIR 5 sensor overview and the estimated standard deviation on filter stationary measurement of the TrackIR 5.

Device	Sensor	Frequency	Axis	Deviation	Deviation $\times 1000$
TrackIR 5	Infrared	100 Hz	σ_{x_H}:	9.19×10^{-8} m	9.19×10^{-5} m
			σ_{y_H}:	3.04×10^{-7} m	3.04×10^{-4} m
			σ_{z_H}:	4.94×10^{-7} m	4.94×10^{-4} m
			σ_{ψ_H}:	1.94×10^{-6} rad	1.94×10^{-3} rad
			σ_{θ_H}:	1.30×10^{-6} rad	1.30×10^{-3} rad
			σ_{ϕ_H}:	2.47×10^{-6} rad	2.47×10^{-3} rad

The consequences of using the wrong accuracy data is looked into in Chapter 7. There, the problem is faced from both sides, the accuracy being smaller/bigger and/or the expected accuracy being smaller/bigger than proposed on the right side of Table 6.2.

6.2. SIMONA Research Simulator

The SIMONA Research Simulator (SRS) is a high fidelity flight simulator on a moving platform. The moving platform is implemented as a Stewart platform giving the SRS six-DoF [64]. As mentioned in Section 2.3.1, the position of the length of each of the six hydraulic actuators are measured via six Heidenhain LC 415 linear encoders [19], depicted in Figure 6.8, one per actuator. The modelling of the noise present on a HeidenHain LC 415 absolute linear encoder will be detailed in Section 6.2.1.

Figure 6.8: Heidenhain LC 415, an absolute linear encoder used for measuring hydraulic actuator length on the SRS.

Also, the IMU attached to upper moving frame of the SRS is presented in Section 6.2.2. Both measurements are published onto Delft University Environment for Communication and Activation (DUECA).

DUECA, a middleware, enabling real-time simulations by using both inter-module communication as well as communication between different modules on a distributed network of computers [50]. Moreover, DUECA uses a publish and subscribe design that enables modules, a self-contained possible real-time computational element, to activate when receiving or sending data. In design and use, it is similar to other middlewares like enhanced Communication Abstraction Layer (eCAL) [11], Message Queuing Telemetry Transport (MQTT) [48] and Robot Operating System (ROS) [49].

Due to the distributed nature of DUECA and the sensors themselves, the time between actual measurement of sensors and other modules receiving said measurements can be up to 50 ms. The success of resolving this delay and accompanying error by applying the Smoothed Delayed Unscented Kalman Filter (SDUKF), outlined in Section 3.3.2, will be analyzed in Section 7.5.4.

6.2.1. Absolute Linear Encoders

The absolute linear encoders used on the SRS, HeidenHain LC 415 [19], are modelled using the information present on their relevant data-sheet. Here, it is stated that the standard deviation on a measurement is 5×10^{-6} m. This noise profile is assumed true for all six absolute linear encoders and is shown in Table 6.3. The impact between the actual and expected noise on the absolute linear encoders will be investigated in Section 7.4.

Table 6.3: Absolute linear encoders sensor overview.

Device	Frequency	Deviation
HeidenHain LC 415	100 Hz	σ_{l_i}: 5×10^{-6} m

6.2.2. Inertial Measurement Unit

For the initial analysis the IMU is assumed to be of similar quality akin to the one used in [43] and are presented in Table 6.4. These values need to be verified experimentally, more on this in Chapter 8. Also, the need for an IMU attached to the SRS will be analyzed initially in Section 7.4. Here, the impact of the accuracy of an IMU, when available will also be analyzed.

Table 6.4: Sensor noise profile overview of the IMU attached to the SRS.

Frequency	Sensor	Velocity/Angle Random Walk	Acceleration/Rate Random Walk
100 Hz	Accelerometer	$\sigma_{f_{\ddot{x}_P}}$: 1.9×10^{-3} m/s^2	$\sigma_{f_{\mu_{\ddot{x}_P}}}$: 1×10^{-3} m/s^3
		$\sigma_{f_{\ddot{y}_P}}$: 2.1×10^{-3} m/s^2	$\sigma_{f_{\mu_{\ddot{y}_P}}}$: 1×10^{-3} m/s^3
		$\sigma_{f_{\ddot{z}_P}}$: 1.8×10^{-3} m/s^2	$\sigma_{f_{\mu_{\ddot{z}_P}}}$: 1×10^{-3} m/s^3
	Gyroscope	$\sigma_{\omega_{p_P}}$: 2.4×10^{-3} rad/s	$\sigma_{\omega_{\mu_{p_P}}}$: 1×10^{-4} rad/s^2
		$\sigma_{\omega_{p_P}}$: 4.1×10^{-3} rad/s	$\sigma_{\omega_{\mu_{q_P}}}$: 1×10^{-4} rad/s^2
		$\sigma_{\omega_{p_P}}$: 3.3×10^{-3} rad/s	$\sigma_{\omega_{\mu_{r_P}}}$: 1×10^{-4} rad/s^2

6.2.3. Setpoints

In this section the time lags $\underline{\tau}$, meaning the duration from sending the 'Motion Command Signals' (setpoints) to the resulting motion, is approximated per DoF based on measurements. As stated in Sections 2.3.3 and 5.4, the resulting motion of the Stewart platform's setpoints can be approximated by a first order response. However, the exact time lags are not crucial but rather a rough approximation of the time lags are required, due to the fact that the kinematics model in combination with the UKF and appropriate selected time lag variance should approximate the first order time lag online.

Following, the time lag $\underline{\tau}$ in the first order response between the position $\underline{\boldsymbol{s}}_{\boldsymbol{c}}$ and attitude setpoints $\underline{\boldsymbol{s}}_{\boldsymbol{e}}$ and actual position $\underline{c}_P$ and attitude $\underline{e}_P$ should be the same as the time lag for the acceleration and angular acceleration cases. This is due to the fact that in the Laplace domain time integration and applying the time lag can be done in any sequence. The time lag $\underline{\tau}$ per axis is estimated by applying a range of time lags $\underline{\tau}$ per axis in a first order system, i.e.,

$$\mathrm{G}(s) = \frac{\mathrm{Y}(s)}{\mathrm{U}(s)} = \frac{1}{\tau s + 1} \tag{6.1}$$

In Equation 6.1, the position $\underline{\boldsymbol{s}}_{\boldsymbol{c}}$ and attitude setpoints $\underline{\boldsymbol{s}}_{\boldsymbol{e}}$ are the input $\mathrm{U}(s)$, the predicted position $\hat{\underline{c}}_P$ and attitude $\hat{\underline{e}}_P$ are subsequently the output $\mathrm{Y}(s)$. In turn, the output are compared to the measured position $\underline{c}_{P_m}$ and attitude $\underline{e}_{P_m}$ of the SRS to compute a Mean Squared Error (MSE). Final, the time lag $\underline{\tau}$ with the lowest MSE is considered optimal for the respective axis.

Both the setpoints and the pose were measured on the SRS while relatively large maneuvers were performed as can be ascertained from the measured position and attitude in Figures 6.9 and 6.10.

The time lags $\underline{\tau}$ for the translation and rotation are presented in Figures 6.9 and 6.10 respectively. In both figures, the measured pose is compared against the most accurate first order response of the pose based setpoints, with the best approximate lag per component given in the label of each figure. Also, the approximate lag per axis and the respective MSE are compiled in Table 6.5.

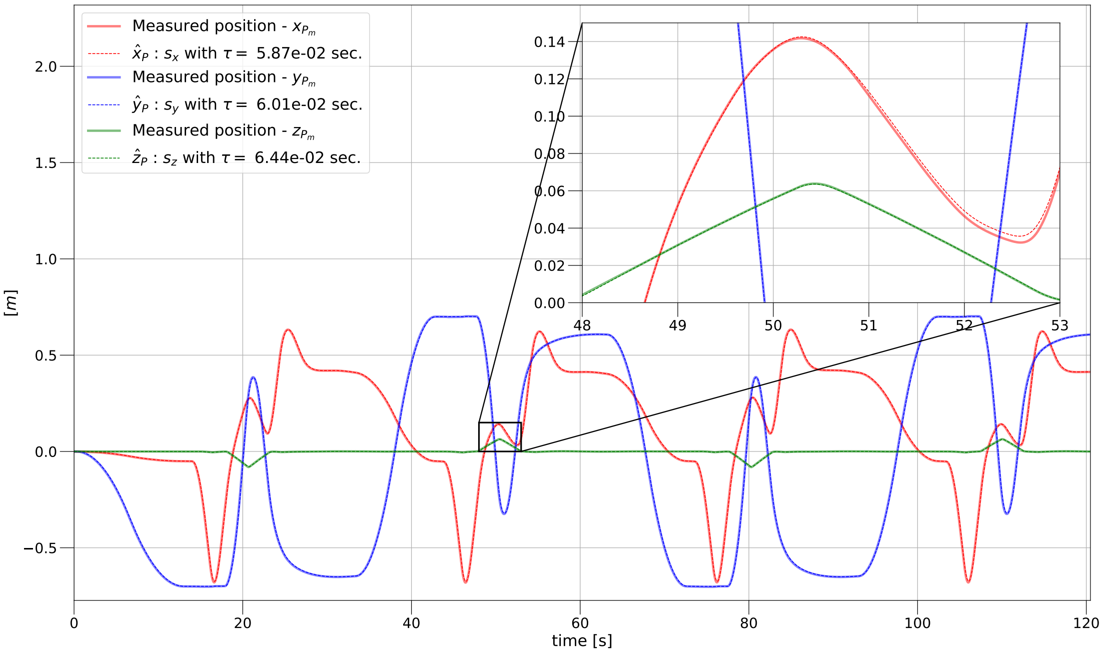


Figure 6.9: The measured position per axis plotted against the most accurate first order response of the position based setpoints.

Both the measured attitude as well as the attitude based setpoints were recorded as Euler angles (ϕ, θ, ψ). The zoomed in section for Figure 6.10 show a relative bigger difference than in Figure 6.9. However, this section accomplished to establish a reasonable time lag estimate, due to the fact that the UKF in combination with the Stewart platform kinematics in Section 5.4 allow for online time lag estimation.

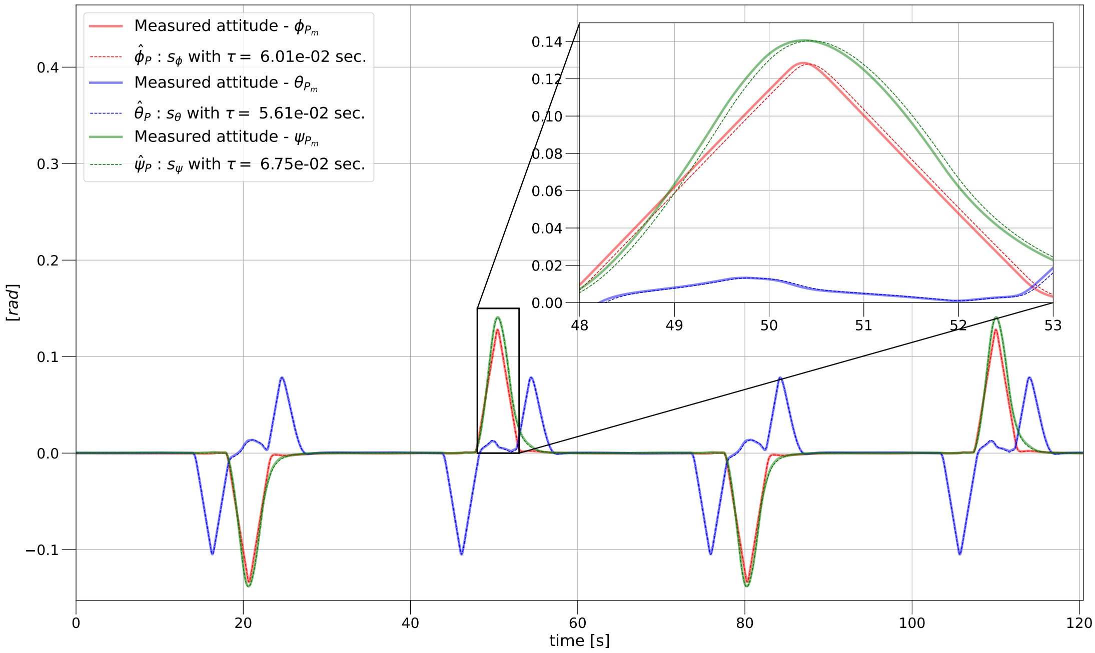


Figure 6.10: Plotted the measured attitude versus the best fitting time lag applied to the attitude setpoints.

Table 6.5: Time lag and MSE observed when comparing the highest order setpoints integrated twice to measured position and attitude of the SRS.

Setpoint	Frequency	Time Lag	MSE
Acceleration $\underline{s}_{\ddot{c}}$	100 Hz	$\tau_{\ddot{x}}$: 5.87×10^{-2} s	1.62×10^{-6} m
		$\tau_{\ddot{y}}$: 6.01×10^{-2} s	1.44×10^{-6} m
		$\tau_{\ddot{z}}$: 6.44×10^{-2} s	1.04×10^{-7} m
Angular Acceleration $\underline{s}_{\dot{\omega}}$		$\tau_{\dot{p}}$: 6.01×10^{-2} s	9.44×10^{-8} rad
		$\tau_{\dot{q}}$: 5.61×10^{-2} s	2.13×10^{-7} rad
		$\tau_{\dot{r}}$: 6.75×10^{-2} s	2.40×10^{-7} rad

7

Model Based Simulation

In this chapter, an initial analysis is performed of the combined Unscented Kalman Filter (UKF) algorithm, kinematic model and sensors within a simulation. Both the Head-Mounted Display (HMD) and the SIMONA Research Simulator (SRS) are simulated to move according to a realistic motion pattern, presented in Section 7.2. In turn, the motion is picked up by the various sensors presented in Chapter 6, depending on the configuration.

Following, these sensor readings are the input to their respective system models that are operated by the UKF, presented in Chapter 3, estimating the proper vehicle-fixed pose of the HMD. In order to integrate the discrete sensor readings into the UKF, a discrete algorithm, the kinematic models of each configuration presented in Chapter 5 are discretized in Section 7.1. Further UKF quaternion implementation is presented in Section 7.3. Here, the methods for dealing with quaternion weighted averages, quaternion covariances, and pseudo quaternion observation are explained.

All previous information presented up until the last section of this chapter was in service of the initial analysis. The initial analysis is presented in Section 7.4. Here, the performance metrics are first defined and subsequently applied to the results of the different configurations. The performance metrics, alongside the plotted data will give an overview of their total performance given the sensors present.

A sensitivity analysis is performed by modifying a predefined set of parameters in Section 7.5. Followed by an analysis on the introduction and mitigation of latency on the linear encoders in Section 7.5.4. Final, this chapter is concluded in Section 7.6.

The initial analysis will form the basis of the experiments. These will be presented in the next chapter, Chapter 8.

7.1. Discretization

In this section the discretization of the continuous integration in the process models $\mathrm{f}_\square$, presented in Chapter 5, are shown. The discretization of the continuous process models $\mathrm{f}_\square$ are necessary to implement the discrete sampled sensors. Moreover, the UKF algorithm, presented in Chapter 3, works with discretized process models, i.e.,.

$$\underline{\dot{\boldsymbol{x}}}_\square(t) = \mathrm{f}_\square\left(\underline{\boldsymbol{x}}_\square(t), \underline{\boldsymbol{u}}_{\square_k}, \underline{\boldsymbol{\nu}}_\square(t)\right) \quad \xrightarrow[\text{Method}]{\text{Discrete Integration}} \quad \underline{\boldsymbol{x}}_{\square_{k+1}} = \mathrm{F}_\square\left(\underline{\boldsymbol{x}}_{\square_k}, \underline{\boldsymbol{u}}_{\square_k}, \underline{\boldsymbol{\nu}}_{\square_k}\right) \tag{7.1}$$

The discretization of the continuous observation models does not require any method and these models are used as is in a discrete fashion, i.e.,

$$\underline{\boldsymbol{y}}_\square(t) = \mathrm{h}_\square\left(\underline{\boldsymbol{x}}_\square(t), \underline{\boldsymbol{o}}_\square(t)\right) \quad \xrightarrow{\qquad\qquad} \quad \underline{\boldsymbol{y}}_{\square_k} = \mathrm{H}_\square\left(\underline{\boldsymbol{x}}_{\square_k}, \underline{\boldsymbol{o}}_{\square_k}\right) \tag{7.2}$$

In principle, the forward Euler method, a first order numerical integration method, is often used to integrate models as a whole [41, 43], i.e.,

$$\underline{\boldsymbol{x}}_{k+1} = \underline{\dot{\boldsymbol{x}}}_k \Delta t + \underline{\boldsymbol{x}}_k \tag{7.3}$$

The forward Euler method ignores any higher order terms and assumes the change per time step is constant. These assumptions introduce errors into the prediction step. The error caused by this method can be reduced by shortening the time-step Δt. The advantage of the forward Euler method is the low computational cost and ease of implementation. Moreover, an analysis on the error introduced by the forward Euler method can be found in Section 7.2.4.

The accurate representation of the pose of a HMD to the user in a timely manner is the most important contributor to the user experience [40, 34, 26]. Due to the coupling of the system, an improvement in attitude accuracy of both systems would also improve the translational accuracy. The lateral states of both system depend on the attitude to translate the specific force reading to the inertial frame $\mathbb{E}_I$, shown in Equations 4.30. Moreover, the attitude of the motion platform is used to translate the difference in position between the HMD and Upper Gimbal Point (UGP) to the upper moving platform $\mathbb{E}_P$. Therefore, it is opted to use an analytical based discrete integration method to improve the quaternion attitude propagation in time [41]. The method is explained and discussed in Section 7.1.1.

Moreover, Configuration 3, the setpoint implementation, was found to be sensitive to initial starting values and integration errors. To mitigate the latter, a method is proposed to reduce the discretization error when integrating the setpoints $\underline{s}$ in Section 7.1.2.

7.1.1. Quaternion Integration

The continuous change in quaternion state e was described in Equations 4.33, and 5.5 by the angular rate $\underline{\omega}_{\square}$ in the respective non-inertial frame, both the HMD frame $\mathbb{E}_H$ and the upper moving frame $\mathbb{E}_P$. The analytical discretization of the quaternion propagation through time is described by Merwe and Wan [41].

The first order derivation of the quaternion attitude e was expressed in short in Equation 4.31, and repeated here again but using a reversed right-to-left quaternion multiplication $\underset{R}{\otimes}$ to ease explanation, i.e.,

$$\dot{\underline{e}}_{\square} = \frac{1}{2}\underline{\omega}_{\square}\underset{R}{\otimes}\underline{e}_{\square} \tag{7.4}$$

Abriviating the right-to-left quaternion multiplication $\underset{R}{\otimes}$ and the angular rate $\underline{\omega}_{\square}$ as $\boldsymbol{\Omega}_{\square}$ for later use:

$$\dot{\underline{e}}_{\square} = \frac{1}{2}\boldsymbol{\Omega}_{\square}\underline{e}_{\square} \tag{7.5}$$

And in turn can be expanded to:

$$\begin{Bmatrix} \dot{e}_{0_\square} \\ \dot{e}_{x_\square} \\ \dot{e}_{y_\square} \\ \dot{e}_{z_\square} \end{Bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -p_{\square} & -q_{\square} & -r_{\square} \\ p_{\square} & 0 & r_{\square} & -q_{\square} \\ q_{\square} & -r_{\square} & 0 & p_{\square} \\ r_{\square} & q_{\square} & -p_{\square} & 0 \end{bmatrix}\begin{Bmatrix} e_{0_\square} \\ e_{x_\square} \\ e_{y_\square} \\ e_{z_\square} \end{Bmatrix} \tag{7.6}$$

The resulting quaternion attitude derivation is no different from Equation 4.33, but the sequence is altered to aid further explanation in this section.

The discrete propagated quaternion $\underline{e}_{\square_{k+1}}$ over one time step Δt with an assumed constant angular rate $\underline{\omega}_{\square}$, expressed in the non-inertial frame $\mathbb{E}_{\square}$, is derived as [41, 51]:

$$\underline{e}_{\square_{k+1}} = \text{exp}\left(\frac{1}{2}\boldsymbol{\Omega}_{\square_k}\Delta t\right)\underline{e}_{\square_k} \tag{7.7}$$

Where $\boldsymbol{\Omega}_{\square_k}$ is a 4×4 skew-symmetrical matrix shown in Equation 7.5, similar to the 3×3 version in Equation 5.6, composed of the angular rate $\underline{\omega}_{\square_k}$ at time-step k. The effective rotation in attitude caused by the angular rate $\underline{\omega}_{\square_k}$, assumed constant, over one time step Δt in the non-inertial moving frame $\mathbb{E}_{\square}$, is defined as:

$$\underline{\omega}_{\square_k}\Delta t = \begin{pmatrix} p_{\square_k}\Delta t & q_{\square_k}\Delta t & r_{\square_k}\Delta t \end{pmatrix}\{\mathbb{E}_{\square}\} \tag{7.8}$$

The effective rotation $\underline{\boldsymbol{\omega}}_{\square_k}$, presented in Equation 7.8, is given in the angle-axis form. In this form, the total rotation $\|\underline{\boldsymbol{\omega}}_{\square_k}\Delta t\|$ is given around a unit-axis $\frac{\underline{\boldsymbol{\omega}}_{\square_k}\Delta t}{\|\underline{\boldsymbol{\omega}}_{\square_k}\Delta t\|}$, as is the case with a Micro-Electro-Mechanical Systems (MEMS) based gyroscopes.

Next, the 4×4 skew-symmetrical matrix $\boldsymbol{\Omega}_{\square_k}\Delta t$ used in Equation 7.7 is composed similar to Equation 7.5, whereby the elements of the effective rotation $\underline{\boldsymbol{\omega}}_{\square_k}\Delta t$, found in Equation 7.8, are used, i.e.,

$$\boldsymbol{\Omega}_{\square_k}\Delta t = \begin{bmatrix} 0 & -p_{\square_k}\Delta t & -q_{\square_k}\Delta t & -r_{\square_k}\Delta t \\ p_{\square_k}\Delta t & 0 & r_{\square_k}\Delta t & -q_{\square_k}\Delta t \\ q_{\square_k}\Delta t & -r_{\square_k}\Delta t & 0 & p_{\square_k}\Delta t \\ r_{\square_k}\Delta t & q_{\square_k}\Delta t & -p_{\square_k}\Delta t & 0 \end{bmatrix} \tag{7.9}$$

The exponential factor in Equation 7.7 can be expanded and truncated using the skew-symmetrical matrix $\boldsymbol{\Omega}_{\square_k}\Delta t$ in Equation 7.9 to the closed-form notation [41, 51] as:

$$\exp\left(\frac{1}{2}\boldsymbol{\Omega}_{\square_k}\Delta t\right) = \boldsymbol{I}\cos\left(s_{\square_k}\right) + \frac{1}{2}\boldsymbol{\Omega}_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{s_{\square_k}} \tag{7.10}$$

Where $s_{\square_k}$ is defined as half the norm of the effective rotation $\underline{\boldsymbol{\omega}}_{\square_k}\Delta t$, i.e.,

$$\begin{aligned} s_{\square_k} &= \frac{1}{2}\left\|\underline{\boldsymbol{\omega}}_{\square_k}\Delta t\right\| = \frac{1}{2}\left\|p_{\square_k}\Delta t \quad q_{\square_k}\Delta t \quad r_{\square_k}\Delta t\right\| \\ &= \frac{1}{2}\sqrt{\left(p_{\square_k}\Delta t\right)^2 + \left(q_{\square_k}\Delta t\right)^2 + \left(r_{\square_k}\Delta t\right)^2} \end{aligned} \tag{7.11}$$

Using the closed-form notation presented in Equation 7.10 in Equation 7.7 results in:

$$\underline{\boldsymbol{e}}_{\square_{k+1}} = \left[\boldsymbol{I}\cos\left(s_{\square_k}\right) + \frac{1}{2}\boldsymbol{\Omega}_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{s_{\square_k}}\right]\underline{\boldsymbol{e}}_{\square_k} \tag{7.12}$$

Expanding Equation 7.12 using the 4×4 skew-symmetrical matrix defined in Equation 7.9 finalizes the discrete integration of the angular rate into the quaternion attitude:

$$\begin{Bmatrix} e_{0\square_{k+1}} \\ e_{x\square_{k+1}} \\ e_{y\square_{k+1}} \\ e_{z\square_{k+1}} \end{Bmatrix} = \begin{bmatrix} \cos\left(s_{\square_k}\right) & -p_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & -q_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & -r_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} \\ p_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & \cos\left(s_{\square_k}\right) & r_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & -q_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} \\ q_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & -r_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & \cos\left(s_{\square_k}\right) & p_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} \\ r_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & q_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & -p_{\square_k}\Delta t\frac{\sin\left(s_{\square_k}\right)}{2s_{\square_k}} & \cos\left(s_{\square_k}\right) \end{bmatrix} \begin{Bmatrix} e_{0\square_k} \\ e_{x\square_k} \\ e_{y\square_k} \\ e_{z\square_k} \end{Bmatrix} \tag{7.13}$$

If there is no angular rate, meaning $\underline{\boldsymbol{\omega}}_{\square_k} = 0$ and causing $s_{\square_k} = 0$, the next quaternion attitude $\underline{\boldsymbol{e}}_{\square_{k+1}}$ is equal to the current one $\underline{\boldsymbol{e}}_{\square_k}$, i.e.,

$$\begin{Bmatrix} e_{0\square_{k+1}} \\ e_{x\square_{k+1}} \\ e_{y\square_{k+1}} \\ e_{z\square_{k+1}} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} e_{0\square_k} \\ e_{x\square_k} \\ e_{y\square_k} \\ e_{z\square_k} \end{Bmatrix} \quad \text{if } s_{\square_k} = 0 \tag{7.14}$$

7.1.2. Setpoint Integration

In this section, the accuracy of the discretization step of the setpoints $\underline{\boldsymbol{s}}$ implementation in the continuous process model f_P of Configuration 3 is improved over the Forward Euler method by deriving an analytical expression of the discretized system.

The discrete terms of a linear state-space state $\boldsymbol{A}$ and input $\boldsymbol{B}$ matrices are found, assuming the input $\underline{\boldsymbol{u}}$ is kept constant over the duration of the time step, working out the following:

$$\exp\left(\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ 0 & 0 \end{bmatrix}\Delta t\right) = \begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{B}_d \\ 0 & \boldsymbol{I} \end{bmatrix} \tag{7.15}$$

In order to apply Equation 7.15 to Equation 5.20 requires both the translational $\underline{\ddot{\boldsymbol{c}}}_P$ and angular jerk $\underline{\ddot{\boldsymbol{\omega}}}_P$ equations to be isolated from the rest of the equation. To accomplish this, the time lag variables, $\underline{\boldsymbol{\tau}}_{\ddot{c}}$ and $\underline{\boldsymbol{\tau}}_{\dot{\omega}}$, are assumed to be constant over one time-step Δt. Even though the derivative of the time-lag is non-zero, this change is expected to be small. The validity of this statement can be found in Section 6.2.3. This results in Equation 5.20 to be written in terms of state $\boldsymbol{A}$ and input $\boldsymbol{B}$ matrices, i.e.,

$$\begin{gathered}\dot{\underline{\boldsymbol{x}}} = \boldsymbol{A}\underline{\boldsymbol{x}} + \boldsymbol{B}\underline{\boldsymbol{u}} \\ \boldsymbol{A} = \left[-\frac{1}{\underline{\boldsymbol{\tau}}_{\ddot{c}}}\right] \qquad \boldsymbol{B} = \left[\frac{1}{\underline{\boldsymbol{\tau}}_{\ddot{c}}}\right]\end{gathered} \tag{7.16}$$

With,

$$\underline{\boldsymbol{x}} = [\underline{\ddot{\boldsymbol{c}}}_P] \qquad \underline{\boldsymbol{u}} = [\underline{\boldsymbol{s}}_{\ddot{c}}] \tag{7.17}$$

Finally applying Equation 7.15 to Equation 7.16, i.e.,

$$\exp\left(\begin{bmatrix} -\frac{1}{\underline{\boldsymbol{\tau}}_{\ddot{c}}} & \frac{1}{\underline{\boldsymbol{\tau}}_{\ddot{c}}} \\ 0 & 0 \end{bmatrix} \Delta t\right) = \begin{bmatrix} \exp\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\ddot{c}}}\right) & 1 - \exp\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\ddot{c}}}\right) \\ 0 & 1 \end{bmatrix} \tag{7.18}$$

Simplifying Equation 7.18, i.e.,

$$\begin{aligned}\underline{\ddot{\boldsymbol{c}}}_{P_{k+1}} &= \underline{\ddot{\boldsymbol{c}}}_{P_k} \exp\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\ddot{c}}}\right) + \underline{\boldsymbol{s}}_{\ddot{c}}\left(1 - \exp\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\ddot{c}}}\right)\right) \\ &= \exp\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\ddot{c}}}\right)\left(\underline{\ddot{\boldsymbol{c}}}_{P_k} - \underline{\boldsymbol{s}}_{\ddot{c}}\right) + \underline{\boldsymbol{s}}_{\ddot{c}}\end{aligned} \tag{7.19}$$

The same method applied to Equation 5.21 results in:

$$\underline{\dot{\boldsymbol{\omega}}}_{P_{k+1}} = \exp\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\dot{\omega}}}\right)\left(\underline{\dot{\boldsymbol{\omega}}}_{P_k} - \underline{\boldsymbol{s}}_{\dot{\omega}}\right) + \underline{\boldsymbol{s}}_{\dot{\omega}} \tag{7.20}$$

7.2. Motion

In this section both the motion as well as the method of generating said motion will be described in this section. The position and attitude of both the SRS and HMD are created from multiple superimposed sinusoidal function over time per axis. Meaning, that per Degree-of-Freedom (DoF), there will be superimposed sinusoids over time t. To simulate motion from rest, the motion will be scaled linearly for the first 10 s, these 10 s are indicated by T. Moreover, the motion is computed in the inertial frame of reference $\mathbb{E}_I$. The attitude is represented per axis in angle-axis formulation and transformed to quaternion representation by Equation 7.27 in Section 7.2.1. Given all this, the sinusoids for both the position and attitude are computed over time t, with amplitude A, and frequency f per axis as:

$$y(t) = \begin{cases} \frac{A}{T} t \sin(2\pi f t) & t \leq T \\ A \sin(2\pi f t) & t > T \end{cases} \tag{7.21}$$

In turn, only the velocity is computed per axis in the inertial frame of reference $\mathbb{E}_I$ by taking the derivative of Equation 7.21, i.e.,

$$y'(t) = \begin{cases} \frac{A}{T}\left(2\pi f t \cos(2\pi f t) + \sin(2\pi f t)\right) & t \leq T \\ 2\pi f A \cos(2\pi f t) & t > T \end{cases} \tag{7.22}$$

The second derivative of Equation 7.21 results in the acceleration per axis of the inertial frame of reference $\mathbb{E}_I$, i.e.,

$$y''(t) = \begin{cases} \frac{2\pi f A}{T}\left(-2\pi f t \sin(2\pi f t) + 2\cos(2\pi f t)\right) & t \leq T \\ -(2\pi f)^2 A \sin(2\pi f t) & t > T \end{cases} \tag{7.23}$$

7.2.1. Quaternion Attitude Derivative

However, the angular rate and angular acceleration are computed numerical. Whereby the difference between subsequent quaternions attitudes are transformed to angle-axis formulation and divided by the time-step Δt, i.e.,

$$\underline{\boldsymbol{\omega}}_k = \frac{\underline{\boldsymbol{e}}_{k+1} \ominus \underline{\boldsymbol{e}}_k}{\Delta t} \tag{7.24}$$

Here, the quaternion difference that results in an angle-axis attitude is symbolized by the quaternion deduction operator $\ominus$. In turn, the deduction operator $\ominus$ is defined as the quaternion multiplication $\underset{L}{\otimes}$ with the opposite attitude of the right-hand quaternion $\sim\underline{\boldsymbol{e}}_k$, the negative quaternion symbol $\sim$ explained in Equation 5.15. The result of the deduction should be transformed from quaternion to angle-axis notation by $\mathrm{Q2R}$, i.e.,

$$= \frac{\mathrm{Q2R}\left(\underline{\boldsymbol{e}}_{k+1} \underset{L}{\otimes} \sim\underline{\boldsymbol{e}}_k\right)}{\Delta t} \tag{7.25}$$

The effective rotation in angle-axis notation devided by the time-step Δt should result in an angular rotation in the same form and reference frame as measured by a MEMS based Inertial Measurement Unit (IMU) in the moving reference frame $\mathbb{E}_{\square}$.

The transformation from quaternion to angle-axis notation $\mathrm{Q2R}$ is found by reversing the angle-axis to quaternion transformation $\mathrm{R2Q}$. The $\mathrm{R2Q}$ transformation can be established by using Equation 7.13 and setting the right quaternion to a real unit quaternion $(1, 0, 0, 0)$ representing no rotation from the inertial reference frame, and thus leaving only the change in quaternion attitude written in terms of the effective rotation $\Delta t \underline{\boldsymbol{\omega}}_k$ in angle-axis notation [51], i.e.,

$$\underline{\boldsymbol{\Delta e}} = \begin{bmatrix} \Delta e_0 \\ \Delta e_x \\ \Delta e_y \\ \Delta e_z \end{bmatrix} = \begin{bmatrix} \cos(s_k) \\ p_k \Delta t \frac{\sin(s_k)}{2 s_k} \\ q_k \Delta t \frac{\sin(s_k)}{2 s_k} \\ r_k \Delta t \frac{\sin(s_k)}{2 s_k} \end{bmatrix} \quad \text{if } s_k \neq 0 \tag{7.26}$$

Here, the quaternion change in attitude $\underline{\boldsymbol{\Delta e}}$ with scalar Δe_0 and vector components $\Delta e_x, \Delta e_y, \Delta e_z$ are expressed in terms of an angle-axis attitude, i.e., constant angular rate $\underline{\boldsymbol{\omega}}_k$ for a time step Δt, with the effective rotation $\underline{\boldsymbol{\omega}}_k \Delta t$ and half the norm of the effective rotation s_k defined similarly to Equations 7.8 and 7.11 respectively.

If however there is no rotation, i.e. $s_k = 0$, a real unit quaternion is inserted instead. Including this information and writing Equation 7.26 in the ordered pair quaternion form, i.e. $\underline{\boldsymbol{e}} = (e_0, \underline{\boldsymbol{q}})$, as:

$$\underline{\boldsymbol{\Delta e}} = \begin{cases} \left(\cos(s_k), \underline{\boldsymbol{\omega}}_k \Delta t \frac{\sin(s_k)}{2 s_k}\right) & \text{if } s_k \neq 0 \\ \left(1, [0]_{3\times 1}\right) & \text{if } s_k = 0 \end{cases} \tag{7.27}$$

And will be further abbreviated as:

$$\underline{\boldsymbol{\Delta e}} = \mathrm{R2Q}\left(\underline{\boldsymbol{\omega}}_k \Delta t\right) \tag{7.28}$$

Having established the effective rotation in angle-axis form to quaternion attitude transformation $\mathrm{R2Q}$, the reverse transformation $\mathrm{Q2R}$ is found by first isolating the quaternion scalar term Δe_0 in Equation 7.26 and deducing half the norm of the rotation s_k, i.e.,

$$s_k = \arccos(e_0) \tag{7.29}$$

Following, the vector component of the quaternion $\underline{\boldsymbol{q}}$ is used to compute the effective rotation $\underline{\boldsymbol{\omega}}_k \Delta t$, i.e.,

$$\underline{\boldsymbol{\omega}}_k \Delta t = \mathrm{Q2R}\left(\underline{\boldsymbol{e}}\right) = \mathrm{Q2R}\left(e_0, \underline{\boldsymbol{q}}\right) \tag{7.30}$$

$$\underline{\boldsymbol{\omega}}_k \Delta t = \begin{cases} \underline{\boldsymbol{q}} \frac{2 s_k}{\sin(s_k)} & \text{if } s_k \neq 0 \\ [0]_{3\times 1} & \text{if } s_k = 0 \end{cases} \tag{7.31}$$

Using the transformation $\mathrm{Q2R}$ in Equation 7.31 the constant angular rate necessary to rotate the current quaternion attitude $\underline{\boldsymbol{e}}_{\square_k}$ to the next $\underline{\boldsymbol{e}}_{\square_{k+1}}$ can now be computed by Equation 7.25.

Next, the angular acceleration $\underline{\dot{\boldsymbol{\omega}}}$ is computed as the inverse of the integration method, the forward Euler method, i.e.,

$$\underline{\dot{\boldsymbol{\omega}}}_k = \frac{\underline{\boldsymbol{\omega}}_{k+1} - \underline{\boldsymbol{\omega}}_k}{\Delta t} \tag{7.32}$$

7.2.2. Motion Data

The motion of the SRS, as pointed out in [43], should be large enough to make sure to explore the nonlinearity in the SRS stewart platform [1]. The level of nonlinearity impacts the UKF Gaussian approximation in both the prediction as well in the correction phase. The frequency and amplitude of each sinusoid per axis is given in the table below, in Table 7.1. Inserting the frequencies and amplitudes, given in Table 7.1, in Equations 7.21, 7.22, 7.25, 7.23, and 7.32 results in the position, attitude, velocity, angular rate, acceleration and angular acceleration. The initial velocity, acceleration, attitude, angular rate, and angular acceleration are all set to zero, except for the initial position of the SRS UGP, which is initially -2.39 m removed over the z-axis from the Lower Gimbal Point (LGP), i.e. the motion reference point, the definition of which is stated in Section 4.4.

Table 7.1: Amplitudes and Frequencies of the sinusoidal motion of the SRS.

	Translation		**Rotation**	
SRS:	Amplitude [m]	Frequency [Hz]	Amplitude [rad]	Frequency [Hz]
x:	-0.1	0.1	-0.07	0.1
	0.1	0.25	0.07	0.25
	0.0123	0.65	0.0123	0.65
	0.016	0.85	0.016	0.85
	0.002	2.0	0.002	2.0
y:	0.1	-0.1	0.07	-0.1
	0.1	-0.25	0.07	-0.25
	0.0123	0.65	0.0123	0.65
	0.016	0.85	0.016	0.85
	0.002	2.0	0.002	2.0
z:	0.03	0.5	0.05	0.5
	0.002	2.0	0.002	2.0

The resulting translational and rotational motion of the SRS is shown in Figures 7.1 and 7.2 respectively. The attitude, in Figure 7.2, is presented in angle-axis formulation for better readability when compared to quaternion attitude formulation. The I subscript in the labels of the rotational figure is to indicate that the value is expressed in the inertial reference frame.

The frequency and amplitude of the motion of the HMD is presented in Table 7.2. The motion of the HMD is similar to the SRS motion with some extra movement on top. This does not reflect real live as the derivatives of the attitude of the SRS influence the velocity and acceleration of the HMD. However, this choice of movement of the HMD simplifies the generation of the input data and does not alter the validity of the observations. The initial velocity, acceleration, attitude, angular rate, and angular acceleration are all set to zero. Except for the initial position of the HMD, where the z-axis position starts at -3.0 m from the LGP.

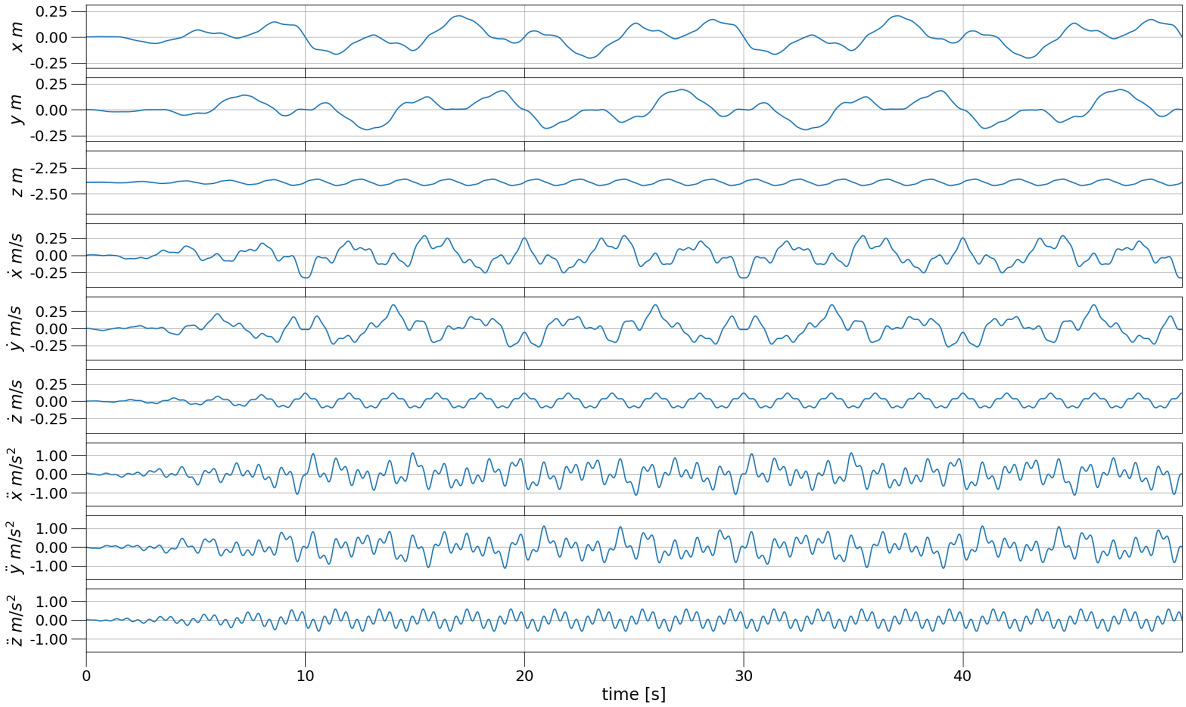


Figure 7.1: Position, velocity, and acceleration in the inertial reference frame $\mathbb{E}_I$ of SRS.

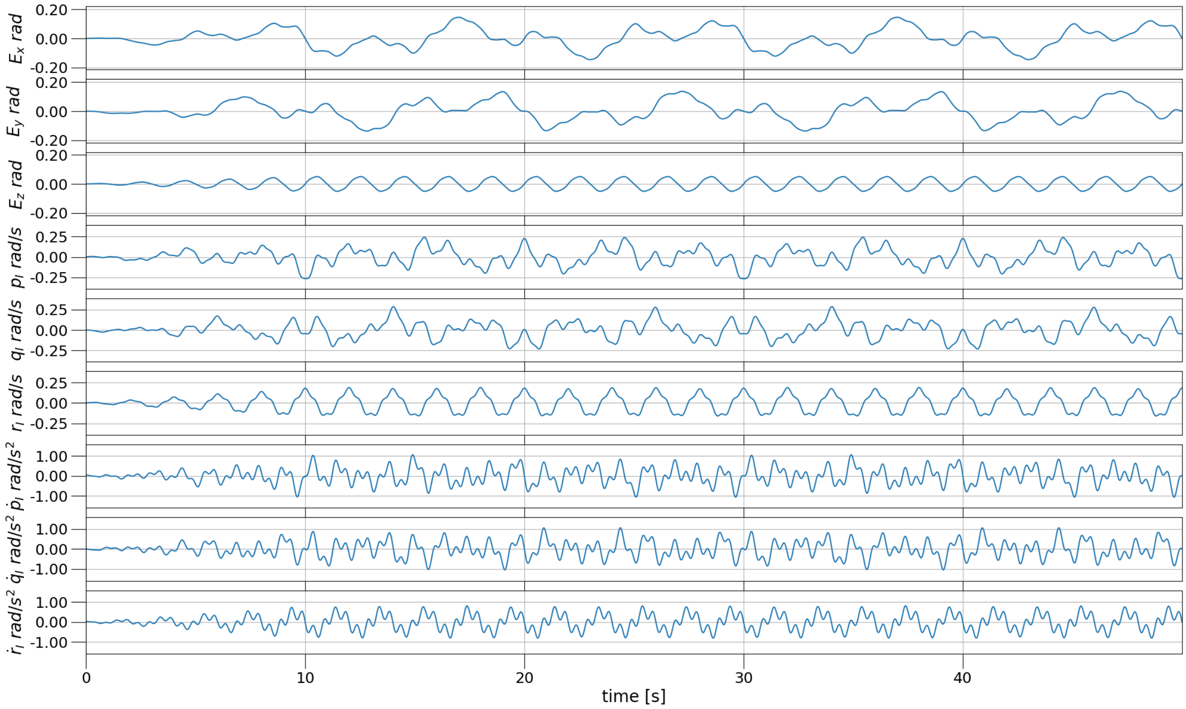


Figure 7.2: Angle-axis attitude, angular rate, and angular acceleration in the inertial reference frame $\mathbb{E}_I$ of SRS.

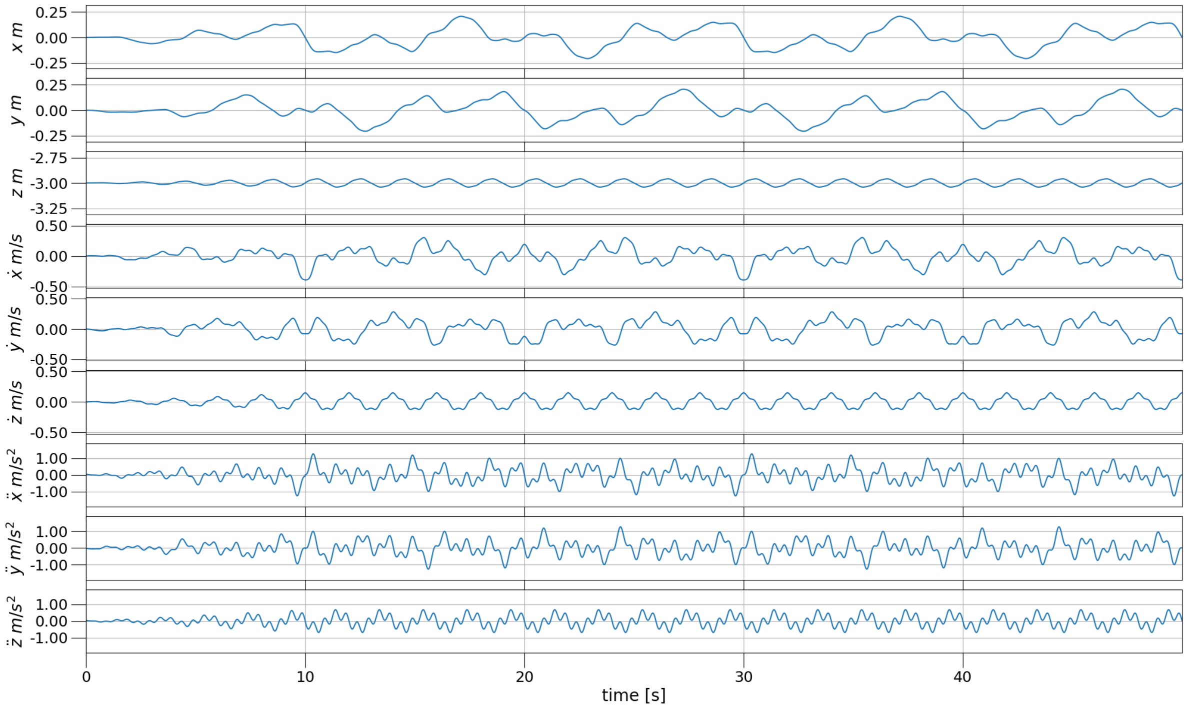


Figure 7.3: Position, velocity, and acceleration in the inertial reference frame $\mathbb{E}_I$ of the HMD.

Table 7.2: Amplitudes and Frequencies of the sinusoidal motion of the HMD on the user's head. The extra sinusoid motion, compared to the SRS motion, highlighted in gray.

	Translation		**Rotation**	
HMD:	Amplitude [m]	Frequency [Hz]	Amplitude [rad]	Frequency [Hz]
x:	-0.1	0.1	-0.07	0.1
	0.1	0.25	0.07	0.25
	0.0123	0.65	0.0123	0.65
	0.016	0.85	0.016	0.85
	0.002	2.0	0.002	2.0
	-0.02	0.5	-0.25	0.25
y:	0.1	-0.1	0.07	-0.1
	0.1	-0.25	0.07	-0.25
	0.0123	0.65	0.0123	0.65
	0.016	0.85	0.016	0.85
	0.002	2.0	0.002	2.0
	0.02	-0.5	0.25	-0.25
z:	0.03	0.5	0.05	0.5
	0.002	2.0	0.002	2.0
	0.01	0.5	0.05	0.25

The resulting translational and rotational motion of the HMD is shown in Figures 7.3 and 7.4 respectively. The attitude, in Figure 7.4, is presented in angle-axis formulation for better readability when compared to quaternion attitude formulation. The I subscript in the rotational figure is to indicate that the value is expressed in reference to the inertial reference frame.

7.2.3. Input and Observation Data Generation

Following the analytical description of the motion of both entities, SRS and HMD, the discrete measured input $\underline{u}_k$ and observation $\underline{y}_k$ of each sensor is sampled using the analytical motion. In this section the

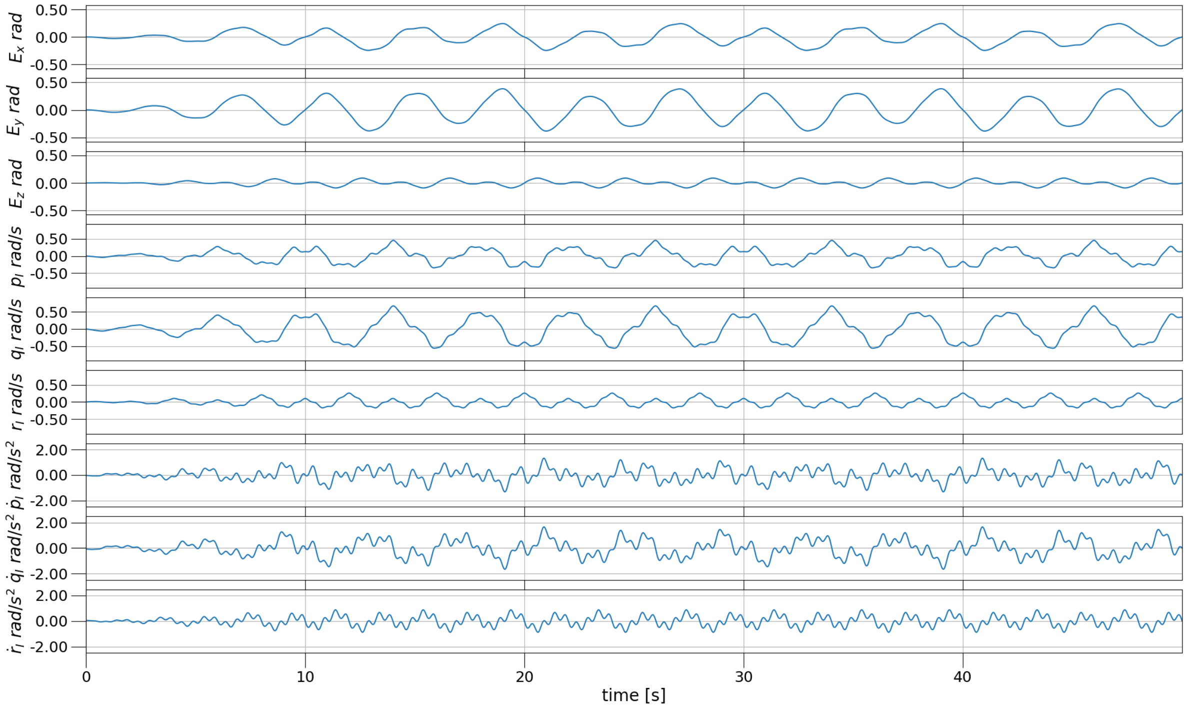


Figure 7.4: Angle-axis attitude, angular rate, and angular acceleration in the inertial reference frame $\mathbb{E}_I$ of the HMD.

data generation of each sensor is discussed and the method used explained.

First, the data generation of both IMU, one attached to the UGP of the SRS and the other one attached to the HMD itself, are discussed in Section 7.2.3.1. Second, the data generation of setpoints for the SRS is presented in Section 7.2.3.3. Last, the data generation of all observation sensors are given in Section 7.2.3.4.

Inertial Measurement Unit: Specific Force and Rate

Both IMU, described in Chapter 6, measure both the specific force $\underline{\boldsymbol{f}}_{\Box_{m_k}}$ and angular rate $\underline{\boldsymbol{\omega}}_{\Box_{m_k}}$ discretely in their respective non-inertial reference frame $\mathbb{E}_\Box$, i.e. the reference frame attached to the HMD $\mathbb{E}_H$ and the reference frame attached to the center of the UGP $\mathbb{E}_P$ of the SRS. In order to simulate the measurements the actual data is necessary.

First, the discrete actual specific force $\underline{\boldsymbol{f}}_{\Box_k}$ for both entities, SRS and HMD, IMU is computed by first deducting the gravity $\underline{\boldsymbol{g}}$ from the acceleration $\underline{\ddot{\boldsymbol{c}}}_{\Box_k}$. The acceleration $\underline{\ddot{\boldsymbol{c}}}_{\Box_k}$ of each entity, SRS and HMD, is sampled at the discrete sample rate of the respective IMU from the analytical acceleration, presented in Figures 7.1 and 7.3 by computing the data in Tables 7.1 and 7.2 with Equation 7.23. Both the sampled acceleration $\underline{\ddot{\boldsymbol{c}}}_{\Box_k}$ and gravity $\underline{\boldsymbol{g}}$, are defined in the inertial frame $\mathbb{E}_I$ in this case. The specific force each entity IMU is subjected to can be computed by the opposite operation of that in Equation 4.30 and transferring the result to the entity respective non-inertial frame $\mathbb{E}_\Box$, i.e.,

$$
\begin{aligned}
\underline{\boldsymbol{f}}_{\Box_k} &= \underline{\ddot{\boldsymbol{c}}}_{\Box_k} - \underline{\boldsymbol{g}} \\
\begin{pmatrix} f_{\ddot{x}_{\Box_k}} & f_{\ddot{y}_{\Box_k}} & f_{\ddot{z}_{\Box_k}} \end{pmatrix}\{\mathbb{E}_P\} &= \begin{pmatrix} \ddot{x}_{\Box_k} & \ddot{y}_{\Box_k} & \ddot{z}_{\Box_k} \end{pmatrix}\{\mathbb{E}_I\} - \begin{pmatrix} 0 & 0 & g \end{pmatrix}\{\mathbb{E}_I\} \\
\begin{pmatrix} f_{\ddot{x}_{\Box_k}} & f_{\ddot{y}_{\Box_k}} & f_{\ddot{z}_{\Box_k}} \end{pmatrix} &= \begin{pmatrix} \ddot{x}_{\Box_k} & \ddot{y}_{\Box_k} & \ddot{z}_{\Box_k} - g \end{pmatrix} \boldsymbol{T}_{\Box I}^T
\end{aligned}
\tag{7.33}
$$

Next, the discrete actual angular rate $\underline{\boldsymbol{\omega}}_{\Box_k}$ each entity IMU is subject to is computed. The angular rate $\underline{\boldsymbol{\omega}}_{\Box_k}$, presented in Figures 7.2 and 7.4 by computing the data in Tables 7.1 and 7.2 with Equations 7.21 and 7.25, is in the inertial reference frame $\mathbb{E}_I$. The actual angular rate $\underline{\boldsymbol{\omega}}_{\Box_k}$ in the non-inertial reference frame $\mathbb{E}_\Box$ is found by applying the transformation matrix $\boldsymbol{T}_{\Box I}$ using the required and already established quaternion attitude representation, i.e.,

$$\begin{aligned} \begin{pmatrix} p_{\square_k} & q_{\square_k} & r_{\square_k} \end{pmatrix} \{\mathbb{E}_{\square}\} &= \begin{pmatrix} p_{I_{\square_k}} & q_{I_{\square_k}} & r_{I_{\square_k}} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} p_{\square_k} & q_{\square_k} & r_{\square_k} \end{pmatrix} &= \begin{pmatrix} p_{I_{\square_k}} & q_{I_{\square_k}} & r_{I_{\square_k}} \end{pmatrix} \boldsymbol{T}_{\square I}^{T} \end{aligned} \tag{7.34}$$

Inertial Measurement Unit: Noise

Both discrete biases of the IMU, accelerometer bias $\underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_{\square_k}}$ and angular rate bias $\underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_{\square_k}}$ are modelled using the accumulation of their respective Brownian Gaussian noise distribution, $\aleph\left(0, \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{f}_{\mu_{\square}}}\right)$ and $\aleph\left(0, \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\omega}_{\mu_{\square}}}\right)$ respectively, over time. The discrete accelerometer biasses $\underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_{\square_k}}$ are modelled as:

$$\begin{aligned} \underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_{\square_{k+1}}} &= \underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_{\square_k}} + \Delta t \underline{\boldsymbol{\mu}}_{\boldsymbol{f}_{\square_k}} \\ &= \underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_{\square_k}} + \Delta t \aleph\left(0, \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{f}_{\mu_{\square}}}\right) \end{aligned} \tag{7.35}$$

And in the same manner the discrete angular rate bias $\underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_{\square_k}}$ is computed, i.e.,

$$\begin{aligned} \underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_{\square_{k+1}}} &= \underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_{\square_k}} + \Delta t \underline{\boldsymbol{\mu}}_{\boldsymbol{\omega}_{\square_k}} \\ &= \underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_{\square_k}} + \Delta t \aleph\left(0, \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\omega}_{\mu_{\square}}}\right) \end{aligned} \tag{7.36}$$

Following, white noise $\underline{\boldsymbol{n}}$ and bias $\underline{\boldsymbol{\lambda}}$ are superimposed on the actual discrete results computed in Equations 7.33 and 7.34 following the modeling specified in Equation 4.8, i.e.,

$$\begin{aligned} \underline{\boldsymbol{f}}_{\square_{m_k}} &= \underline{\boldsymbol{f}}_{\square_k} + \underline{\boldsymbol{n}}_{\boldsymbol{f}_{\square}} + \underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_{\square_k}} \\ \underline{\boldsymbol{f}}_{\square_{m_k}} &= \underline{\boldsymbol{f}}_{\square_k} + \aleph\left(0, \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{f}_{\square}}\right) + \underline{\boldsymbol{\lambda}}_{\boldsymbol{f}_{\square_k}} \end{aligned} \tag{7.37}$$

And in the discrete measured angular rate is computed in the same manner, i.e.,

$$\begin{aligned} \underline{\boldsymbol{\omega}}_{\square_{m_k}} &= \underline{\boldsymbol{\omega}}_{\square_k} + \underline{\boldsymbol{n}}_{\boldsymbol{\omega}_{\square}} + \underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_{\square_k}} \\ \underline{\boldsymbol{\omega}}_{\square_{m_k}} &= \underline{\boldsymbol{\omega}}_{\square_k} + \aleph\left(0, \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\omega}_{\square}}\right) + \underline{\boldsymbol{\lambda}}_{\boldsymbol{\omega}_{\square_k}} \end{aligned} \tag{7.38}$$

Setpoints

In this section the setpoint generation of the SRS used for modeling purposes are explained. Equations 7.19 and 7.20 are written in terms of their second order setpoints, $\underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}}$ and $\underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}}$. Both equations becoming a function of the current and next discrete acceleration $\underline{\ddot{\boldsymbol{c}}}_{P_k}$ and angular acceleration $\underline{\dot{\boldsymbol{\omega}}}_{P_{k+1}}$ of the SRS, i.e.,

$$\underline{\boldsymbol{s}}_{\ddot{\boldsymbol{c}}_k} = \frac{\underline{\ddot{\boldsymbol{c}}}_{P_{k+1}} - \underline{\ddot{\boldsymbol{c}}}_{P_k} \text{exp}\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\ddot{\boldsymbol{c}}}}\right)}{1 - \text{exp}\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\ddot{\boldsymbol{c}}}}\right)} \tag{7.39}$$

$$\underline{\boldsymbol{s}}_{\dot{\boldsymbol{\omega}}_k} = \frac{\underline{\dot{\boldsymbol{\omega}}}_{P_{k+1}} - \underline{\dot{\boldsymbol{\omega}}}_{P_k} \text{exp}\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\dot{\boldsymbol{\omega}}}}\right)}{1 - \text{exp}\left(\frac{-\Delta t}{\underline{\boldsymbol{\tau}}_{\dot{\boldsymbol{\omega}}}}\right)} \tag{7.40}$$

The acceleration setpoints are based in the inertial frame and no changes are required. However, the angular acceleration of the motion is computed in the inertial frame $\mathbb{E}_I$. Here, a transfer to the upper moving platform $\mathbb{E}_P$ is required, i.e.,

$$\begin{aligned} \begin{pmatrix} \dot{p}_{P_k} & \dot{q}_{P_k} & \dot{r}_{P_k} \end{pmatrix} \{\mathbb{E}_P\} &= \begin{pmatrix} \dot{p}_{I_{P_k}} & \dot{q}_{I_{P_k}} & \dot{r}_{I_{P_k}} \end{pmatrix} \{\mathbb{E}_I\} \\ \begin{pmatrix} \dot{p}_{P_k} & \dot{q}_{P_k} & \dot{r}_{P_k} \end{pmatrix} &= \begin{pmatrix} \dot{p}_{I_{P_k}} & \dot{q}_{I_{P_k}} & \dot{r}_{I_{P_k}} \end{pmatrix} \boldsymbol{T}_{PI}^{T} \end{aligned} \tag{7.41}$$

Observation

In order to generate the data of the observational sensors, namely the linear encoders of the SRS and the TrackIR 5 of the Virtual Reality (VR) system, the respective observational models are used. For the observation of all the observational sensors, the actual state $\underline{\boldsymbol{x}}_k$ is sampled through the respective observation models $\mathrm{H}_{\square}$, outlined in Sections 4.4.2 and 5.2.2. The noise is superimposed as a Gaussian distribution on the samples taken in place of the observation random noise values $\underline{\boldsymbol{o}}_{\square}$.

In the case of the observation model H_P of the SRS, Equation 4.44, this results in:

$$\left.\begin{aligned} \underline{\boldsymbol{y}}_{P_k} &= \left\{l_{i_k} + \aleph\left(0, \sigma_{l_i}^2\right)\right\} \\ &= \left\{\left\|\begin{pmatrix} x_{P_k} & y_{P_k} & z_{P_k}\end{pmatrix} + \begin{pmatrix} p_{x_i} & p_{y_i} & 0\end{pmatrix} \boldsymbol{T}_{PI} - \begin{pmatrix} b_{x_i} & b_{y_i} & 0\end{pmatrix}\right\| + \aleph\left(0, \sigma_{l_i}^2\right)\right\} \end{aligned}\right\} \forall i = 1, \dots, 6 \tag{7.42}$$

And in the case of the observation model H_{HP} of the VR system, found in Equation 5.18, becomes:

$$\begin{aligned} \underline{\boldsymbol{y}}_{HP_k} &= \begin{Bmatrix} x_{HP_k} \\ y_{HP_k} \\ z_{HP_k} \\ \psi_{HP_k} \\ \theta_{HP_k} \\ \phi_{HP_k} \end{Bmatrix} + \begin{Bmatrix} \aleph\left(0, \sigma_{x_H}^2\right) \\ \aleph\left(0, \sigma_{y_H}^2\right) \\ \aleph\left(0, \sigma_{z_H}^2\right) \\ \aleph\left(0, \sigma_{\psi_H}^2\right) \\ \aleph\left(0, \sigma_{\theta_H}^2\right) \\ \aleph\left(0, \sigma_{\phi_H}^2\right) \end{Bmatrix} \\ &= \begin{Bmatrix} \boldsymbol{T}_{PI} \begin{Bmatrix} x_{H_k} - x_{P_k} \\ y_{H_k} - y_{P_k} \\ z_{H_k} - z_{P_k} \end{Bmatrix} \\ \text{arctan}\left(\left(e_{0_{HP_k}} e_{x_{HP_k}} + e_{y_{HP_k}} + e_{z_{HP_k}}\right), \left(0.5 - e_{x_{HP_k}} - e_{y_{HP_k}}\right)\right) \\ \text{arcsin}\left(2\left(e_{0_{HP_k}} e_{y_{HP_k}} - e_{x_{HP_k}} e_{z_{HP_k}}\right)\right) \\ \text{arctan}\left(\left(e_{0_{HP_k}} e_{z_{HP_k}} + e_{x_{HP_k}} + e_{y_{HP_k}}\right), \left(0.5 - e_{y_{HP_k}} - e_{z_{HP_k}}\right)\right) \end{Bmatrix} + \begin{Bmatrix} \aleph\left(0, \sigma_{x_H}^2\right) \\ \aleph\left(0, \sigma_{y_H}^2\right) \\ \aleph\left(0, \sigma_{z_H}^2\right) \\ \aleph\left(0, \sigma_{\psi_H}^2\right) \\ \aleph\left(0, \sigma_{\theta_H}^2\right) \\ \aleph\left(0, \sigma_{\phi_H}^2\right) \end{Bmatrix} \end{aligned} \tag{7.43}$$

7.2.4. Analytical vs. numerical Integration

In this section, the influence of the discrete integration method, the forward Euler method, is analyzed as it will increase and color the perceived noise in the process model. Certain mitigations will be proposed in order to adapt to the increased noise and reduce the coloring of said noise.

A discrete integration method, such as the forward Euler method, can introduce a numerical error in the prediction phase of the UKF. By using the 'more true to real life' analytically generated translational motion, see Sections 7.2 and 7.2.3, the induced error by the forward Euler integration method is analyzed in this section. Due to the numerical method used for the attitude and its derivatives lack this error and therefore cannot be analyzed.

The analytical generated higher order state values are different from the required discrete higher order state values. The term 'required', in this case, meaning the value that is necessary for the integration method to compute the lower order state value. The numerical errors introduced in the integration method in the process model of an UKF should be taken into account and solved by increasing the process noise [36]. The process noise should cover for the whole process model, including its integration method, rather than just the sensor noises.

For example, the induced error on the IMU of the SRS due to its translational motion profile on the x-axis, presented in Section 7.2, is found by first computing the required discrete velocity. Where, the required discrete velocity $\dot{\underline{\boldsymbol{c}}}_{\text{req.}_k}$ is the velocity needed for the forward Euler integration to compute the correct position and can thus be defined as:

$$\dot{\underline{\boldsymbol{c}}}_{\text{req.}_k} = \frac{\underline{\boldsymbol{c}}_{k+1} - \underline{\boldsymbol{c}}_k}{\Delta t} \tag{7.44}$$

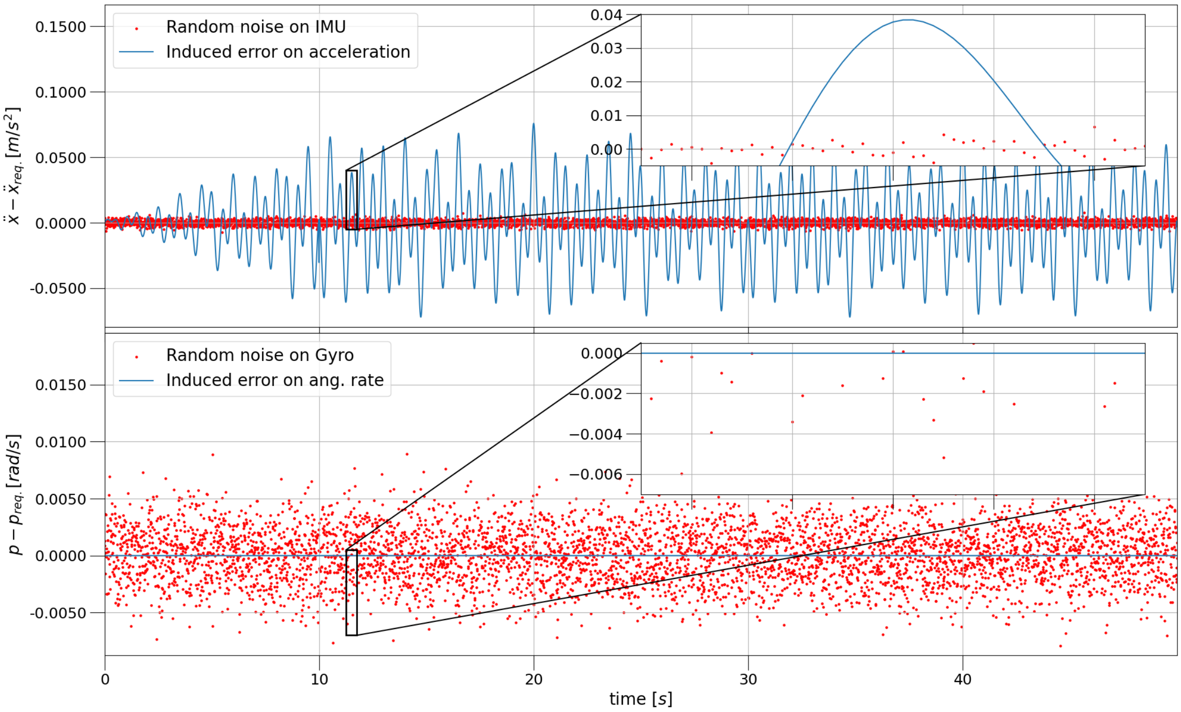


Figure 7.5: Plotting of the induced error on the velocity, acceleration, and angular rate of the SRS at 100 Hz.

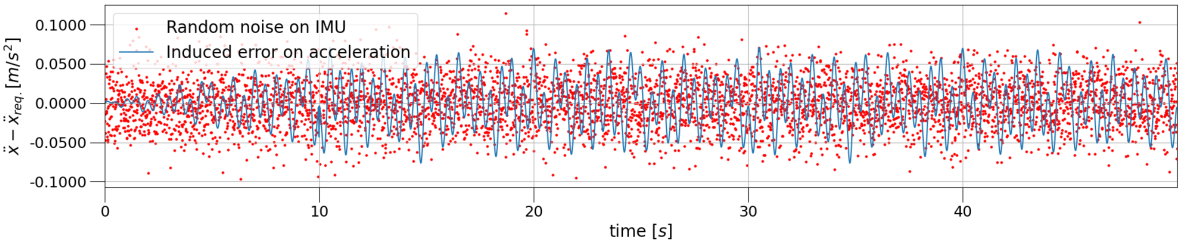


Figure 7.6: Plotting of the induced error on the velocity, acceleration, and angular rate of the HMD at 100 Hz.

Following, the required discrete acceleration $\underline{\ddot{\boldsymbol{c}}}_{\text{req.}_k}$ is the acceleration needed for the forward Euler integration to compute the correct required discrete velocity $\underline{\dot{\boldsymbol{c}}}_{\text{req.}_k}$, i.e.,

$$\underline{\ddot{\boldsymbol{c}}}_{\text{req.}_k} = \frac{\underline{\dot{\boldsymbol{c}}}_{\text{req.}_{k+1}} - \underline{\dot{\boldsymbol{c}}}_{\text{req.}_k}}{\Delta t} \tag{7.45}$$

The acceleration induced error, due to the forward Euler integration, is computed as $\ddot{x}_k - \ddot{x}_{\text{req.}_k}$. The acceleration induced errors of the both entities, SRS and HMD, are plotted in Figures 7.5 and 7.6 at 100 Hz. The angular rate induced error is zero in both entities cases as the data was generated discretely, see Section 7.2, but plotted in case of the SRS for a comparison later on in this section to Figure 7.7, where a higher integration frequency was used. The red scattered dots in the plot, are a representation of the white noise present on the IMU of the SRS as presented in Section 6.2.2. When examining both plots the importance of the ratio between the strength of the white noise to the numerical error of the motion is of importance.

Even though the movement pattern, and thus the induced error, is largely the same, between the SRS and HMD, the fact that the noise strength of the IMU is ten times larger makes the ratio different. Analyzing the noise superimposed over the induced error, i.e. the combined error, can give an indication of the actual process noise applicable to the process model. However, it must be noted that the combined error is no longer Gaussian as the underlying motion clearly permeates through.

For example the x-axis, taking the Standard Deviation (SD) of the combined error on the x-axis of the

Table 7.3: Forward Euler integration method induced error on the IMU of the SRS at 100 Hz.

SRS @ 100 Hz	Standard deviation Accelerometer $[\mathrm{m/s^2}]$	 Combined error $[\mathrm{m/s^2}]$	 $\times$ Scale
x:	1.9×10^{-3}	3.13×10^{-2}	16.4
y:	2.1×10^{-3}	3.14×10^{-2}	14.9
z:	1.8×10^{-3}	2.69×10^{-2}	14.9

Table 7.4: Forward Euler integration method induced error on the IMU of the HMD at 100 Hz.

HMD @ 100 Hz	Standard deviation Accelerometer $[\mathrm{m/s^2}]$	 Combined error $[\mathrm{m/s^2}]$	 $\times$ Scale
x:	3.0×10^{-2}	4.30×10^{-2}	1.4
y:	2.9×10^{-2}	4.32×10^{-2}	1.5
z:	4.7×10^{-2}	5.42×10^{-2}	1.2

SRS results in 3.13×10^{-2} m/s^2, scaled by 16.4 from the SD on the accelerometer of 1.9×10^{-3} m/s^2. Moreover, the SD of the combined error on the x-axis of the HMD results in 4.30×10^{-2} m/s^2, scaled by 1.4 from the SD on the accelerometer of 3.0×10^{-2} m/s^2. In the case of the SRS, the forward Euler method induced error is very large compared to the noise on the accelerometer. So much so, that improving the SD on the accelerometer by a factor of 1000 would result in the SD on the combined error of 3.12×10^{-2} m/s^2. The rest of the combined errors SD and their impact on the perceived accelerometer precision is found in Tables 7.3 and 7.4.

One of the solution would be to increase the process noise to the level indicated by the combined error SD. This, however, reduces the apparent quality of the accelerometer used. Moreover, this solution was applied by Miletović et al. described in [43], where the the process noise $\boldsymbol{Q}$, comprised of the IMU variance, was scaled by a factor of 1000, manually found by tuning the filter on measured data. The factor 1000 would equate to scaling the SD by $\sqrt{1000} \approx 31.62$. It must be noted that the motion used was different and that the jerk, third time deviation of position, in the motion matters how much induced error is present.

It is known that the induced error due to discrete integration methods, such as the forward Euler method, can be reduced by reducing the time-step Δt, i.e. increasing the computational frequency. A question to be answered would be what the reduction in integration induced error is if the entity, in this case the SRS, transmits its data at only 100 Hz. For the following analysis the computational frequency is raised to 600 Hz, dividable by both 120 Hz, frequency the TrackIR 5 runs at, and 100 Hz. The result of increasing the frequency to 600 Hz is presented in Figures 7.7 and 7.8.

In Figure 7.7, the IMU of the SRS sample-rate is kept at a 100 Hz but the computations are run at 600 Hz. Meaning, that the velocity and position are updated six times with every accelerometer reading. In the zoomed in area of Figure 7.7, the induced error is seen to increase sharply every six time-steps as the accelerometer measurement is held constant for the next six measurements. The increased frequencies of the velocity and positional updates decreases the forward Euler integration induced error on the x-axis to a combined error SD of 2.03×10^{-2} m/s^2, a decrease of $35\,\%$. The combined error per axis is presented in Table 7.5.

In the UKF algorithm, this method of updating more frequently with the same sensor reading has the added benefit that it is easier to maintain a proper state covariance $\boldsymbol{P}_{\hat{x}}$ at every prediction step. A state covariance with only a part up to date would lead to suboptimal performance of the UKF in general.

A similar pattern is observed in the induced error on the angular rate in Figure 7.7. Here, the induced error seemingly increased by increasing the computational frequency. The reason for this behavior is due to the fact that the actual angular rate uses the same computation as the required angular rate, i.e. the actual angular rate is computed discretely. By computing the required angular rate at 600 Hz,

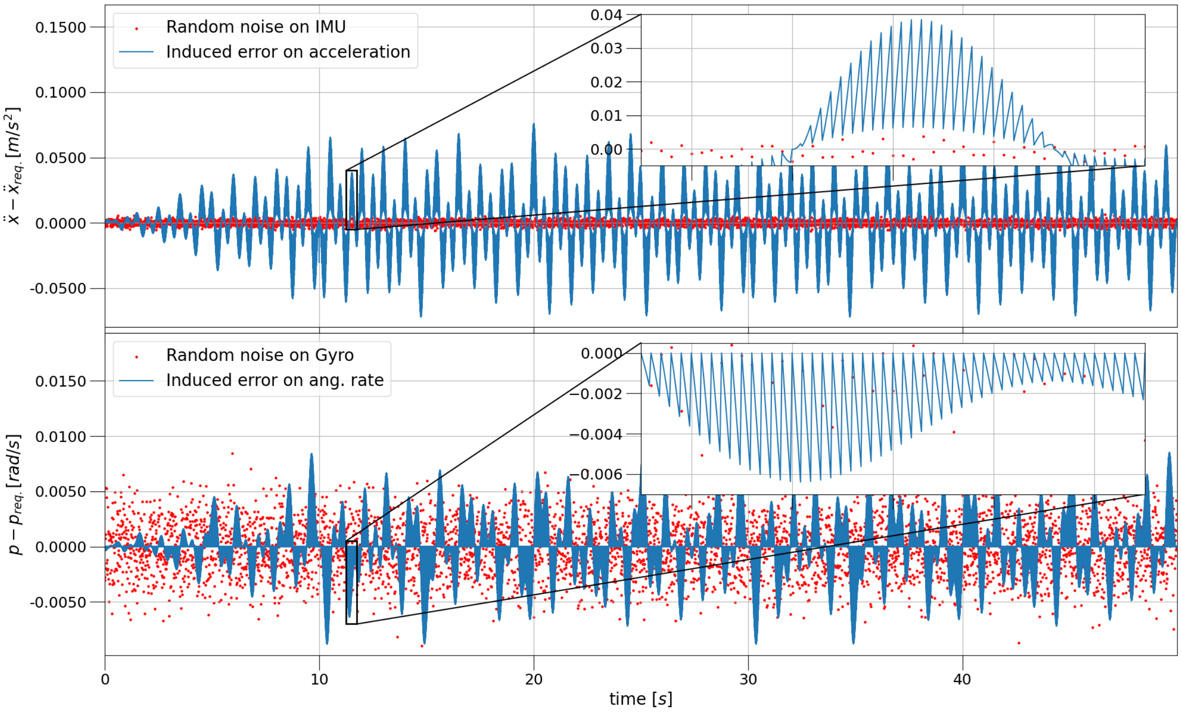


Figure 7.7: Plotting of the induced error on the velocity, acceleration, and angular rate of the SRS at 600 Hz.

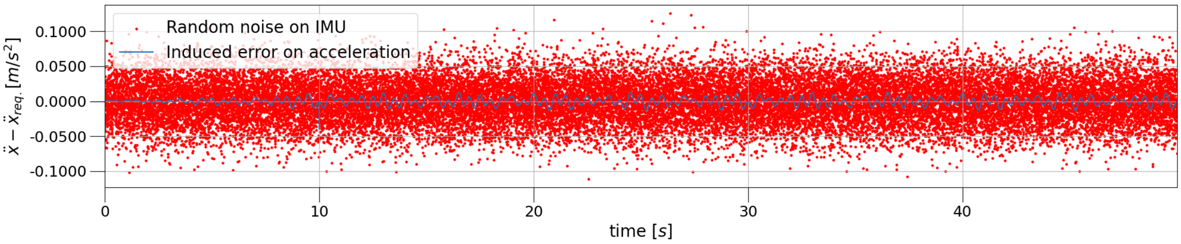


Figure 7.8: Plotting of the induced error on the velocity and acceleration of the HMD at 600 Hz.

Table 7.5: Forward Euler integration method induced error on the IMU of the SRS sampled at 100 Hz but computed at 600 Hz.

SRS @ 600 Hz			
	Standard deviation		
	Accelerometer $[\mathrm{m/s^2}]$	Combined error $[\mathrm{m/s^2}]$	$\times$ Scale
x:	1.9×10^{-3}	2.03×10^{-2}	10.7
y:	2.1×10^{-3}	2.05×10^{-2}	9.8
z:	1.8×10^{-3}	1.75×10^{-2}	9.7
	Gyroscope $[\mathrm{rad/s}]$	Combined error $[\mathrm{rad/s}]$	$\times$ Scale
p:	2.4×10^{-3}	3.15×10^{-3}	1.3
q:	4.1×10^{-3}	4.58×10^{-3}	1.1
r:	3.3×10^{-3}	3.77×10^{-3}	1.1

Table 7.6: Discrete integration methods induced errors.

HMD @ 600 Hz			
	Standard deviation		
	Accelerometer $[\mathrm{m/s^2}]$	Combined error $[\mathrm{m/s^2}]$	$\times$ Scale
x:	3.0×10^{-2}	3.05×10^{-2}	1.0
y:	2.9×10^{-2}	2.96×10^{-2}	1.0
z:	4.7×10^{-2}	4.70×10^{-2}	1.0

and using only every sixth element to alter the attitude, the angular rate is only based on the last five predictions. This is exactly what is shown in the figure and will increase the perceived noise in the simulation of the gyroscope of the SRS. Moreover, this behavior brings it closer in line to reality. Taking this behavior to the extreme, i.e. increasing the frequency at which the discrete motion is computed, would eventually lead to exactly the continuous figures.

In Figure 7.8, the induced error on the acceleration on the x-axis of an IMU of the HMD read at 600 Hz is plotted next to the accelerometer noise as red dots for perspective. The induced error is now much smaller than the noise present on the accelerometer itself in this case. The combined errors per axis of the IMU of the HMD can be found in Table 7.6.

It can be concluded that the selection of the computational frequency, and choice of integration method should be dictated by the quality of the IMU, expected jerk in the motion profile and computation power. Moreover, not accounting for this induced integration error in the process noise Q will lead to overconfidence in the prediction phase of the UKF.

7.3. UKF Quaternion Implementation

In this Section, the UKF implementation specificities are discussed. The quaternion attitude representation used throughout this Thesis has many advantages over its counterparts such as the Euler-angle attitude representation. The main advantage is its computational cost due to the linear kinematic equations of the attitude, see Equation 4.33, and the simple transformation matrices, see Equation 4.6, when compared to the Euler-angle attitude representation [51].

However, care must be taken when computing the weighted mean and covariance, in Equations 3.20, 3.21, 3.23, and 3.24 of the UKF, of the quaternion values without any regard of their origin. This is due to the fact that a four dimensional unit-quaternion represents the three dimensional attitude space in a two to one mapping. The weighted mean and covariance of the quaternion are discussed in Sections 7.3.1 and 7.3.2 respectively.

7.3.1. Quaternion Averaging

The UKF requires the computation of the weighted mean of the state as per Equation 3.20. Computing said equation results in the proper weighted average state's scalars except for the weighted average attitude.

By simply taking the average individually of the quaternion scalar e_0 and its vector components (e_x, e_y, e_z) two problems arise.

The first problem is that the resulting quaternion would not be of unit length. For example, given two quaternions, $(1 \quad 0 \quad 0 \quad 0)$ and $(0 \quad 0 \quad 1 \quad 0)$, the individual component average would be $(0.5 \quad 0 \quad 0.5 \quad 0)$ while the proper quaternion attitude average should be $(\sqrt{2}/2 \quad 0 \quad \sqrt{2}/2 \quad 0)$. However, this could be fixed afterwards by dividing the resulting non-unit quaternion by its norm, i.e. $\underline{\boldsymbol{e}}/\|\underline{\boldsymbol{e}}\|$.

The second problem is caused by the fact that a quaternion with all individual components negative represent the same attitude as the positive ones, due to the quaternion's characteristic two to one mapping of the rotational space [61], i.e.,

$$\begin{pmatrix} e_0 & e_x & e_y & e_z \end{pmatrix} = \begin{pmatrix} -e_0 & -e_x & -e_y & -e_z \end{pmatrix} \tag{7.46}$$

Changing the sign of the quaternion scalars in (weighted) averaging equations, such as Equation 3.20, should not change the average attitude but taking the individual component average of a quaternion would.

In [37] it is proven that average quaternion $\bar{\underline{e}}$ is the result from solving the following maximization equation:

$$\bar{\underline{\boldsymbol{e}}} = \arg\max_{\boldsymbol{e}\in} \underline{\boldsymbol{e}}^T \boldsymbol{M} \underline{\boldsymbol{e}} \tag{7.47}$$

Where the 4×4 matrix $\boldsymbol{M}$ is defined as:

$$\boldsymbol{M} = \sum_{i=1}^{2L} \boldsymbol{w}_i^{(m)} \underline{\boldsymbol{e}}_i \underline{\boldsymbol{e}}_i^T$$

Here, the weights are represented as $\boldsymbol{w}_i^{(m)}$ and $\underline{\boldsymbol{e}}_i \underline{\boldsymbol{e}}_i^T$ simply means the each quaternion's individual components covariance, i.e.,

$$= \sum_{i=1}^{2L} \boldsymbol{w}_i^{(m)} \begin{bmatrix} e_0 \\ e_x \\ e_y \\ e_z \end{bmatrix} \cdot \begin{bmatrix} e_0 & e_x & e_y & e_z \end{bmatrix} \tag{7.48}$$

The average quaternion $\bar{\underline{e}}$, the solution to this maximization problem, in Equation 7.47, is proven in [72] to be the eigenvector of the 4×4 matrix $\boldsymbol{M}$ with the highest eigenvalue.

7.3.2. Quaternion Covariance

Taking a 'scalar covariance' of a quaternion results neither in the attitude covariance nor a unit-quaternion. The solution is to represent any covariance of a quaternion as the angle-axis covariance [9]. The angle-axis form, contrary to the quaternion notation, is mapped to three-dimensional rotational space. Moreover, a convenient transformation to and from quaternion form exist, see Equations 7.27 and 7.31. Both the linearly mapping and convenient transformations make the angle-axis attitude form a solution for the covariance of the attitude. Also, it must be noted that the contribution to the total degrees of freedom L due to the attitude are three, in line with the expectation from the angle-axis form. This is crucial for the computation of the sigma-point weights $\boldsymbol{w}$. The angle-axis form of an attitude or effective rotation was given in Equation 7.8. The angle-axis representation for the uncertainty of the state's attitude results in amendments to the UKF. Two kind of functions in the UKF need to be examined.

First, the difference between quaternion state attitudes should result in an angle-axis from which the covariance $\boldsymbol{P}$ can be computed as with any other scalar. The deduction operator $\ominus$ deducting two unit-quaternions, $\underline{\boldsymbol{e}}_1$ and $\underline{\boldsymbol{e}}_2$, and presents the resulting attitude in an angle-axis form. The difference between unit-quaternions is first resolved by quaternion multiplying $\underset{L}{\otimes}$ the opposite $\sim$ quaternion before transformed in angle-axis form $\mathrm{Q2R}$, i.e.,

$$\underline{\boldsymbol{e}}_1 \ominus \underline{\boldsymbol{e}}_2 = \mathrm{Q2R}\left(\underline{\boldsymbol{e}}_1 \underset{L}{\otimes} {\sim}\underline{\boldsymbol{e}}_2\right) \tag{7.49}$$

The difference operator $\ominus$ is used to alter Equations 3.21 and 3.25 to signify the altered subtraction operation for the attitude specifically. The predicted state covariance $\boldsymbol{P}^{-}_{\hat{x}\ k}$ computation in Equation 3.21 results in:

$$\boldsymbol{P}^{-}_{\hat{x}\ k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left[\boldsymbol{\mathcal{X}}^{x,-}_{k,i} \ominus \underline{\hat{\boldsymbol{x}}}^{-}_{k}\right]^T \left[\boldsymbol{\mathcal{X}}^{x,-}_{k,i} \ominus \underline{\hat{\boldsymbol{x}}}^{-}_{k}\right] \tag{7.50}$$

And the cross-covariance between the predicted state $\underline{\hat{\boldsymbol{x}}}^{-}_{k}$ and the estimated observation $\underline{\hat{\boldsymbol{y}}}_k$ in Equation 3.25 becomes:

$$\boldsymbol{P}_{\hat{x}_k \hat{y}_k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left[\boldsymbol{\mathcal{X}}^{x,-}_{k,i} \ominus \underline{\hat{\boldsymbol{x}}}^{-}_{k}\right]^T \left[\boldsymbol{\mathcal{Y}}^{-}_{k} - \underline{\hat{\boldsymbol{y}}}_k\right] \tag{7.51}$$

It must be noted that the $\ominus$ operator only modifies the attitude subtraction, for all other scalars the $\ominus$ operator is the same as the $-$ operator. Due to the transformation from quaternion to angle-axis form, the dimension of the state covariance $\boldsymbol{P}_{\hat{x}}$, cross-covariance $\boldsymbol{P}_{\hat{y}_k}$, and indirectly Kalman gain $\boldsymbol{K}$ will be one smaller on their state-axis than the state $\underline{\boldsymbol{x}}$.

Second, the addition and subtraction function of an attitude in angle-axis form to the state's quaternion attitude are examined. This occurs when an attitude, related to a covariance P or Kalman gain $\boldsymbol{K}$, in angle-axis form, is added or subtracted from the state's quaternion attitude. More precisely, in Equations 3.17 and 3.27 angle-axis attitudes are added to the state's quaternion attitude. An addition operator $\oplus$ is defined to signify this change in behavior in the UKF, The angle-axis attitude $\underline{\boldsymbol{\omega}}_{\square_k} \Delta t$ is first transformed to a unit-quaternion, using Equation 7.27, before being 'added' to the quaternion attitude by quaternion multiplication $\underset{L}{\otimes}$, i.e.,

$$\underline{\boldsymbol{e}} \oplus \underline{\boldsymbol{\omega}}_{\square_k} \Delta t \qquad = \mathrm{R2Q}\left(\underline{\boldsymbol{\omega}}_{\square_k} \Delta t\right) \underset{R}{\otimes} \underline{\boldsymbol{e}} \quad = \underline{\boldsymbol{e}} \underset{L}{\otimes} \mathrm{R2Q}\left(\underline{\boldsymbol{\omega}}_{\square_k} \Delta t\right) \tag{7.52}$$

And similar for the subtraction:

$$\underline{\boldsymbol{e}} \oplus \left(-\underline{\boldsymbol{\omega}}_{\square_k} \Delta t\right) = \mathrm{R2Q}\left(-\underline{\boldsymbol{\omega}}_{\square_k} \Delta t\right) \underset{R}{\otimes} \underline{\boldsymbol{e}} = \underline{\boldsymbol{e}} \underset{L}{\otimes} \mathrm{R2Q}\left(-\underline{\boldsymbol{\omega}}_{\square_k} \Delta t\right) \tag{7.53}$$

Following, the addition and subtraction of the scaled square-root of the previous augmented state covariance $\boldsymbol{P}^{a}_{\hat{x}\ k-1}$ to the previous estimated state $\underline{\hat{\boldsymbol{x}}}^{a}_{k-1}$ from Equation 3.17 is redefined as:

$$\boldsymbol{\mathcal{X}}^{a}_{k-1} = \left[\underline{\hat{\boldsymbol{x}}}^{a}_{k-1} \quad \underline{\hat{\boldsymbol{x}}}^{a}_{k-1} \oplus \sqrt{(L+\zeta)\,\boldsymbol{P}^{a}_{\hat{x}\ k-1}} \quad \underline{\hat{\boldsymbol{x}}}^{a}_{k-1} \oplus \left(-\sqrt{(L+\zeta)\,\boldsymbol{P}^{a}_{\hat{x}\ k-1}}\right)\right] \tag{7.54}$$

Moreover, the correction of the predicted state $\underline{\hat{\boldsymbol{x}}}^{-}_{k}$ by adding the innovation $\underline{\boldsymbol{\epsilon}}$ scaled by the Kalman gain $\boldsymbol{K}$ is redefined as:

$$\underline{\hat{\boldsymbol{x}}}_k = \underline{\hat{\boldsymbol{x}}}^{-}_{k} \oplus \boldsymbol{K}_k \underline{\boldsymbol{\epsilon}}_k \tag{7.55}$$

Again, it must be noted that for non-attitude scalars $\oplus$ is the same operator as $+$.

7.3.3. Additive Noise

The UKF as presenting in Chapter 3, can be simplified by noting that in all observation models $\mathrm{H}_{\square}$ presented, i.e. Equations 4.44 and 5.18, the observation noise random variables $\underline{\boldsymbol{o}}_{\square}$ are introduced linearly. The linear addition of the observation noise random variables $\underline{\boldsymbol{o}}_{\square}$ causes the inclusion of the observation noise covariance $\boldsymbol{R}$ in the augmented state covariance $\boldsymbol{P}^{a}_{\hat{x}}$ to be unbeneficial. Following, the additive noise model of the UKF, a special and simplified case of the regular UKF, can be used instead for the correction phase. The simplifications are as follows:

- The observation noise variables $\underline{\boldsymbol{o}}$ are removed from the augmented state $\underline{\boldsymbol{x}}^a$ as per Equation 3.10, i.e.,
$$\underline{\boldsymbol{x}}^a = \begin{bmatrix}\underline{\boldsymbol{x}}^T & \underline{\boldsymbol{\nu}}^T\end{bmatrix}^T \tag{7.56}$$

- The covariance of the observation noise variables $\boldsymbol{R}$ is removed from the covariance $\boldsymbol{P}^a_{\hat{x}}$ of the augmented state $\underline{\boldsymbol{x}}^a$ as per Equation 3.11, i.e.,
$$\boldsymbol{P}^a_{\hat{x}} = \begin{bmatrix}\boldsymbol{P}_{\hat{x}} & 0 \\ 0 & \boldsymbol{Q}\end{bmatrix} \tag{7.57}$$

- The reduction of the state's dimension L due to the removal of the observation noise, i.e.,
$$L = L^x + L^v \tag{7.58}$$

- The removal of the observation noise random variables $\underline{\boldsymbol{o}}_\square$ from the observation models H_P and H_{HP} found in Equations 4.44 and 5.18, i.e.,
$$\begin{aligned}\underline{\boldsymbol{y}}_{P_k} &= \mathrm{H}_P\left(\underline{\boldsymbol{x}}_{P_k}\right)\\ \left\{l_{i_k}\right\} &= \left\{\left\|\left(x_{P_k} \quad y_{P_k} \quad z_{P_k}\right) + \left(p_{x_i} \quad p_{y_i} \quad 0\right)\boldsymbol{T}_{PI} - \left(b_{x_i} \quad b_{y_i} \quad 0\right)\right\|\right\} \forall i = 1, \dots, 6\end{aligned} \tag{7.59}$$

  And,
$$\begin{aligned}\underline{\boldsymbol{y}}_{HP_k} &= \mathrm{H}_{HP}\left(\underline{\boldsymbol{x}}_{H_k}, \underline{\boldsymbol{x}}_{P_k}\right)\\ \begin{Bmatrix} x_{HP_k} \\ y_{HP_k} \\ z_{HP_k} \\ \psi_{HP_k} \\ \theta_{HP_k} \\ \phi_{HP_k}\end{Bmatrix} &= \begin{Bmatrix} \boldsymbol{T}_{PI}\begin{Bmatrix} x_{H_k} - x_{P_k} \\ y_{H_k} - y_{P_k} \\ z_{H_k} - z_{P_k}\end{Bmatrix} \\ \text{arctan}\left(\left(e_{0_{HP_k}} e_{x_{HP_k}} + e_{y_{HP_k}} + e_{z_{HP_k}}\right), \left(0.5 - e_{x_{HP_k}} - e_{y_{HP_k}}\right)\right) \\ \text{arcsin}\left(2\left(e_{0_{HP_k}} e_{y_{HP_k}} - e_{x_{HP_k}} e_{z_{HP_k}}\right)\right) \\ \text{arctan}\left(\left(e_{0_{HP_k}} e_{z_{HP_k}} + e_{x_{HP_k}} + e_{y_{HP_k}}\right), \left(0.5 - e_{y_{HP_k}} - e_{z_{HP_k}}\right)\right)\end{Bmatrix}\end{aligned} \tag{7.60}$$

- The direct addition of the covariance of the observation noise $\boldsymbol{R}$ to the computation of the covariance of the estimated observation $\boldsymbol{P}_{\hat{y}_k}$, in Equation 3.24, i.e.,
$$\boldsymbol{P}_{\hat{y}_k} = \sum_{i=0}^{2L} \boldsymbol{w}_i^{(c)} \left[\underline{\boldsymbol{\mathcal{Y}}}^-_{k,i} - \underline{\hat{\boldsymbol{y}}}_k\right]\left[\underline{\boldsymbol{\mathcal{Y}}}^-_{k,i} - \underline{\hat{\boldsymbol{y}}}_k\right]^T + \boldsymbol{R} \tag{7.61}$$

The ability to reduce the augmented state's dimension L leads to a direct reduction in computational cost.

7.4. Analysis

In this section the various configurations, presented in Chapter 5, are simulated, the states estimated using the data generated, explained in Section 7.2, and the performance of the proposed Unscented Kalman Filter (UKF) algorithm analyzed.

For this reason, the performance metrics of the algorithm are presented and discussed in Section 7.4.1. With these performance metrics in mind, three base runs are performed, comparing the performance of each configuration against Configuration 0 in Section 7.4.2. Following these three base runs, a sensitivity analysis is performed per configuration, in Section 7.5, whereby each run a different set of parameters are altered, including the noise values of the different sensors, motion profile, UKF specific parameters, and computational frequency. The resulting performance metrics are in turn compared against their base runs and conclusions are drawn.

After which, the base runs are repeated each twice with a delay present of 50 ms on the output of the SIMONA Research Simulator (SRS) linear encoders sensors in Section 7.5.4. The first of the two runs with a delay present will be shown as is, meaning no alteration will be made to the algorithm. The second of the two runs with a delay present will be performed with the Smoothed Delayed Unscented Kalman Filter (SDUKF) algorithm, presented in Section 3.3.2. Following both runs will be plotted and the discussed performance metrics applied.

7.4.1. Performance Metrics

In this section the performance metrics are described that help to ascertain the quality of the overall system in tandem with analyzing the plotted data. The performance of the algorithm on the systems is defined by two main metrics sets, pose errors and robustness.

The error that defines the user experience of a Virtual Reality (VR) system is the Head-Mounted Display (HMD) vehicle-fixed pose error, consisting out of error on the vehicle-fixed position $\varepsilon_{\underline{\boldsymbol{c}}_{HP}}$ as well as the error on the vehicle-fixed attitude $\varepsilon_{\underline{\boldsymbol{e}}_{HP}}$. However, this error can only be ascertained in simulation as in real life the correct pose would be unknown.

In contrast, the observational innovation $\underline{\boldsymbol{\epsilon}}$ of a Kalman Filter (KF), defined in Equation 3.27, can always be analyzed. The innovation $\underline{\boldsymbol{\epsilon}}$ is analyzed by plotting it together with its covariance $\boldsymbol{P}_{\hat{y}_k}$ to get a view onto the condition of the KF. This will be done in various plots in the next sections. In order to analyze the innovation $\underline{\boldsymbol{\epsilon}}$ at a quick glance, the Normalized Estimated Error Squared (NEES) [6] is computed for both observation sensors, i.e. linear encoders $\nu_{\boldsymbol{\epsilon}_{l_i}}$ and TrackIR $\nu_{\boldsymbol{\epsilon}_{HP}}$.

Also, the NEES is computed of both the HMD $\left(\nu_{\boldsymbol{c}_H}, \nu_{\boldsymbol{e}_H}\right)$ and SRS pose $\left(\nu_{\boldsymbol{c}_P}, \nu_{\boldsymbol{e}_P}\right)$ by comparing the pose error with respect to (w.r.t.) the inertial reference frame $\mathbb{E}_I$, the same reference frame as the generated motion in Section 7.2.3, against its relevant part of the state covariance $\boldsymbol{P}_{\hat{x}}$. This is done in order to relate and analyze the state based NEES $\left(\nu_{\boldsymbol{c}_\square}, \nu_{\boldsymbol{e}_\square}\right)$ and plots to the innovation based NEES $\left(\nu_{\boldsymbol{\epsilon}_{l_i}}, \nu_{\boldsymbol{\epsilon}_{HP}}\right)$.

Absolute Pose Errors

The HMD vehicle-fixed absolute pose errors, $\varepsilon_{\boldsymbol{c}_{HP_k}}$ and $\varepsilon_{\boldsymbol{e}_{HP_k}}$, are computed at each iteration of the algorithm. The actual $\underline{\boldsymbol{e}}_{HP_k}$ and estimated HMD vehicle-fixed quaternion attitude $\hat{\underline{\boldsymbol{e}}}_{HP_k}$ are computed using Equation 5.13. Meanwhile, the actual $\underline{\boldsymbol{c}}_{HP_k}$ and estimated HMD vehicle-fixed position $\hat{\underline{\boldsymbol{c}}}_{HP_k}$ are derived using the upper part, i.e. the positional part, of Equation 5.17. Using said equations, both the actual and estimated HMD vehicle-fixed pose can be computed using the generated actual state $\underline{\boldsymbol{x}}_k$ and the recorded estimated state $\hat{\underline{\boldsymbol{x}}}_k$ respectively.

First, the absolute attitude vehicle-fixed error $\varepsilon_{\boldsymbol{e}_{HP}}$ is defined as the smallest angle between the estimated $\hat{\underline{\boldsymbol{e}}}_{HP}$ and actual attitude $\underline{\boldsymbol{e}}_{HP}$ of the HMD in the upper motion platform reference frame $\mathbb{E}_P$. The smallest angle [71] between two quaternions is computed as per time-step k as:

$$\varepsilon_{e_{HP_k}} = \text{arccos}\left(\frac{\hat{\underline{\boldsymbol{e}}}_{HP_k} \cdot \underline{\boldsymbol{e}}_{HP_k}}{\|\hat{\underline{\boldsymbol{e}}}_{HP_k}\|\|\underline{\boldsymbol{e}}_{HP_k}\|}\right)$$

Assuming the norm of both quaternions are equal to one, i.e. $\|\underline{\hat{\boldsymbol{e}}}_{HP_k}\| = \|\underline{\boldsymbol{e}}_{HP_k}\| = 1$, the equation can be simplified to:

$$= \text{arccos}\left(e_{0_{HP_k}}\hat{e}_{0_{HP_k}} + e_{x_{HP_k}}\hat{e}_{x_{HP_k}} + e_{y_{HP_k}}\hat{e}_{y_{HP_k}} + e_{z_{HP_k}}\hat{e}_{z_{HP_k}}\right) \tag{7.62}$$

Second, the absolute positional error is defined as the Euclidean distance of the difference between the actual $\underline{\boldsymbol{c}}_{HP_k}$ and estimated $\underline{\hat{\boldsymbol{c}}}_{HP_k}$ vehicle-fixed position, i.e.,

$$\varepsilon_{c_{HP_k}} = \sqrt{\left(x_{HP_k} - \hat{x}_{HP_k}\right)^2 + \left(y_{HP_k} - \hat{y}_{HP_k}\right)^2 + \left(z_{HP_k} - \hat{z}_{HP_k}\right)^2} \tag{7.63}$$

The absolute SRS pose error in the inertial frame $\mathbb{E}_I$ is also computed as a secondary performance indicator of the SRS pose estimation itself. The total attitude error $\varepsilon_{e_{P_k}}$ of the SRS in the inertial frame $\mathbb{E}_I$ can be computed using Equation 7.62 using the actual $\underline{\boldsymbol{e}}_{P_k}$ and estimated $\underline{\hat{\boldsymbol{e}}}_{P_k}$ attitude of the SRS. The total positional error $\varepsilon_{c_{P_k}}$ of the SRS in the inertial frame $\mathbb{E}_I$ is computed using the actual $\underline{\boldsymbol{c}}_P$ and estimated $\underline{\hat{\boldsymbol{c}}}_{P_k}$ attitude of the SRS in Equation 7.63.

Normalized Estimated Error Squared

The NEES metric, when applied to the pose error ε or innovation $\underline{\boldsymbol{\epsilon}}$, can be an indication for a 'properly' behaving and tuned UKF [6], meaning the noise is white and within expected covariance.

The NEES ν is computed by averaging the result of division between the variance on the state estimation error ε or innovation $\underline{\boldsymbol{\epsilon}}$ by the KF expected covariance per iteration. If the measured/actual variance is equal over time to the expected covariance, this should equate to one per Degree-of-Freedom (DoF).

The NEES $\nu_{\boldsymbol{\epsilon}}$ of the innovation $\underline{\boldsymbol{\epsilon}}$ can be computed as:

$$\nu_{\boldsymbol{\epsilon}} = \underline{\boldsymbol{\epsilon}}_k^T \boldsymbol{P}_{\hat{y}_k}^{-1} \underline{\boldsymbol{\epsilon}}_k \tag{7.64}$$

Meanwhile, the NEES ν_{ε} of the pose of the state is defined as:

$$\nu_{\varepsilon} = \left[\underline{\boldsymbol{x}}_k \ominus \underline{\hat{\boldsymbol{x}}}_k\right]^T \boldsymbol{P}_{\hat{x}_k}^{-1} \left[\underline{\boldsymbol{x}}_k \ominus \underline{\hat{\boldsymbol{x}}}_k\right] \tag{7.65}$$

Where, $\ominus$ was defined in Equation 7.49, turning quaternion differences into angle-axis attitudes. In this form, the attitude difference can be directly compared to the state covariance $\boldsymbol{P}_{\hat{x}}$, also in angle-axis attitude formulation.

Equations 7.64 and 7.65 result in a scalar, i.e. defining the dimension of $\underline{\boldsymbol{\epsilon}}$ or $\underline{\boldsymbol{x}}$ as $(n \times 1)$, the result of the NEES computation is $(1 \times n)(n \times n)(n \times 1) = (1 \times 1)$. The relationship between random variables and their covariance is supposed to be chi-squared distributed with n degrees of freedom [6]. Or in other words the resulting average NEES $\tilde{\nu}$ should be equal or smaller than the degrees of freedom n. Meaning, 3 for the position and attitude, and 6 for both observational sensors. This will be highlighted when published in the tables with: green to indicate smaller or equal values; orange to indicate slightly larger values (but still satisfactory results in most-cases); red to indicate NEES to large and not satisfactory.

The reason for only computing the NEES of the pose and not the velocity or similar is the method of motion generation outlined in Section 7.2.4. The estimated velocity in particularly will be in an 'awkward' phase. The acceleration, used as input in certain models, is generated based on the second derivate of a cumulative sinusoidal motion that is not equal to the acceleration necessary to the velocity and subsequently position needed. Due to the fact that the both systems observe the pose, the UKF will try to correct the estimated velocity based on the given pose. This correction will be based on the required velocity according to the process model. Hence, an omnipresent error will exist due to the conflict between the result of the integrated acceleration, due to the process model and the velocity required for the positional difference between iterations, due to the observation model. The influence of each model, process or observation, is determined by the ratio between the process $\boldsymbol{Q}$ and observation noise covariance $\boldsymbol{R}$. Following, the velocity generated from sampling the first derivate of a cumulative sinusoidal motion will be close to the estimated velocity but not exact. The resulting error will show the underlying motion depending on the scale of the difference between the integrated acceleration and velocity required.

7.4.2. Base Runs

In this section, the three configurations, presented in Chapter 5, are simulated and subsequently analyzed. First, in Section 7.4.2.0 the base configuration, Configuration 0, with only visual tracking of the HMD is simulated and analyzed. Following, the Base run of Configurations 1 is analyzed in Section 7.4.2.1. This configuration uses the Micro-Electro-Mechanical Systems (MEMS) based Inertial Measurement Unit (IMU) in the HMD, while also visual tracking it. Moreover, the linear encoders are also used to track the state of the SRS, which is necessary to relate the HMD's IMU to the vehicle-fixed frame. Next, the Base run of Configuration 2 is discussed in Section 7.4.2.2. This configuration, in comparison to Configurations 1, enables the MEMS based IMU of the SRS to aid the estimation. Last, the Base run of Configuration 3 is analyzed in Section 7.4.2.3. This configuration swaps out the SRS's IMU, enabled in Configuration 2, for an implementation that uses the Motion Command Signals to the SRS itself.

Before each simulation of each configuration the initial estimated state $\underline{\hat{\boldsymbol{x}}}_0$, initial estimated state covariance $\boldsymbol{P}_{\hat{x}_0}$, process noise covariance $\boldsymbol{Q}$, observation noise covariance $\boldsymbol{R}$, the Scaled Unscented Transformation (SUT) parameters, and the computational frequency are defined.

All runs will be computed at 600 Hz unless specified differently as mentioned in Section 7.2.4. Also stated and analyzed in Section 7.2.4, the publishing frequency of the SRS subsystem is set to 100 Hz. The initial implications were already discussed in Section 7.2.4. In this section, the consequences in terms of performance and condition of the algorithm are analyzed.

After each simulation the plots of the relevant states are plotted against time, performance metrics and NEES are computed of the last $25\,\%$ of the run. Taking only the last $25\,\%$ of the run is done to give ample time for the algorithm to converge and be able to compare run to run performance independent of initial values. The convergence itself will be analyzed visually from the plots.

The error of the estimated states or innovation of the observation will be plotted as the blue line —. Moreover, the error or innovation will be plotted against its covariance. The covariance, representing the certainty on said error or innovation, is plotted as an red dotted - - upper and lower bound. The area within both bounds representing the $95\,\%$-certainty range. The values for the upper and lower bound given by two times the square-root of the relevant part of the relevant covariance P, i.e. $\pm 2 \times \sqrt{P}$.

Configuration 0: Only Visual Tracking of the HMD

In this section, the baseline will be analyzed to which all subsequent configurations are compared against. While, the TrackIR 5 published its measurements at 120 Hz, this analysis is performed at 600 Hz, the baseline for all other runs. This means that one out of five ticks (of the 600 Hz) the measured pose is compared to the actual pose with the same time stamp. However, the other four out of five ticks the same measured pose will be compared to a slightly newer current actual pose.

The actual and measured HMD vehicle-fixed pose are computed using the observation model presented in Equations 7.59 and 7.43 respectively. The resulting pose error, both positional and attitude, are computed via Equations 7.63 and 7.62 and the results compiled in Table 7.7. From this point on this run will be marked as 'Run 0'.

Table 7.7: Performance overview of the base run of configuration 0 using the error of the vehicle-fixed position and attitude of the HMD.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	Standard Deviation (SD)	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}

Configuration 1: SRS's Linear Encoders + HMD's Visual Tracking and IMU

In Equation 7.66, the expected initial state of both the SRS $\underline{\hat{\boldsymbol{x}}}_{P_0}$ and HMD $\underline{\hat{\boldsymbol{x}}}_{H_0}$ are presented based on Equation 3.14. Both expected initial state, $\underline{\hat{\boldsymbol{x}}}_{P_0}$ and $\underline{\hat{\boldsymbol{x}}}_{H_0}$, were given a small random offset to their expected estimated starting position $\left(\underline{\hat{\boldsymbol{c}}}_{P_0}, \underline{\hat{\boldsymbol{c}}}_{H_0}\right)$, and attitude $\left(\underline{\hat{\boldsymbol{e}}}_{P_0}, \underline{\hat{\boldsymbol{e}}}_{H_0}\right)$. Resulting in an initial total error on the estimated attitude of $1.71°$ and $2.22°$ in case of the SRS and HMD respectively. Moreover, the total Euclidean distance error on the estimated position is 0.34 m and 0.14 m for the SRS and HMD respectively. The assumption here is that the entities start from a standstill, but the expected position and attitude estimates are wrong.

$$
\begin{aligned}
\hat{\underline{\boldsymbol{x}}}_0 &= E\left|\underline{\boldsymbol{x}}_0\right| = E\left|\left(\underline{\boldsymbol{x}}_{P_0}^T \quad \underline{\boldsymbol{x}}_{H_0}^T\right)^T\right| \\
&= E\left|\left(\underline{\boldsymbol{c}}_{P_0}^T \quad \dot{\underline{\boldsymbol{c}}}_{P_0}^T \quad \underline{\boldsymbol{e}}_{P_0}^T \quad \underline{\boldsymbol{\omega}}_{P_0}^T\right)^T \left(\underline{\boldsymbol{c}}_{H_0}^T \quad \dot{\underline{\boldsymbol{c}}}_{H_0}^T \quad \underline{\boldsymbol{e}}_{H_0}^T \quad \underline{\boldsymbol{\lambda}}_{H_0}^T\right)^T\right| \\
\begin{Bmatrix} \hat{\underline{\boldsymbol{x}}}_{P_0} \\ \hat{\underline{\boldsymbol{x}}}_{H_0} \end{Bmatrix} &= \begin{Bmatrix} \hat{\underline{\boldsymbol{c}}}_{P_0} \\ \hat{\dot{\underline{\boldsymbol{c}}}}_{P_0} \\ \hat{\underline{\boldsymbol{e}}}_{P_0} \\ \hat{\underline{\boldsymbol{\omega}}}_{P_0} \\ \hat{\underline{\boldsymbol{c}}}_{H_0} \\ \hat{\dot{\underline{\boldsymbol{c}}}}_{H_0} \\ \hat{\underline{\boldsymbol{e}}}_{H_0} \\ \hat{\underline{\boldsymbol{\lambda}}}_{H_0} \end{Bmatrix} = \begin{Bmatrix} \left(0.1173657 \quad -0.19418465 \quad -2.13843659\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0.99955551 \quad 0.01350639 \quad 0.00515045 \quad -0.02607374\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0.12892825 \quad -0.04333829 \quad -3.01418109\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0.99924809 \quad 0.00405833 \quad 0.02351122 \quad 0.03056158\right)^T \\ \left(0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right)^T \end{Bmatrix}
\end{aligned}
\tag{7.66}
$$

Following, the expected initial state covariance $\boldsymbol{P}_{\hat{x}_0}$ should reflect the certainty on the expected initial state, see Equation 3.15. The expected initial state covariance $\boldsymbol{P}_{\hat{x}_0}$ is presented in Equation 7.67. For presentation the diagonal of the covariance is split into two according the the subsystem they belong. The upper values are from the SRS and the lower are from the HMD system.

$$
\begin{aligned}
\boldsymbol{P}_{\hat{x}_0} &= E\left|\left[\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0\right]\left[\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0^T\right]\right| \\
&= \operatorname{diag}\left(\begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}^T \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}^T \begin{Bmatrix} (\mathrm{rad}(25))^2 \\ (\mathrm{rad}(25))^2 \\ (\mathrm{rad}(25))^2 \end{Bmatrix}^T \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}^T \right. \\
&\qquad \left. \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}^T \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}^T \begin{Bmatrix} (\mathrm{rad}(30))^2 \\ (\mathrm{rad}(30))^2 \\ (\mathrm{rad}(30))^2 \end{Bmatrix}^T \begin{Bmatrix} (10^{-3})^2 \\ (10^{-3})^2 \\ (10^{-3})^2 \\ (10^{-4})^2 \\ (10^{-4})^2 \\ (10^{-4})^2 \end{Bmatrix}^T \right)
\end{aligned}
\tag{7.67}
$$

Here, the initial state covariance was set with relative large values for the systems at hand. The relative large initial state covariance would indicate an uncertain start. However, the covariances of the IMU biases of the HMD, were set small ($(10^{-3})^2$, $(10^{-4})^2$) relative to the other variances. This will cause the UKF to correct the other estimated states more based on the observations in the initial iterations. Later, in Section 7.5, this will be further investigated by setting the initial bias covariance larger and smaller.

In Equation 7.68 the expected covariance of the process noise $\boldsymbol{Q}$ present in Configuration 1 is presented. The acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ noise parameters of the SRS, which are there to provide a means for the UKF to adjust the estimated velocity $\hat{\dot{\underline{c}}}_P$ and angular rate $\hat{\underline{\omega}}_P$ of the SRS as in this configuration the SRS has no IMU, were chosen after some trial and error and should be tuned according to the expected motion. Later, in Section 7.5, a sensitivity analysis is performed on the acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{\boldsymbol{c}}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\boldsymbol{\omega}}_P}$ noise parameters of the SRS.

$$\begin{aligned}\boldsymbol{Q} &= \operatorname{diag}\left(\underline{\boldsymbol{\sigma}}^2_{\ddot{\boldsymbol{c}}_P} \quad \underline{\boldsymbol{\sigma}}^2_{\dot{\boldsymbol{\omega}}_P} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{f}_H} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\omega}_H} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{f}_{\mu_P}} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\omega}_{\mu_P}}\right) \\ &= \operatorname{diag}\left(\left\{\begin{matrix}1\\1\\1\end{matrix}\right\}^T \left\{\begin{matrix}1\\1\\1\end{matrix}\right\}^T \right. \\ &\qquad \left. \left\{\begin{matrix}0.030^2\\0.029^2\\0.047^2\end{matrix}\right\}^T \cdot 1.1^2 \left\{\begin{matrix}0.0033^2\\0.0036^2\\0.0038^2\end{matrix}\right\}^T \cdot 1.1^2 \left\{\begin{matrix}0.015^2\\0.064^2\\0.048^2\end{matrix}\right\}^T \cdot 1.1^2 \left\{\begin{matrix}0.00018^2\\0.00034^2\\0.00045^2\end{matrix}\right\}^T \cdot 1.1^2 \right)\end{aligned} \tag{7.68}$$

The $10\,\%$ scaling of the expected accelerometer $\underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_H}$ and gyroscope SD $\underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_H}$ are based on the analysis in Section 7.2.4. Furthermore, to create a safety margin from measurement to measurement variance the expected variances was set $10\,\%$ higher than measured in Chapter 6. $10\,\%$ was also added to the expected accelerometer $\underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_{\mu_P}}$ and gyroscope biases SD $\underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_{\mu_P}}$.

The expected covariance of the observation noise $\boldsymbol{R}$ is shown in Equation 7.69. The observation noise covariance $\boldsymbol{R}$ is constructed from the noise parameters gathered in Chapter 6. Here, they are also scaled by $10\,\%$.

$$\begin{aligned}\boldsymbol{R} &= \operatorname{diag}\left(\left\{\begin{matrix}\sigma^2_{x_H}\\ \sigma^2_{y_H}\\ \sigma^2_{z_H}\\ \sigma^2_{\psi_H}\\ \sigma^2_{\theta_H}\\ \sigma^2_{\phi_H}\end{matrix}\right\}^T \left\{\begin{matrix}\sigma^2_{l_1}\\ \vdots \\ \sigma^2_{l_6}\end{matrix}\right\}^T\right) \\ &= \operatorname{diag}\left(\left\{\begin{matrix}\left(9.19\times10^{-5}\right)^2\\ \left(3.04\times10^{-4}\right)^2\\ \left(4.94\times10^{-4}\right)^2\\ \left(1.94\times10^{-3}\right)^2\\ \left(1.30\times10^{-3}\right)^2\\ \left(2.47\times10^{-3}\right)^2\end{matrix}\right\}^T \left\{\begin{matrix}\left(5\times10^{-6}\right)^2\\ \vdots\\ \left(5\times10^{-6}\right)^2\end{matrix}\right\}^T\right)\cdot 1.1^2\end{aligned} \tag{7.69}$$

The Scaled Unscented Transformation parameters are defined as:

$$\alpha = 0.01 \quad \beta = 2 \quad \kappa = 0 \tag{7.70}$$

These SUT parameters, where found to be proper default values in [41, 43, 31].

In Tables 7.8, 7.10, and 7.9 the performance metrics are collected of Configuration 1. From this point forward, this run will be indicated as ‘Run 1’. The total accuracy of the HMD vehicle-fixed pose is compared against Configuration 0 in Table 7.8. In this configuration, both the position as well as the attitude improve as expected.

Table 7.8: Performance overview of the base run of Configuration 1 using the error of the vehicle-fixed position and attitude of the HMD.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}
1	1.70×10^{-4}	8.22×10^{-5}	5.05×10^{-4}	6.87×10^{-3}	3.35×10^{-3}	2.01×10^{-2}

Following, the NEES values for Configuration 1 are computed and presented in Table 7.9. As explained in Section 7.4.1 the various NEES are coloured relative to their degrees of freedom.

The linear encoder innovations are slightly larger than expected, this can have two reasons not mutually exclusive: coloured innovation within bounds and/or random innovation out of bounds. Figure 7.16 shows that the innovation is coloured but within bounds.

Table 7.9: Condition overview of the base run of Configuration 1, see Section 7.4.1 for the explanation on the color use.

	NEES					
	SRS			HMD		
Run	Position	Attitude	Lin. Enc. Inno.	Position	Attitude	TrackIR 5 Inno.
1	2.59	2.62	7.43	3.11	2.01	5.00

Below, the position and attitude estimated error, i.e. $\underline{\hat{\boldsymbol{x}}}_k \ominus \underline{\boldsymbol{x}}_k$, are plotted against their relevant part of the state covariance $\boldsymbol{P}_{\hat{x}_k}$ over time t in Figures 7.9 and 7.10. Table 7.9 indicated that the error on the inertial position was slightly large relative to its covariance or coloured. Figure 7.9 shows hints of the underlying motion. This is explained by the method the position and acceleration are generated, see Section 7.2.4. Also both the position and attitude of the HMD converge after approximately 1 s.

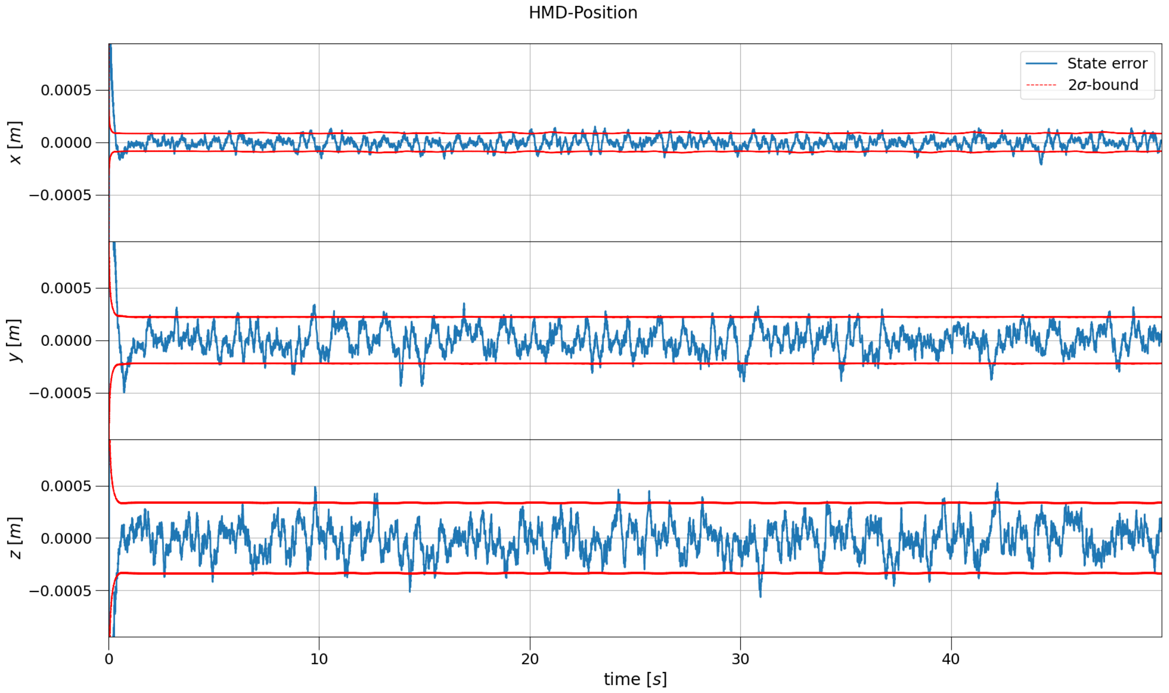


Figure 7.9: Base run Configuration 1, the error of the HMD estimated position per axis of the inertial frame plotted against its covariance over time.

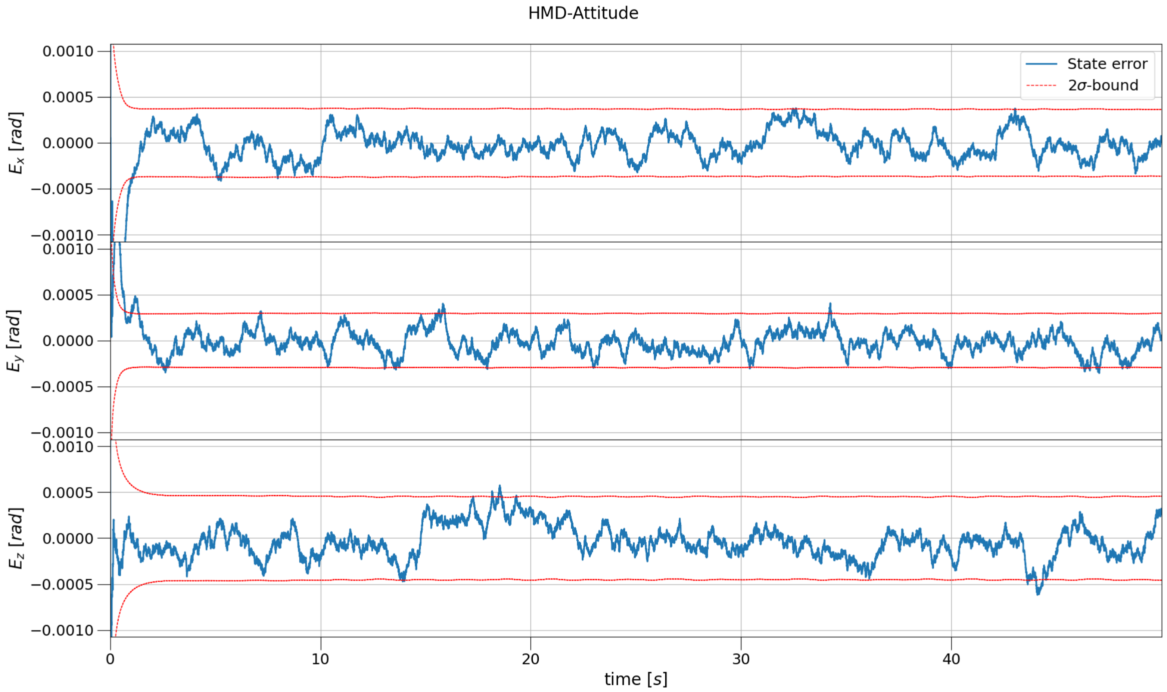


Figure 7.10: Base run Configuration 1, the error of the HMD estimated attitude in angle-axis formulation per axis of the inertial frame plotted against its covariance over time.

Both the accelerometer and gyroscope biases stay well within their sigma-bounds in Figures 7.11 and 7.12. The small initial covariance is visualized in case of the accelerometer. Here, the covariance expands initially.

Moreover, the HMD accelerometer bias $\underline{\lambda}_{\boldsymbol{f}_H}$ follows the underlying motion error. Here, the bias is used by the UKF to compensate for the forward Euler integration error on the motion generation. This also explains why the coloured noise is absent on the bias of the gyroscope $\underline{\lambda}_{\omega_H}$ the attitude as they were computed discretely.

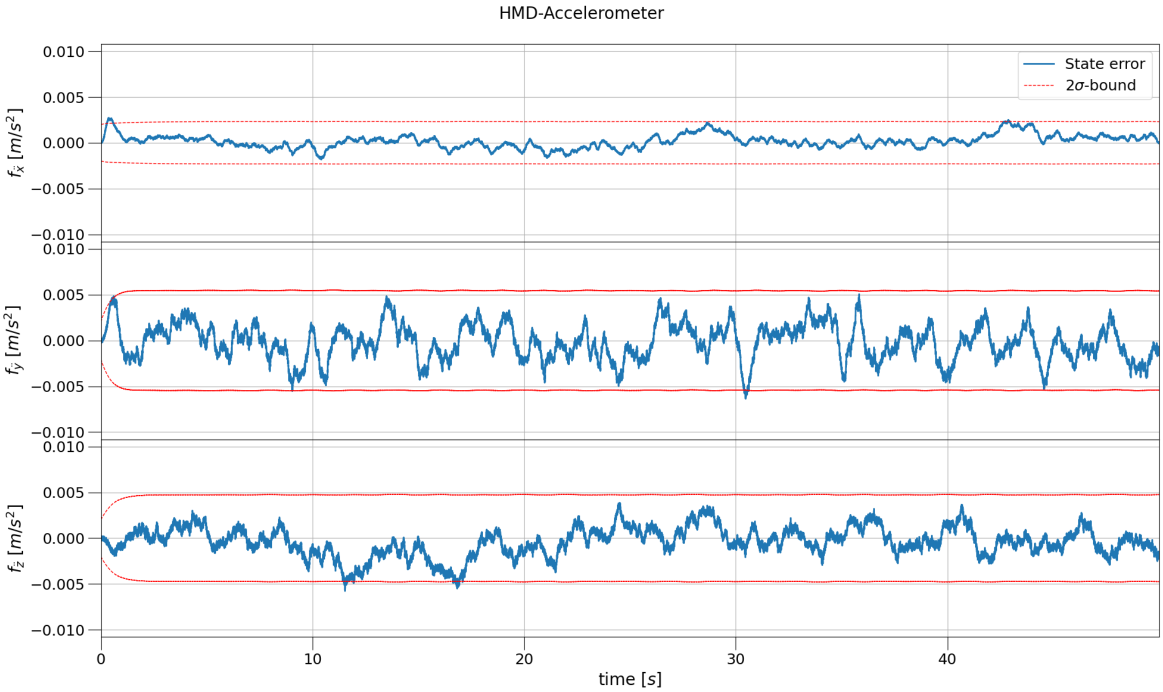


Figure 7.11: Base run Configuration 1, the error of the accelerometer bias of the HMD per axis plotted against its covariance over time.

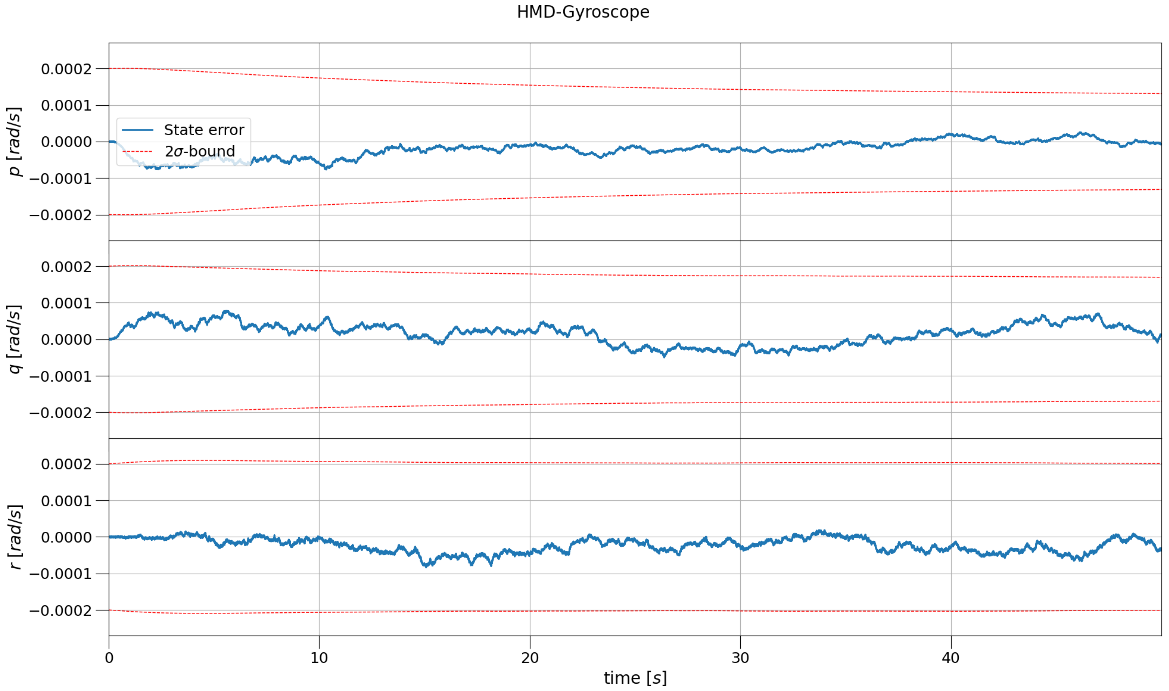


Figure 7.12: Base run Configuration 1, the error of the gyroscope bias of the HMD per axis plotted against its covariance over time.

In Table 7.10 the total pose error of the SRS is collected. At the end of this section, SRS performance in all three configurations is compared to another in Table 7.16.

Table 7.10: Performance overview of the base run of Configuration 1 using the SRS error of the inertial position and attitude.

	SRS					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
1	2.95×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.71×10^{-4}	5.52×10^{-4}	3.58×10^{-3}

The error on the estimated position and attitude of the SRS are random and within sigma-bounds, as hinted by the NEES in Table 7.9. Even though the expected acceleration and angular acceleration SD were set equal over all axes, see Equation 7.68, the covariance on the z-axis of both the position as well as the attitude are smaller than the x and y-axis. For the SRS, the certainty on the z-axis is increased by $100\,\%$ and $50\,\%$ respectively. Here, the reduction in covariance is solely due to the physical shape of the Stewart platform. This also explains the change in covariance over time of the pose due to the change in sensor attitude and thus certainty per axis in the inertial frame.

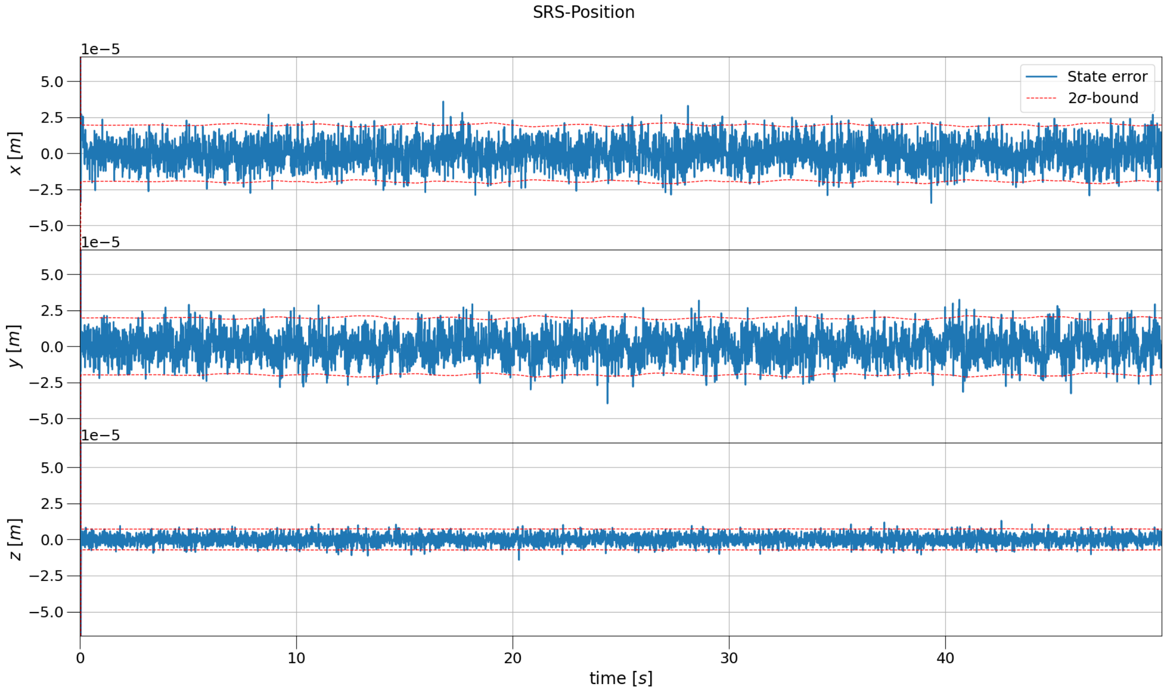


Figure 7.13: Base run Configuration 1, the error of the SRS estimated position per axis of the inertial frame plotted against its covariance over time.

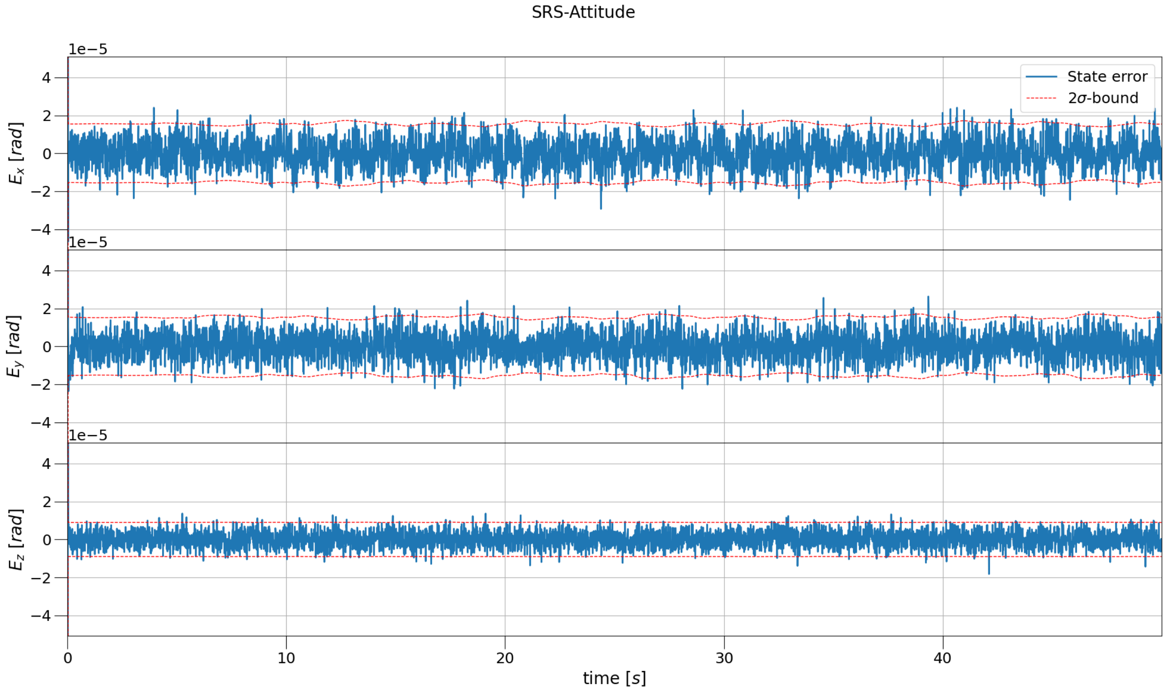


Figure 7.14: Base run Configuration 1, the error of the SRS estimated attitude in angle-axis formulation per axis of the inertial frame plotted against its covariance over time.

The error on the estimated velocity in the inertial frame of the SRS is plotted in Figure 7.15. At first glance the velocity is coloured and not random. Here, the error is computed with respect to the first derivate of the actual position, as stated in Section 7.2. However, the linear encoder observations correct the velocity to the required velocity according to the process model.

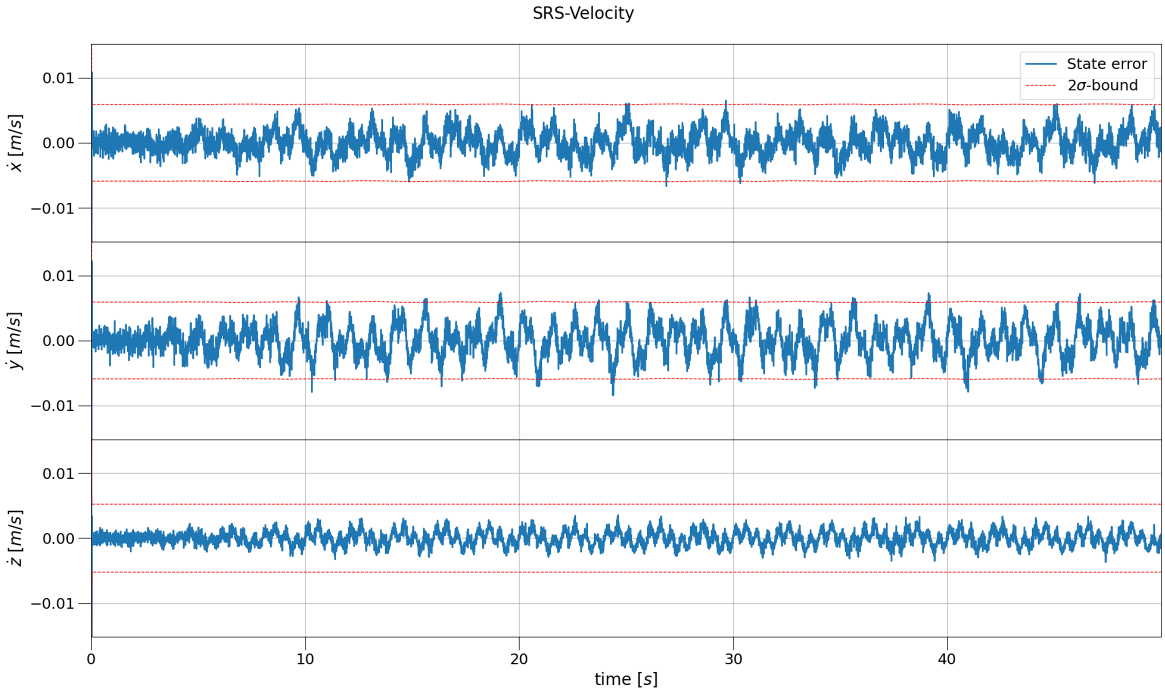


Figure 7.15: Base run Configuration 1, the error of the SRS estimated velocity per axis of the inertial frame plotted against its covariance over time.

Moreover, the innovations of the linear encoders $\underline{\boldsymbol{\epsilon}}_{l_i}$ and the TrackIR 5 $\underline{\boldsymbol{\epsilon}}_{HP}$ are depicted against their part of the observation covariance $\boldsymbol{P}_{\hat{y}_k}$ in Figures 7.17 and 7.16 respectively.

In contrast to SRS pose errors in Figures 7.13 and 7.14, the motion permeates the innovation $\underline{\boldsymbol{\epsilon}}_{l_i}$ on the linear encoders, and therefore not random. This is due to the fact that the SRS has no IMU in this configuration, only a acceleration $\underline{\boldsymbol{\nu}}_{\ddot{\boldsymbol{c}}_P}$ and angular acceleration noise parameter $\underline{\boldsymbol{\nu}}_{\dot{\boldsymbol{\omega}}_P}$ with zero mean and a variance of $\underline{\boldsymbol{\sigma}}^2_{\ddot{\boldsymbol{c}}_P}$ and $\underline{\boldsymbol{\sigma}}^2_{\dot{\boldsymbol{\omega}}_P}$ respectively. The lack of IMU hinders the predictive capabilities of the UKF in case of the SRS in this configuration, but the noise parameters $\underline{\boldsymbol{\nu}}_{\ddot{\boldsymbol{c}}_P}$ and $\underline{\boldsymbol{\nu}}_{\dot{\boldsymbol{\omega}}_P}$ allow the UKF to correct the state, including the velocity and angular rate of the SRS based on the measurements of the linear encoders. The fact that the UKF cannot predict the SRS estimated state effectively but mostly react to the measurements is why the innovations of the linear encoders show the motion in this configuration.

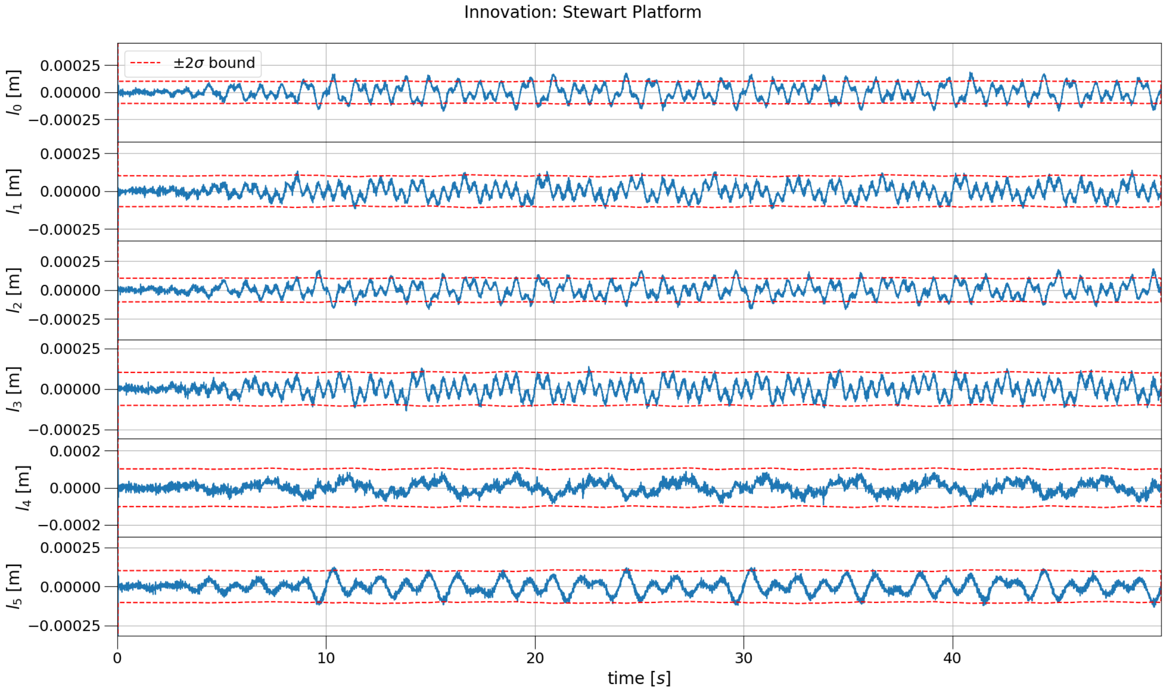


Figure 7.16: Base run Configuration 1, the innovation of linear encoders measuring the length of the hydraulic actuators of the SRS plotted against its covariance over time.

The TrackIR 5 innovation in Figure 7.17 is random and within bounds as the NEES in Table 7.9 would indicate.

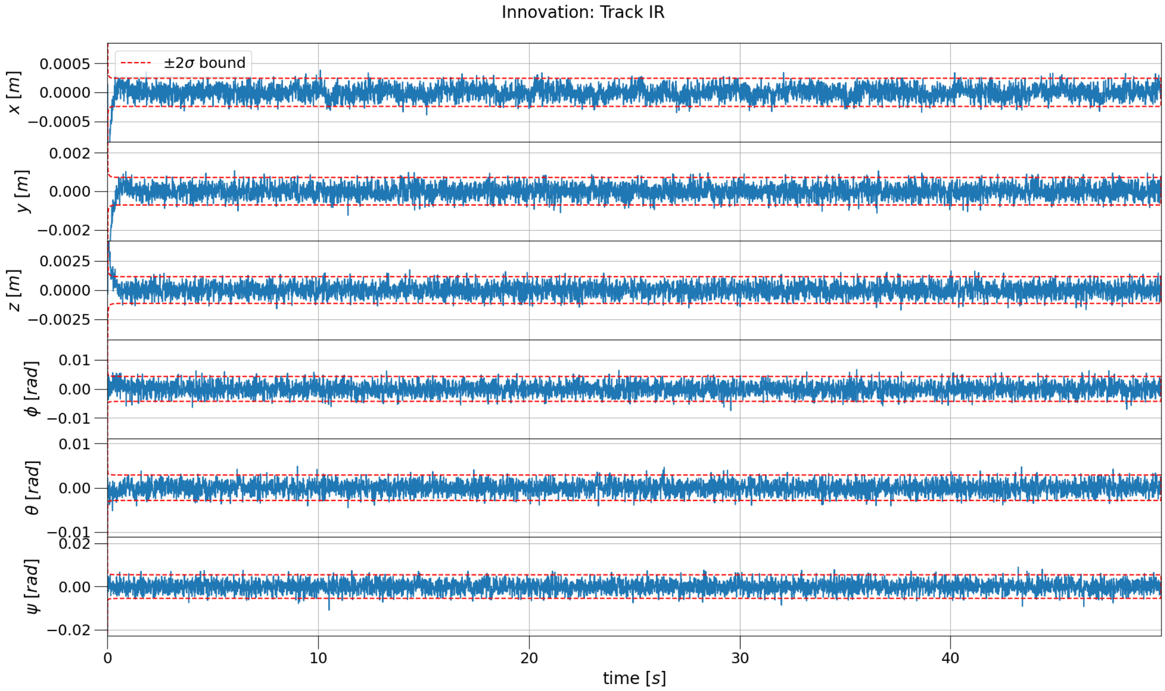


Figure 7.17: Base run Configuration 1, the innovation of Track IR 5 measuring the position and attitude (Euler attitude formulation) of the HMD attached to the SRS plotted against its covariance over time.

Configuration 2: SRS's Linear Encoders and IMU + HMD's Visual Tracking and IMU

The initial estimated state of both the SRS $\hat{\underline{\boldsymbol{x}}}_{P_0}$ and HMD $\hat{\underline{\boldsymbol{x}}}_{H_0}$ are presented in Equation 7.71 based on Equation 3.14. In comparison to the expected initial state $\hat{\underline{\boldsymbol{x}}}_0$ of Configuration 1 in Equation 7.66, only the SRS IMU biases, the values set similar to the ones of the HMD, are new.

$$\begin{aligned}
\hat{\underline{\boldsymbol{x}}}_0 &= E\left|\underline{\boldsymbol{x}}_0\right| = E\left|\begin{pmatrix}\underline{\boldsymbol{x}}_{P_0}^T & \underline{\boldsymbol{x}}_{H_0}^T\end{pmatrix}^T\right| \\
&= E\left|\begin{pmatrix}\underline{\boldsymbol{c}}_{P_0}^T & \dot{\underline{\boldsymbol{c}}}_{P_0}^T & \underline{\boldsymbol{e}}_{P_0}^T & \underline{\boldsymbol{\lambda}}_{P_0}^T\end{pmatrix}^T \begin{pmatrix}\underline{\boldsymbol{c}}_{H_0}^T & \dot{\underline{\boldsymbol{c}}}_{H_0}^T & \underline{\boldsymbol{e}}_{H_0}^T & \underline{\boldsymbol{\lambda}}_{H_0}^T\end{pmatrix}^T\right|
\end{aligned}$$

$$\begin{Bmatrix}\hat{\underline{\boldsymbol{c}}}_{P_0}\\ \hat{\dot{\underline{\boldsymbol{c}}}}_{P_0}\\ \hat{\underline{\boldsymbol{e}}}_{P_0}\\ \hat{\underline{\boldsymbol{\lambda}}}_{P_0}\\ \hat{\underline{\boldsymbol{c}}}_{H_0}\\ \hat{\dot{\underline{\boldsymbol{c}}}}_{H_0}\\ \hat{\underline{\boldsymbol{e}}}_{H_0}\\ \hat{\underline{\boldsymbol{\lambda}}}_{H_0}\end{Bmatrix} = \begin{Bmatrix}\begin{pmatrix}0.1173657 & -0.19418465 & -2.13843659\end{pmatrix}^T\\ \begin{pmatrix}0 & 0 & 0\end{pmatrix}^T\\ \begin{pmatrix}0.99955551 & 0.01350639 & 0.00515045 & -0.02607374\end{pmatrix}^T\\ \begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T\\ \begin{pmatrix}0.12892825 & -0.04333829 & -3.01418109\end{pmatrix}^T\\ \begin{pmatrix}0 & 0 & 0\end{pmatrix}^T\\ \begin{pmatrix}0.99924809 & 0.00405833 & 0.02351122 & 0.03056158\end{pmatrix}^T\\ \begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T\end{Bmatrix} \tag{7.71}$$

The expected initial state covariance $\boldsymbol{P}_{\hat{x}_0}$, presented in Equation 7.72, is just as the expected initial state $\hat{\underline{\boldsymbol{x}}}_0$ similar to the expected initial state covariance $\boldsymbol{P}_{\hat{x}_0}$ presented in Configuration 1 in Equation 7.67.

$$\begin{aligned}
\boldsymbol{P}_{\hat{x}_0} &= E\left|\left[\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0\right]\left[\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0^T\right]\right| \\
&= \operatorname{diag}\left(\begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}(\operatorname{rad}(25))^2\\(\operatorname{rad}(25))^2\\(\operatorname{rad}(25))^2\end{Bmatrix}^T \begin{Bmatrix}(10^{-3})^2\\(10^{-3})^2\\(10^{-3})^2\\(10^{-4})^2\\(10^{-4})^2\\(10^{-4})^2\end{Bmatrix}^T\right. \\
&\qquad \left.\begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}(\operatorname{rad}(30))^2\\(\operatorname{rad}(30))^2\\(\operatorname{rad}(30))^2\end{Bmatrix}^T \begin{Bmatrix}(10^{-3})^2\\(10^{-3})^2\\(10^{-3})^2\\(10^{-4})^2\\(10^{-4})^2\\(10^{-4})^2\end{Bmatrix}^T\right)
\end{aligned} \tag{7.72}$$

The process noise covariance $\boldsymbol{Q}$ is compiled based on the IMU noise parameters and the combined error established in Section 7.2.4. The $10\,\%$ scaling to the expected accelerometer $\underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_{\mu_\square}}$ and gyroscope bias SD $\underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_{\mu_\square}}$ of both the SRS and HMD. Also the case when presenting the process noise covariance $\boldsymbol{Q}$ in Configuration 1 in Equation 7.68.

$$\begin{aligned}
\boldsymbol{Q} &= \operatorname{diag}\begin{pmatrix}\underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_P}^2 & \underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_P}^2 & \underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_{\mu_P}}^2 & \underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_{\mu_P}}^2 & \underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_H}^2 & \underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_H}^2 & \underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_{\mu_H}}^2 & \underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_{\mu_H}}^2\end{pmatrix} \\
&= \operatorname{diag}\left(\begin{Bmatrix}0.0019^2\\0.0021^2\\0.0018^2\end{Bmatrix}^T \cdot 11.0^2 \begin{Bmatrix}0.0024^2\\0.0041^2\\0.0033^2\end{Bmatrix}^T \cdot 1.5^2 \begin{Bmatrix}(10^{-3})^2\\(10^{-3})^2\\(10^{-3})^2\end{Bmatrix}^T \cdot 1.1^2 \begin{Bmatrix}(10^{-4})^2\\(10^{-4})^2\\(10^{-4})^2\end{Bmatrix}^T \cdot 1.1^2\right. \\
&\qquad \left.\begin{Bmatrix}0.030^2\\0.029^2\\0.047^2\end{Bmatrix}^T \cdot 1.1^2 \begin{Bmatrix}0.0033^2\\0.0036^2\\0.0038^2\end{Bmatrix}^T \cdot 1.1^2 \begin{Bmatrix}0.015^2\\0.064^2\\0.048^2\end{Bmatrix}^T \cdot 1.1^2 \begin{Bmatrix}0.00018^2\\0.00034^2\\0.00045^2\end{Bmatrix}^T \cdot 1.1^2\right)
\end{aligned} \tag{7.73}$$

The observation noise covariance $\boldsymbol{R}$ due to the fact that the observation sensors do not change w.r.t. Configuration 1. Also the SUT parameters do not change in this configuration from Configuration 1 in order to make the comparison about the configurations themselves. Later, in Section 7.5 the SUT parameters are altered to investigate their influence. Following, the observation noise covariance $\boldsymbol{R}$ and SUT parameters, α, β, and κ, presented in Equations 7.69 and 7.70 respectively are used.

With the initial state $\underline{\hat{\boldsymbol{x}}}_0$, initial state covariance $\boldsymbol{P}_{\hat{x}_0}$, process noise covariance $\boldsymbol{Q}$, observation noise covariance $\boldsymbol{R}$, and the SUT parameters set of Configuration 2 the performance metrics are presented in Tables 7.11, 7.13, and 7.12. From this point forward, this run will be indicated as 'Run 2'.

The total accuracy of the HMD vehicle-fixed pose is compared against Configurations 0 and 1 in Table 7.11. Here, the data serves to give an understanding of the scale and quantity of the improvement brought forth by the overall configuration and algorithm.

Table 7.11: Performance overview of the base run of Configuration 2 using the error of the vehicle-fixed position and attitude of the HMD.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}
1	1.70×10^{-4}	8.22×10^{-5}	5.05×10^{-4}	6.87×10^{-3}	3.35×10^{-3}	2.01×10^{-2}
2	1.93×10^{-4}	9.56×10^{-5}	5.80×10^{-4}	9.43×10^{-3}	4.78×10^{-3}	2.47×10^{-2}

Compared to the base-line, i.e. Run 0, improvements across the board are observed similar to Run 1. However, contrary to expectations Configuration 1 actually performed better with regards to the HMD's vehicle fixed pose, approximately $15\,\%$ reduction in error on the position and $30\,\%$ on the attitude estimation. As Configuration 2 has an extra sensor, in theory, should perform better compared to Configuration 1. In this case, an optimal, or close to optimal, process noise covariance $\boldsymbol{Q}$ could have been found for Configuration 1 for the motion at hand.

The source of the increase in error is found in the estimation of the SRS state $\underline{\hat{\boldsymbol{x}}}_P$ and its covariance $\boldsymbol{P}_{\hat{x}_P}$ and discussed later in Table 7.13 and corresponding Figures 7.20 and 7.21. Even so, less tuning should be necessary to make Configuration 2 work compared to Configuration 1, as will be evident in Section 7.5 where the sensitivity of each configuration is analyzed.

Following, the NEES values for Configuration 2 are presented in Table 7.12. Here, the impact of the motion generation in combination with the forward Euler integration method is evident in the case of the SRS. The error shows the underlying motion clearly in Figures 7.20 and 7.21 even after scaling the expected SD present on the IMU. As the UKF assumes a Gaussian noise, the underlying motion is still present. The motion also shows in the innovation of the linear encoders and to a lesser extent in the HMD pose error NEES.

Table 7.12: Condition overview of the base run of Configuration 2, see Section 7.4.1 for the explanation on the color use.

	NEES					
	SRS			HMD		
Run	Position	Attitude	Lin. Enc. Inno.	Position	Attitude	TrackIR 5 Inno.
1	2.59	2.62	7.43	3.11	2.01	5.00
2	26.40	15.92	26.59	3.90	3.79	5.18

The resulting errors of the HMD position $\underline{\boldsymbol{c}}_H$ and attitude $\underline{\boldsymbol{e}}_H$ in the inertial reference frame $\mathbb{E}_I$ are presented in Figures 7.18 and 7.19 are very similar to their configuration counterparts in Figures 7.9 and 7.10. Both position and attitude of the HMD are mostly within 2σ-bounds indicated by the red line. Concluding that even though more underlying motion is present in the error the UKF algorithm is still capable in estimating the pose.

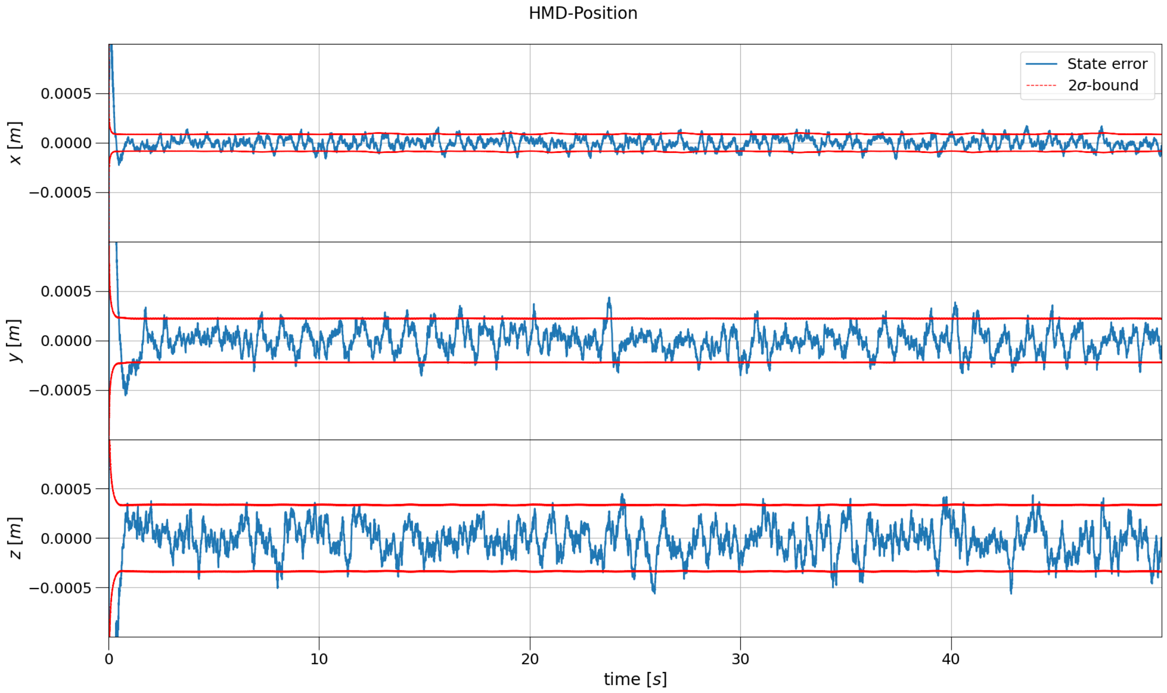


Figure 7.18: Base run Configuration 2, the error of the HMD estimated position per axis of the inertial frame plotted against its covariance over time.

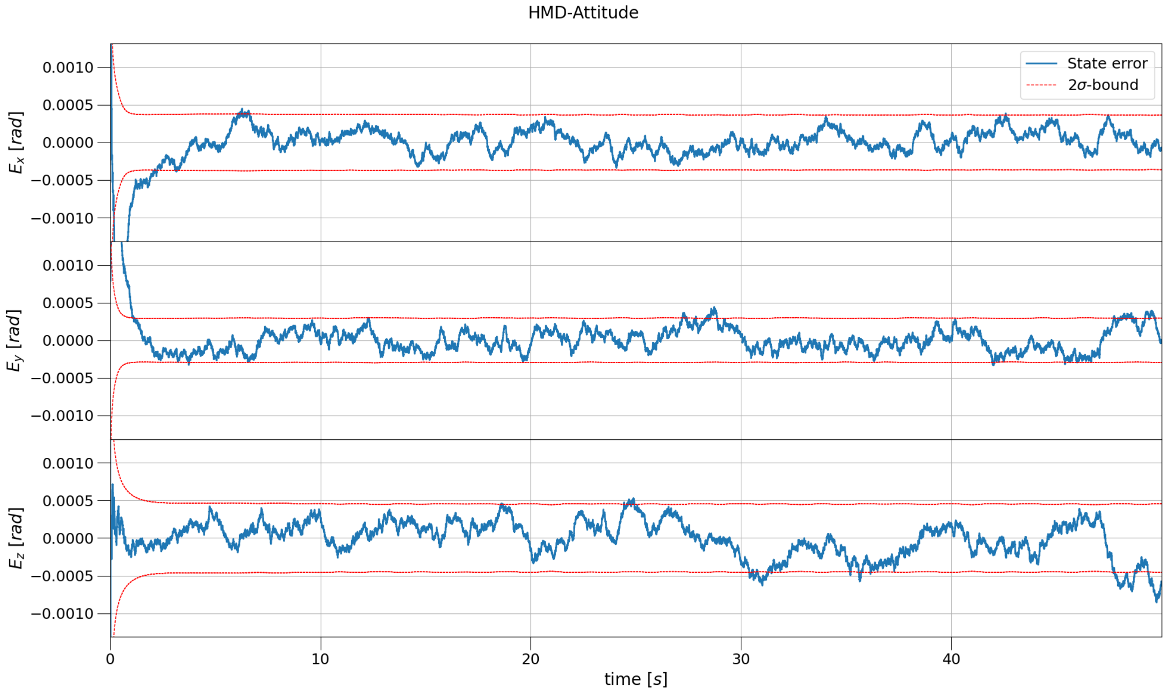


Figure 7.19: Base run Configuration 2, the error of the HMD estimated attitude in angle-axis formulation per axis of the inertial frame plotted against its covariance over time.

For brevity's sake, the HMD accelerometer and gyroscope biases are not shown. They performed similarly in Configuration 1.

The statistics on the total error on the SRS pose is compiled in Table 7.13. Even though the NEES values indicated suboptimal performance in terms of error profile versus covariance, the regression in total pose error is small if any. The variance and max positional error actually improve.

Table 7.13: Performance overview of the base run of Configuration 2 using the SRS error of the inertial position and attitude.

	SRS					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
1	2.95×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.71×10^{-4}	5.52×10^{-4}	3.58×10^{-3}
2	2.98×10^{-5}	1.71×10^{-5}	9.23×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.54×10^{-3}

Following, the SRS position and attitude error in the inertial frame $\mathbb{E}_I$ are shown in Figures 7.20 and 7.21. Here, the increase in z-axis certainty in both the position as well as the attitude is still observable as was the case in Figures 7.13 and 7.14. The certainty increased in this configuration, compared to Configuration 1, due to the smaller overall process noise covariance $\boldsymbol{Q}$ due to the addition of the IMU. However, in Configuration 2 both the positional as well as the attitude error of the SRS show the underlying motion in comparison to Configuration 1. The forward Euler integration induced error is clearly visible due to the 100 Hz sampling rate set on the IMU of the SRS.

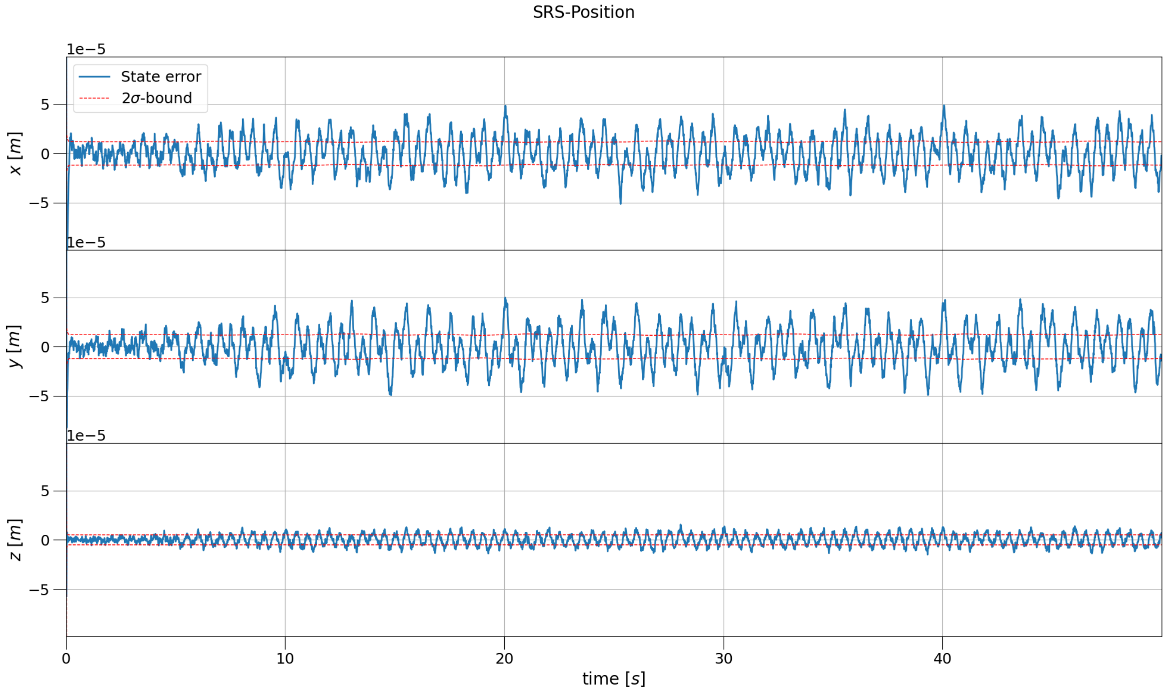


Figure 7.20: Base run Configuration 2, the error of the SRS estimated position per axis of the inertial frame plotted against its covariance over time.

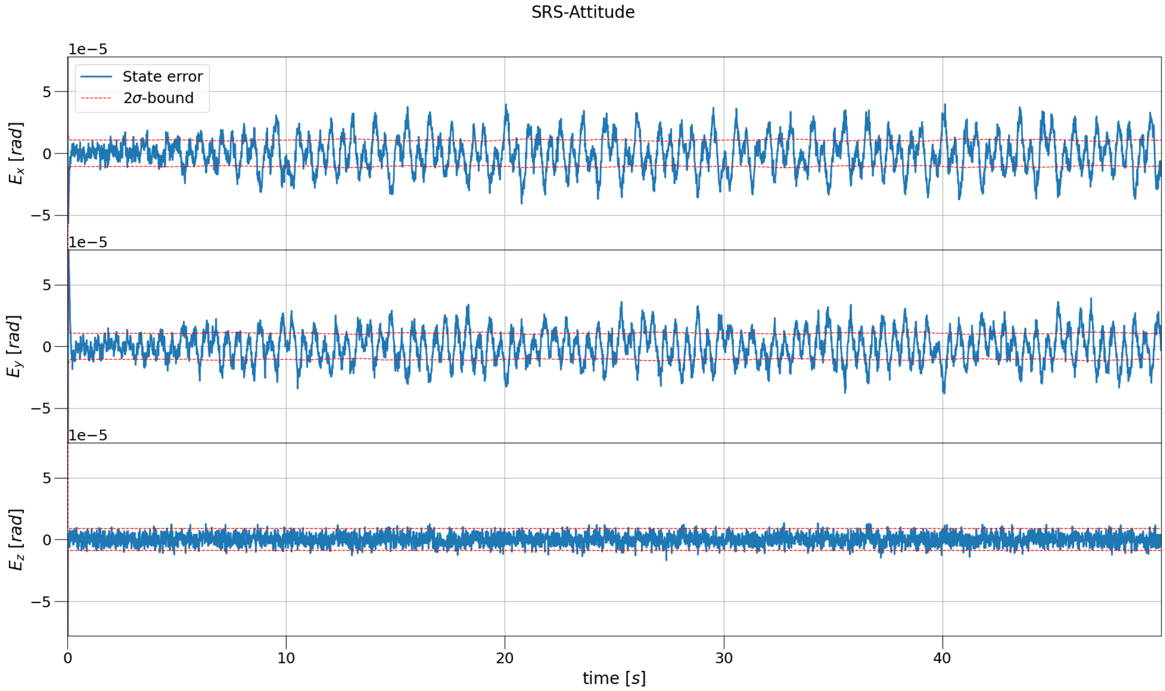


Figure 7.21: Base run Configuration 2, the error of the SRS estimated attitude in angle-axis formulation per axis of the inertial frame plotted against its covariance over time.

The SRS accelerometer $\underline{\lambda}_{\boldsymbol{f}_P}$ and gyroscope biases $\underline{\lambda}_{\boldsymbol{\omega}_H}$, depicted in Figures 7.22 and 7.23 respectively, try to to compensate for the forward Euler induced error.

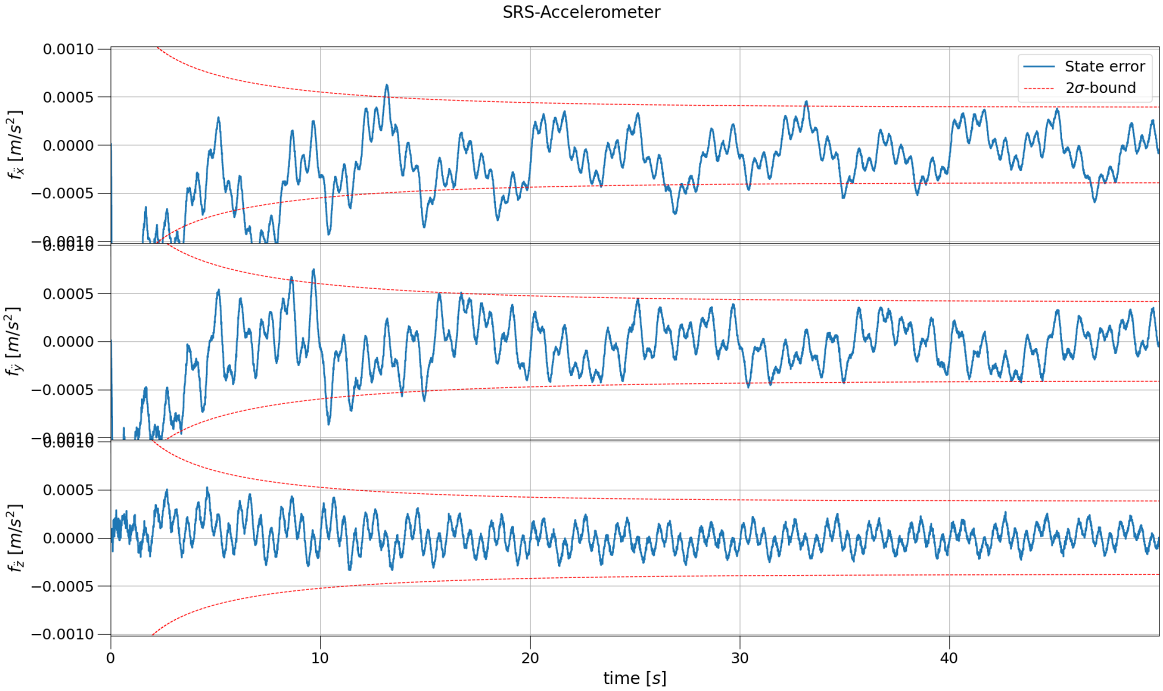


Figure 7.22: Base run Configuration 2, the error of the accelerometer bias of the SRS per axis plotted against its covariance over time.

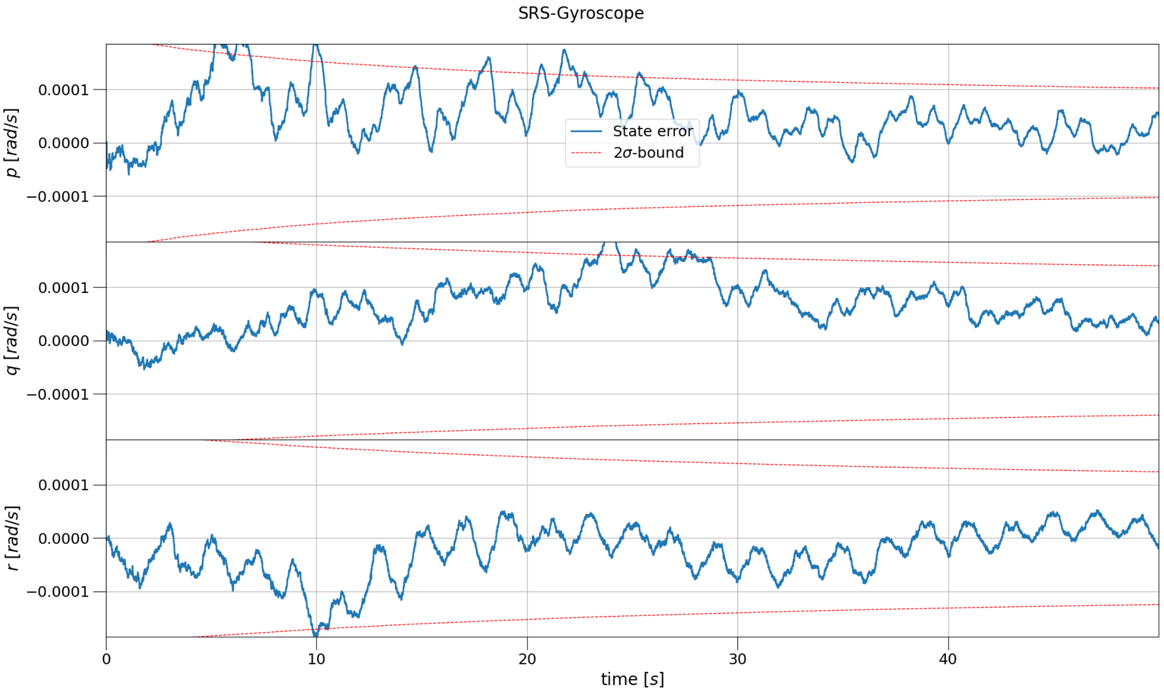


Figure 7.23: Base run Configuration 2, the error of the gyroscope bias of the SRS per axis plotted against its covariance over time.

Moreover, the innovations from TrackIR 5 $\underline{\boldsymbol{\epsilon}}_{HP}$ and linear encoders $\underline{\boldsymbol{\epsilon}}_{l_i}$ are depicted against their own part of the observation covariance $\boldsymbol{P}_{\hat{y}_k}$ in Figures 7.24 and 7.25 respectively. The innovation resulting from the TrackIR 5 observations are within bounds and seemingly Gaussian as the NEES results in Table 7.12 indicate. However, the innovation of the six linear encoders show clear signs of the underlying motion as well as not respecting the expectations set by the 2σ-bounds. This is caused by the discrepancy between the non-Gaussian error introduced in the prediction and the expectations of the correction phase.

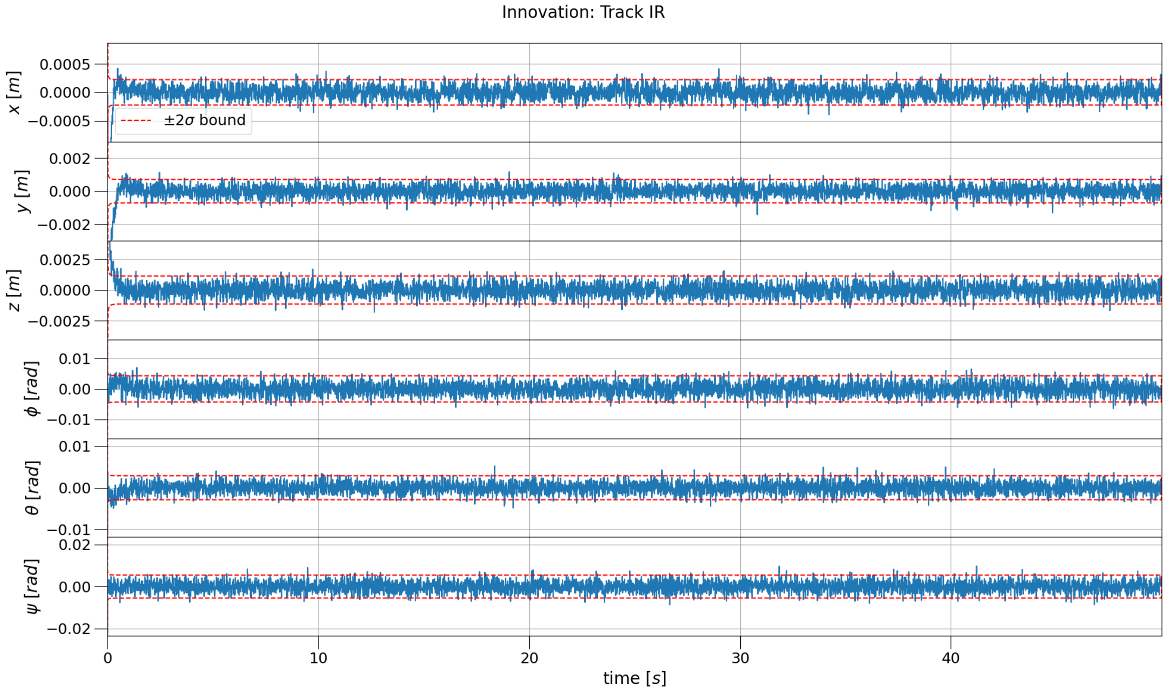


Figure 7.24: Base run Configuration 2, the innovation of Track IR 5 measuring the position and attitude (Euler attitude formulation) of the HMD attached to the SRS plotted against its covariance over time.

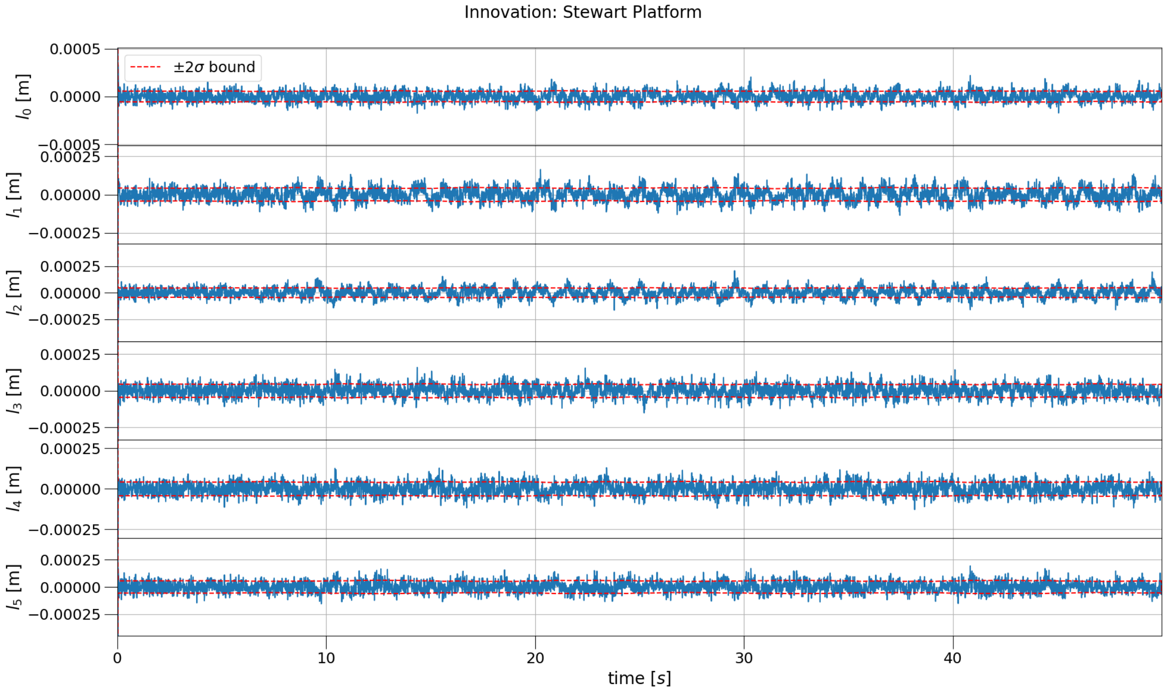


Figure 7.25: Base run Configuration 2, the innovation of linear encoders measuring the length of the hydraulic actuators of the SRS plotted against its covariance over time.

Configuration 3: SRS's Linear Encoders and Setpoints + HMD's Visual Tracking and IMU

The setpoints for Configuration 3 were created at $100\,$Hz but the prediction was performed at $600\,$Hz. Similar to the IMU data generation for the SRS.

Again, the initial estimated state of both the SRS $\hat{\underline{\boldsymbol{x}}}_{P_0}$ and the HMD $\hat{\underline{\boldsymbol{x}}}_{H_0}$ are presented in Equation 7.74. In this configuration the expected initial lag, $\underline{\boldsymbol{\tau}}_{\ddot{c}_0}$ and $\underline{\boldsymbol{\tau}}_{\dot{\omega}_0}$, were all set to 0.06. Moreover, the SRS expected initial acceleration and angular acceleration were set to zero.

$$\begin{aligned}
\hat{\underline{\boldsymbol{x}}}_0 &= E\left|\underline{\boldsymbol{x}}_0\right| = E\left|\left(\underline{\boldsymbol{x}}_{P_0}^T \quad \underline{\boldsymbol{x}}_{H_0}^T\right)^T\right| \\
&= E\left|\left(\underline{\boldsymbol{c}}_{P_0}^T \quad \dot{\underline{\boldsymbol{c}}}_{P_0}^T \quad \ddot{\underline{\boldsymbol{c}}}_{P_0}^T \quad \underline{\boldsymbol{e}}_{P_0}^T \quad \underline{\boldsymbol{\omega}}_{P_0}^T \quad \dot{\underline{\boldsymbol{\omega}}}_{P_0}^T \quad \underline{\boldsymbol{\tau}}_{\ddot{c}_0}^T \quad \underline{\boldsymbol{\tau}}_{\dot{\omega}_0}^T\right)^T \left(\underline{\boldsymbol{c}}_{H_0}^T \quad \dot{\underline{\boldsymbol{c}}}_{H_0}^T \quad \underline{\boldsymbol{e}}_{H_0}^T \quad \underline{\boldsymbol{\lambda}}_{H_0}^T\right)^T\right|
\end{aligned}$$

$$\begin{Bmatrix} \hat{\underline{\boldsymbol{c}}}_{P_0} \\ \hat{\dot{\underline{\boldsymbol{c}}}}_{P_0} \\ \hat{\ddot{\underline{\boldsymbol{c}}}}_{P_0} \\ \hat{\underline{\boldsymbol{e}}}_{P_0} \\ \hat{\underline{\boldsymbol{\omega}}}_{P_0} \\ \hat{\dot{\underline{\boldsymbol{\omega}}}}_{P_0} \\ \hat{\underline{\boldsymbol{\tau}}}_{\ddot{c}_0} \\ \hat{\underline{\boldsymbol{\tau}}}_{\dot{\omega}_0} \\ \hat{\underline{\boldsymbol{c}}}_{H_0} \\ \hat{\dot{\underline{\boldsymbol{c}}}}_{H_0} \\ \hat{\underline{\boldsymbol{e}}}_{H_0} \\ \hat{\underline{\boldsymbol{\lambda}}}_{H_0} \end{Bmatrix} = \begin{Bmatrix} \left(0.1173657 \quad -0.19418465 \quad -2.13843659\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0.99955551 \quad 0.01350639 \quad 0.00515045 \quad -0.02607374\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0.06 \quad 0.06 \quad 0.06\right)^T \\ \left(0.06 \quad 0.06 \quad 0.06\right)^T \\ \left(0.12892825 \quad -0.04333829 \quad -3.01418109\right)^T \\ \left(0 \quad 0 \quad 0\right)^T \\ \left(0.99924809 \quad 0.00405833 \quad 0.02351122 \quad 0.03056158\right)^T \\ \left(0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\right)^T \end{Bmatrix} \tag{7.74}$$

The expected initial state covariance $\boldsymbol{P}_{\hat{x}_0}$, presented in Equation 7.75, is just as the expected initial state $\hat{\underline{\boldsymbol{x}}}_0$ similar to the expected initial state covariance $\boldsymbol{P}_{\hat{x}_0}$ presented in Configuration 2 in Equation 7.72. The initial covariance of the lag states of the SRS were set initially small, $\left(10^{-5}\right)^2$. The covariance of the lags was set low relative to the others, i.e. the certainty was relatively high, in order to promote the correction of other states in favor of the lag state in the initial phase of the UKF.

$$\begin{aligned}
\boldsymbol{P}_{\hat{x}_0} &= E\left|\left[\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0\right]\left[\underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0^T\right]\right| \\
&= \operatorname{diag}\left(\begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}(\mathrm{rad}(25))^2\\(\mathrm{rad}(25))^2\\(\mathrm{rad}(25))^2\end{Bmatrix}^T \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}\left(10^{-5}\right)^2\\\left(10^{-5}\right)^2\\\left(10^{-5}\right)^2\end{Bmatrix}^T \begin{Bmatrix}\left(10^{-5}\right)^2\\\left(10^{-5}\right)^2\\\left(10^{-5}\right)^2\end{Bmatrix}^T \right. \\
&\qquad \left. \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}1\\1\\1\end{Bmatrix}^T \begin{Bmatrix}(\mathrm{rad}(30))^2\\(\mathrm{rad}(30))^2\\(\mathrm{rad}(30))^2\end{Bmatrix}^T \begin{Bmatrix}\left(10^{-3}\right)^2\\\left(10^{-3}\right)^2\\\left(10^{-3}\right)^2\\\left(10^{-4}\right)^2\\\left(10^{-4}\right)^2\\\left(10^{-4}\right)^2\end{Bmatrix}^T \right)
\end{aligned} \tag{7.75}$$

The SRS related variances were set after some trial and error. The expected process noise covariance on the lag was set small so the estimated lag would not change quickly and represent the errors in the system. The expected variance on the SRS acceleration and angular acceleration relaxes the prediction step of the UKF by introducing additional uncertainty. Moreover, the addition of said variance makes this configuration more robust to apply to different motion platforms. If the setpoints of a motion platform are generated by a different discrete integration method than presented here, the variance on the acceleration and angular acceleration relaxes the need for an exact knowledge on the generation of said setpoints. This is a trade-off between precision and ease of integration, i.e. tuning.

$$\begin{aligned}\boldsymbol{Q} &= \operatorname{diag}\left(\underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\tau}_{\ddot{\boldsymbol{c}}_P}} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\tau}_{\dot{\boldsymbol{\omega}}_P}} \quad \underline{\boldsymbol{\sigma}}^2_{\ddot{\boldsymbol{c}}_P} \quad \underline{\boldsymbol{\sigma}}^2_{\dot{\boldsymbol{\omega}}_P} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{f}_H} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\omega}_H} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{f}_{\boldsymbol{\mu}_H}} \quad \underline{\boldsymbol{\sigma}}^2_{\boldsymbol{\omega}_{\boldsymbol{\mu}_H}}\right) \\ &= \operatorname{diag}\left(\begin{Bmatrix}0.001^2\\0.001^2\\0.001^2\end{Bmatrix}^T \begin{Bmatrix}0.001^2\\0.001^2\\0.001^2\end{Bmatrix}^T \begin{Bmatrix}0.5^2\\0.5^2\\0.5^2\end{Bmatrix}^T \begin{Bmatrix}0.5^2\\0.5^2\\0.5^2\end{Bmatrix}^T \right. \\ &\qquad \left. \begin{Bmatrix}0.030^2\\0.029^2\\0.047^2\end{Bmatrix}^T \cdot 1.1^2 \begin{Bmatrix}0.0033^2\\0.0036^2\\0.0038^2\end{Bmatrix}^T \cdot 1.1^2 \begin{Bmatrix}0.015^2\\0.064^2\\0.048^2\end{Bmatrix}^T \cdot 1.1^2 \begin{Bmatrix}0.00018^2\\0.00034^2\\0.00045^2\end{Bmatrix}^T \cdot 1.1^2 \right)\end{aligned} \tag{7.76}$$

The total accuracy of the HMD vehicle-fixed pose of Configuration 3 is compared against Configuration 0, Configuration 1, and Configuration 2 in Table 7.14. This configuration improves over the base-line in similar fashion to the other configuration. The scale of improvements on the total accuracy of the HMD vehicle-fixed pose of all three configurations over the base-line configuration are similar. Leading to the conclusion that the main gain can be contributed to the addition of the IMU on the HMD.

Table 7.14: The error of each run in terms of the vehicle-fixed position and attitude of the HMD.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}
1	1.70×10^{-4}	8.22×10^{-5}	5.05×10^{-4}	6.87×10^{-3}	3.35×10^{-3}	2.01×10^{-2}
2	1.93×10^{-4}	9.56×10^{-5}	5.80×10^{-4}	9.43×10^{-3}	4.78×10^{-3}	2.47×10^{-2}
3	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.76×10^{-3}	3.29×10^{-3}	1.95×10^{-2}

The NEES in this configuration can be tuned for the SRS easily by the variance on the acceleration and angular acceleration. Based on the values for the SRS the variance on the acceleration and angular acceleration in the process noise covariance $\boldsymbol{Q}$ is set marginally too large. Decreasing said variances would increase the SRS accuracy slightly. The innovation based NEES on the linear encoders verify the NEES found on the states themselves. This is an important indicator for real life tests. For both system the innovation based NEES can be employed to tune the parameters set in the process noise covariance $\boldsymbol{Q}$.

Table 7.15: Condition overview of the base run of Configuration 3, see Section 7.4.1 for the explanation on the color use.

	NEES					
	SRS			HMD		
Run	Position	Attitude	Lin. Enc. Inno.	Position	Attitude	TrackIR 5 Inno.
1	2.59	2.62	7.43	3.11	2.01	5.00
2	26.40	15.92	26.59	3.90	3.79	5.18
3	2.35	2.31	4.64	3.11	1.97	5.11

The HMD error on the position and attitude, depicted in Figures 7.26 and 7.27 are similar to the error found in Configuration 1. Both the position and attitude take about one second to converge.

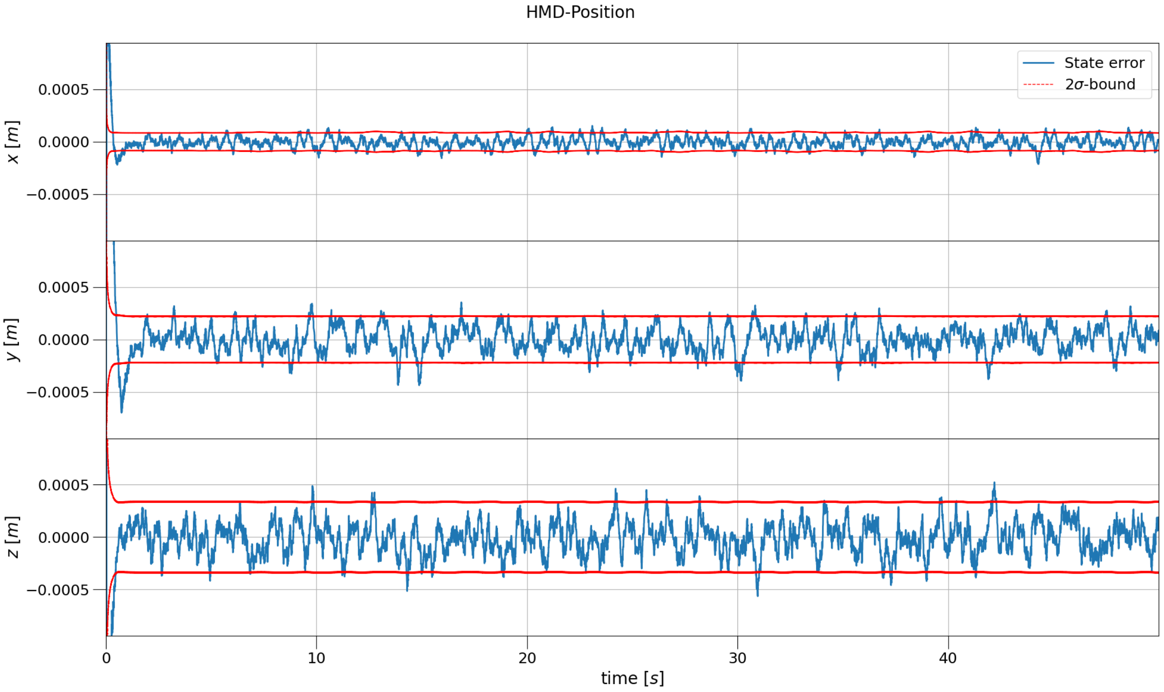


Figure 7.26: Base run configuration 3, the error of the HMD estimated position per axis of the inertial frame plotted against its covariance over time.

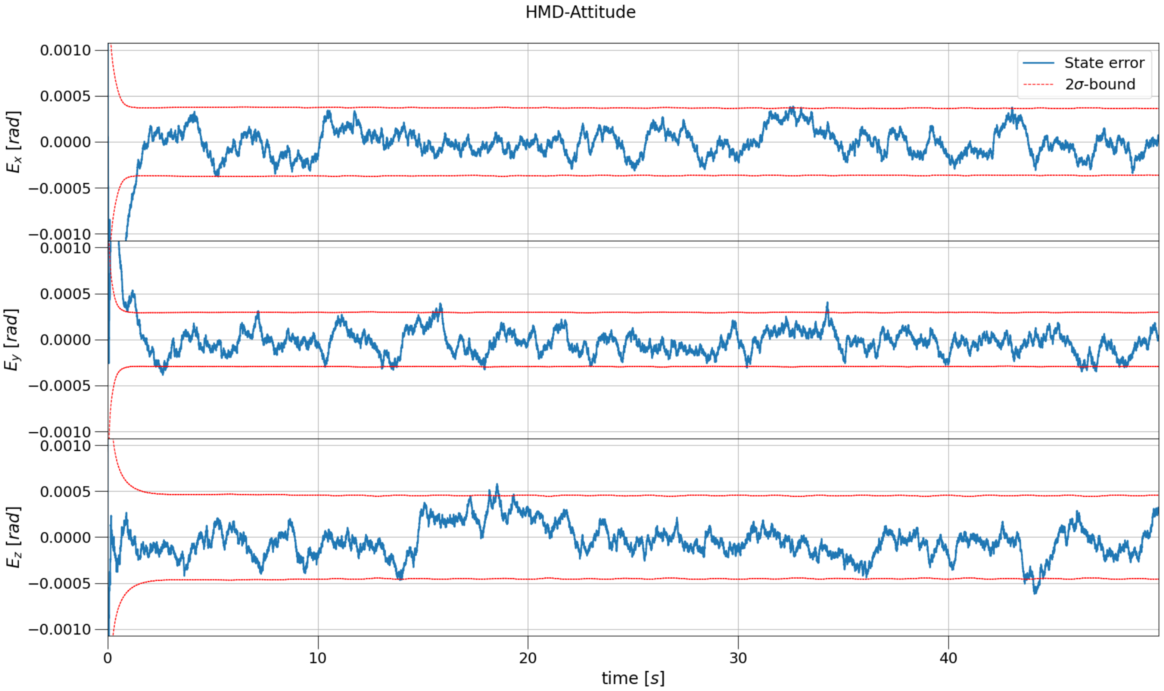


Figure 7.27: Base run configuration 3, the error of the HMD estimated attitude in angle-axis formulation per axis of the inertial frame plotted against its covariance over time.

For brevity's sake, the HMD accelerometer and gyroscope biases are not shown. They performed similar as in Configuration 1.

The total accuracy of the SRS in the inertial frame of Configuration 3 is compiled in Table 7.16. Even though Configuration 3 results in the most accurate configuration in terms of SRS. However, this does not translate to the HMD vehicle-fixed pose compiled in Table 7.14. Following, the total accuracy of the HMD vehicle-fixed pose is seemingly limited by the sensors used, i.e. TrackIR 5 and IMU HMD. If the accuracy of the sensors, TrackIR 5 or MPU-6500 IMU, used would improve, Configuration 3 could be more beneficial.

Table 7.16: The error of each run in terms of the inertial position and attitude of the SRS.

	SRS					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
1	2.95×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.71×10^{-4}	5.52×10^{-4}	3.58×10^{-3}
2	2.98×10^{-5}	1.71×10^{-5}	9.23×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.54×10^{-3}
3	7.61×10^{-6}	3.46×10^{-6}	2.12×10^{-5}	1.80×10^{-4}	7.92×10^{-5}	5.16×10^{-4}

The SRS error on the position and attitude is plotted in Figures 7.28 and 7.29 against there respective covariance. As the NEES in Table 7.15 hinted at, the error on the position and attitude is seemingly Gaussian and well within 2σ-bounds.

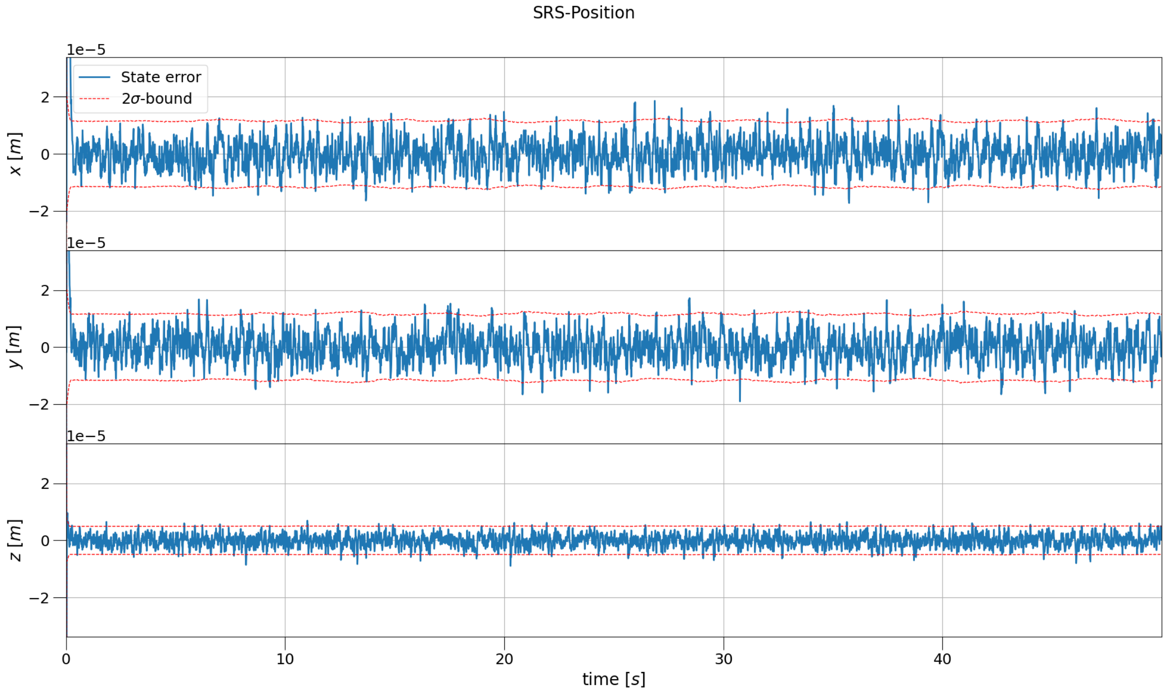


Figure 7.28: Base run configuration 3, the error of the SRS estimated position per axis of the inertial frame plotted against its covariance over time.

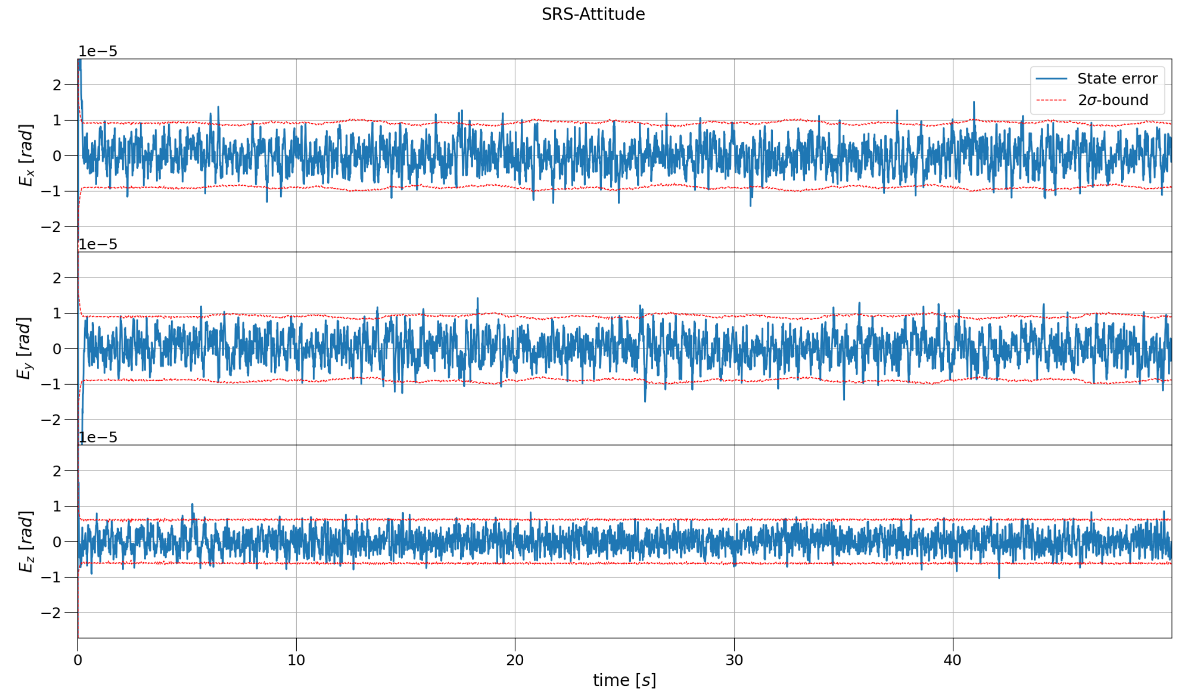


Figure 7.29: Base run configuration 3, the error of the SRS estimated attitude in angle-axis formulation per axis of the inertial frame plotted against its covariance over time.

The error on the translational lags $\underline{\tau}_{\ddot{c}}$ show interesting behavior in Figure 7.30. In this figure, all three lag errors converge to approximately -0.0017 s, roughly equal to the negative time-step Δt. This is not a coincidence, in the forward Euler method employed in the translation integration, the acceleration takes one extra time-step to influence the position. As the acceleration setpoints are generated based on the actual acceleration, and not the 'required' acceleration, see Section 7.2.3.3, the lag is reduced to bring the setpoints, i.e. the acceleration, forward in time to match the observation by the linear encoders. Moreover, the lag converges only after 10 s. In Section 7.5, it is looked into if this is linked to the initial 10 s linear scaling of the motion.

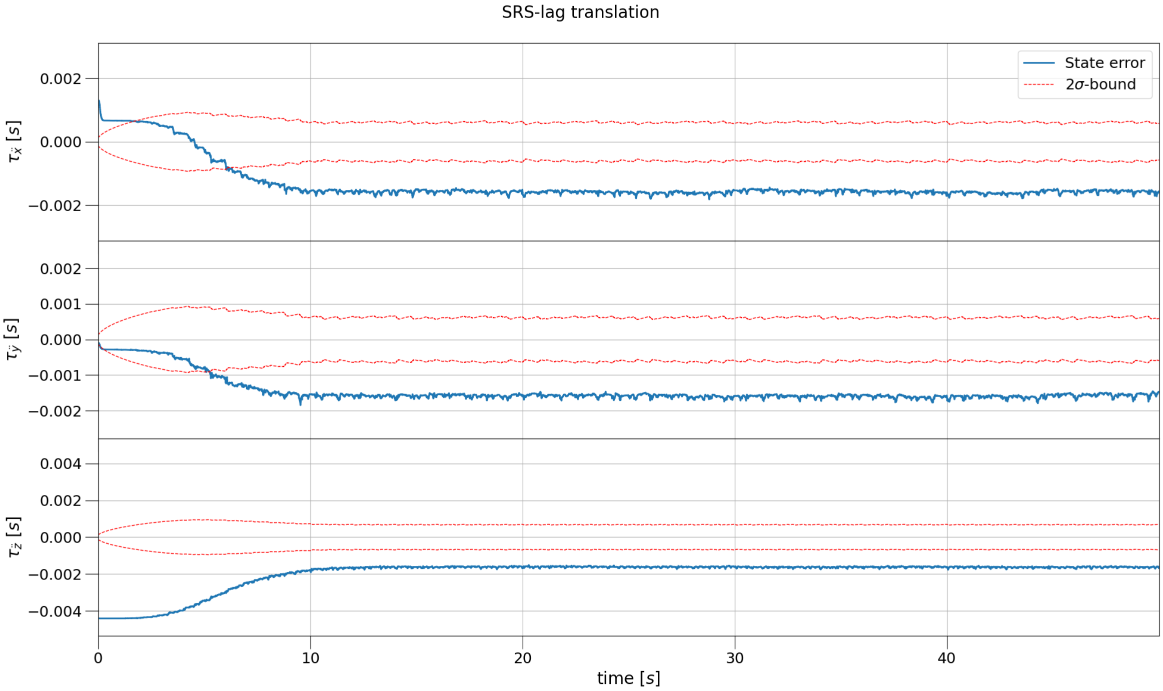


Figure 7.30: Base run configuration 3, the error on the translational lag over the acceleration setpoints of the SRS per axis plotted against its covariance over time.

The error on the rotational lags $\underline{\tau}_{\dot{\omega}}$ are depicted in Figure 7.31. These, in contrary to the translational lags $\underline{\tau}_{\ddot{c}}$, converge around zero. This makes sense as the angular acceleration setpoints are based on the ‘required’ angular acceleration rather than the actual acceleration, see Sections 7.2.3.3 in combination with Equations 7.25 and 7.32. Similar to their translational counterparts in Figure 7.30 they converge after 10 s.

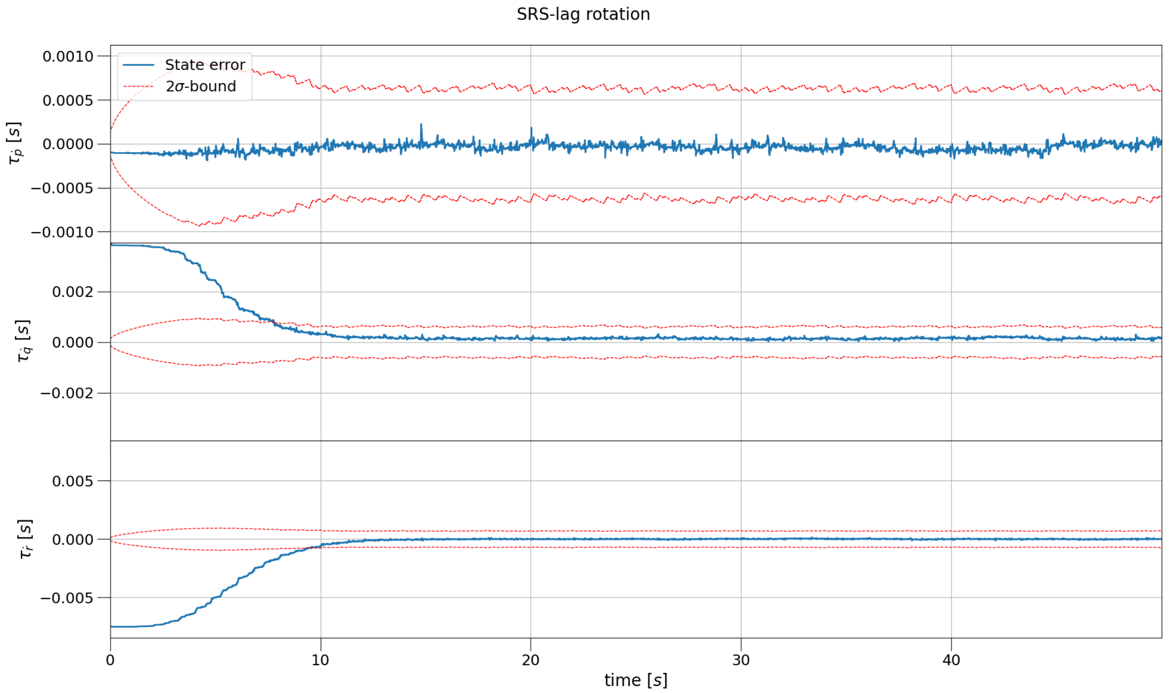


Figure 7.31: Base run configuration 3, the error on the rotational lag over the angular acceleration setpoints of the SRS per axis plotted against its covariance over time.

Following, the innovations, from TrackIR 5 $\underline{\boldsymbol{\epsilon}}_{HP}$ and linear encoders $\underline{\boldsymbol{\epsilon}}_{l_i}$, are depicted against their relevant observation covariance $\boldsymbol{P}_{\hat{y}_k}$ in Figures 7.32 and 7.33. Both innovations plotted are within 2σ-bounds and seemingly Gaussian as the NEES values in Table 7.15 would indicate.

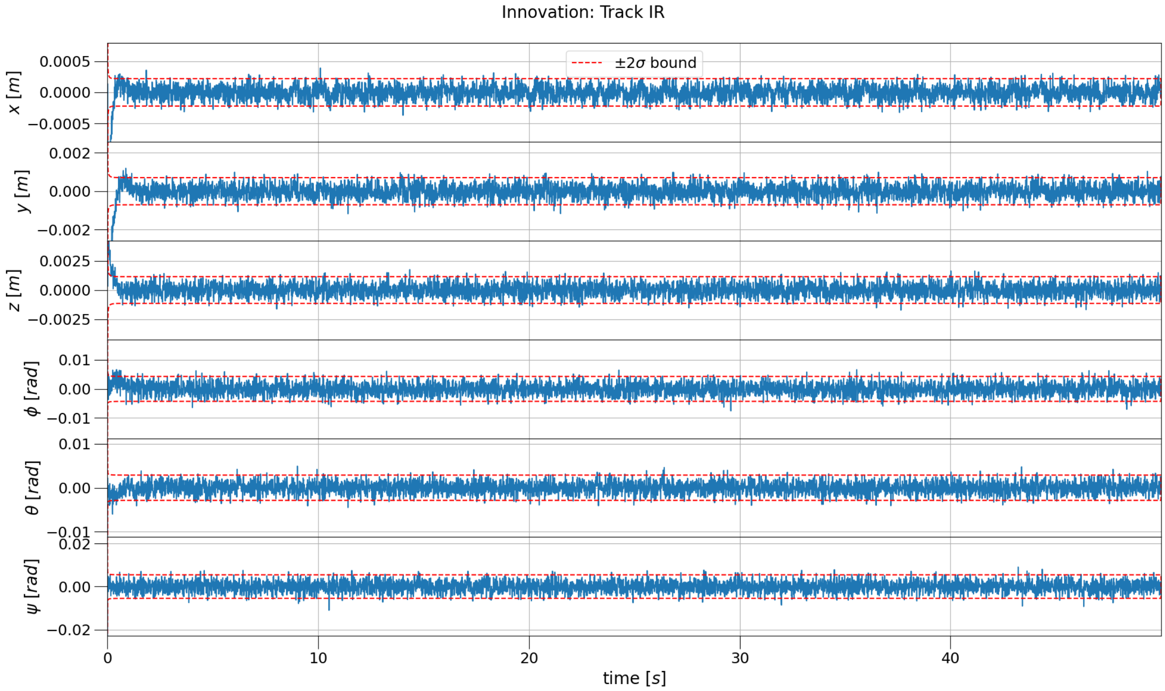


Figure 7.32: Base run configuration 3, the innovation of Track IR 5 measuring the position and attitude (Euler attitude formulation) of the HMD attached to the SRS plotted against its covariance over time.

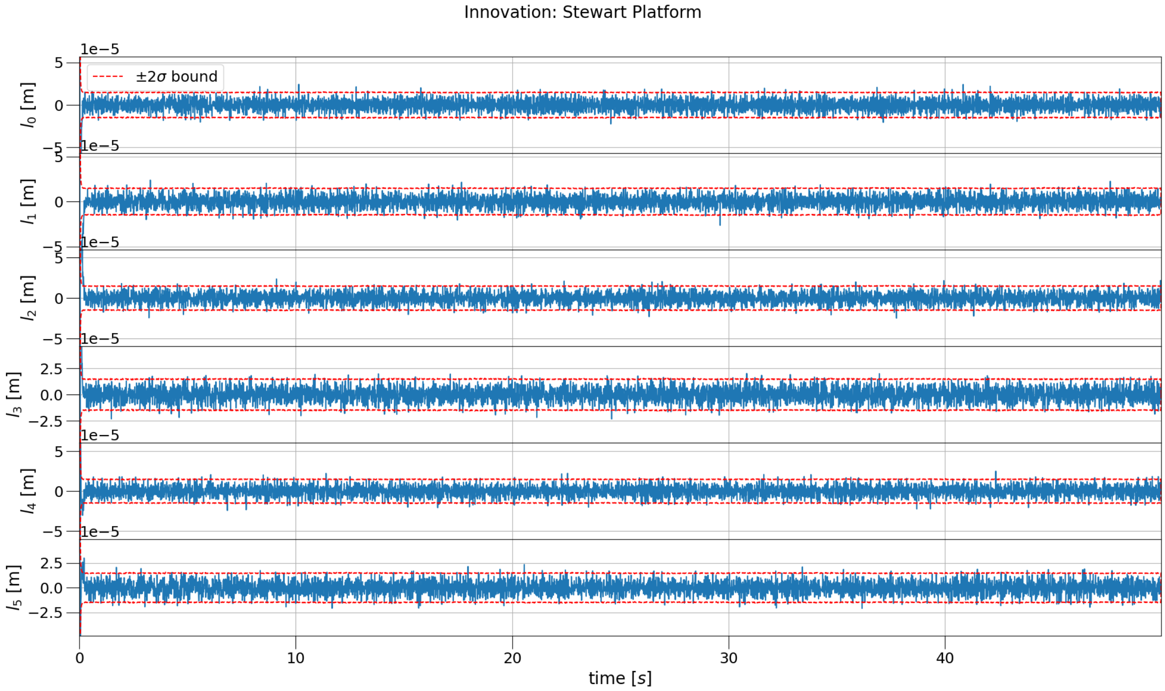


Figure 7.33: Base run configuration 3, the innovation of linear encoders measuring the length of the hydraulic actuators of the SRS plotted against its covariance over time.

7.5. Sensitivity Analysis

In this section a sensitivity analysis is performed. Meaning, certain initial parameters per configuration are modified and the resulting run analyzed in terms of performance metrics. The parameters selected to be modified are listed for every configuration.

The resulting performance metrics are compared to their base run established in Section 7.4.2 in tables. The tables that present the error on the HMD vehicle-fixed pose and inertial SRS pose will use color to help identify improvements or decreases to the accuracy due to the modified parameters. Green is used to highlight improvements and red for reductions in accuracy.

7.5.1. Configuration 1: SRS's Linear Encoders + HMD's Visual Tracking and IMU

The sensitivity analysis of Configuration 1 looks at the influence of process noise covariance $\boldsymbol{Q}$ in terms of acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{\boldsymbol{c}}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\boldsymbol{\omega}}_P}$. As these terms were initial set due to trial and error after all. Moreover, the influence of the expected/actual accuracy of the IMU sensor of the HMD and linear encoders of the SRS are investigated. Next, the robustness of this configuration is tested by increasing the estimation error on the initial state. Subsequently, the reduction in accuracy is investigated if the computational frequency were reduced from $600\,$Hz to $100\,$Hz. Last, the amplitudes and frequencies presented in Section 7.2 are doubled to test the robustness of this configuration against an increase in the overall motion.

Configuration 1

- **Process noise $\boldsymbol{Q}$**

  1.1 SRS: Acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{\boldsymbol{c}}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\boldsymbol{\omega}}_P}$

  1.1.1 Increase expected SD: $\times 10$

  1.1.2 Decrease expected SD: $\div 10$

  1.2 HMD: IMU $\underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_H}$, $\underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_H}$

  1.2.1 Increase actual SD: $\times 10$

  1.2.1.1 Change expected SD accordingly

  1.2.1.2 No change to expected SD

  1.2.2 Decrease actual SD: $\div 10$

  1.2.2.1 Change expected SD accordingly

  1.2.2.2 No change to expected SD

- **Observation noise $\boldsymbol{R}$**

  1.3 SRS: linear encoders $\underline{\boldsymbol{\sigma}}_{\underline{\boldsymbol{l}}_i}$

  1.3.1 Increase actual SD: $\times 10$

  1.3.1.1 Change expected SD accordingly

  1.3.1.2 No change to expected SD

  1.3.2 Decrease actual SD: $\div 10$

  1.3.2.1 Change expected SD accordingly

  1.3.2.2 No change to expected SD

- **Initial state $\hat{\underline{\boldsymbol{x}}}_0$**

  1.4 Increase error

- **Computational frequency**

  1.5 Decrease to $100\,$Hz

- **Motion profile**

  1.6 Double amplitude and frequency

The impact on the performance metrics of each alteration on Configuration 1 is compiled into Tables 7.17, 7.18, and 7.19. In each of these tables, the modified runs are compared to the base run established in Section 7.4.2.1.

First, **Runs 1.1.1 and 1.1.2**, where the acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ are scaled up and down, are analyzed from the data provided in the three tables below, Tables 7.17, 7.18, and 7.19. Increasing the acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ improved absolute performance of the SRS, indicated in Table 7.19. However the NEES values in Table 7.18 of the SRS, especially the innovation, indicates that the uncertainty is larger than necessary. The increased uncertainty influenced also the TrackIR 5 sensor as evident from the NEES value. Nonetheless, this barely influenced the HMD vehicle-fixed pose as evident in Table 7.17 due to the limited accuracy of the TrackIR 5 sensor when compared to the SRS linear encoders.

Decreasing the expected SD on the acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ increases the error for both subsystems, HMD and SRS. In a real use case, the very small NEES values of the innovation could be an indicator of a lack of certainty, as seen in Table 7.18.

Second, **Runs 1.2.1.1 to 1.2.2.2** alter the actual and expected SD of the HMD IMU. These changes have an impact on the accuracy of the HMD vehicle-fixed pose. The result of increasing the actual SD of the HMD IMU are as expected. The error goes up, especially in Run 1.2.1.2 where the increase in actual SD on the IMU is not taken into account.

The results of the runs in which the actual SD of the IMU was decreased, i.e. 1.2.2.1 and 1.2.2.2, are unexpected. The NEES values indicate suboptimal conditions in both runs. Run 1.2.2.1, that takes into account the increased accuracy of the IMU,scales the expected SD of the accelerometer with 2.1 according to Section 7.2.4. Moreover, Run 1.2.2.1 has decreased positional but increased attitude performance according to Table 7.17. While Run 1.2.2.2, that not takes into account the increased accuracy of the IMU of the HMD, its performance moves in the opposite manner to Run 1.2.2.1, i.e. increased positional but decreased attitude performance.

The scaled expected SD of the accelerometer in Run 1.2.2.1 of 2.1 is apparently not enough to overcome the non-Gaussian noise present due to the forward Euler integration on the accelerometer of the HMD. The forward induced error is not present on the attitude in this simulation for reasons mentioned in Section 7.2. The difference between the method of computation of the angular rate and acceleration relevant data also explains why sometimes only the attitude but not the position accuracy improves.

These runs, Runs 1.2.1.1 to 1.2.2.2 have an impact on the SRS estimated pose as well. This shows that the HMD IMU has an impact on the SRS pose estimation translated through the TrackIR 5. However, the impact is rather small according to Table 7.19.

Runs 1.3.1.1 to 1.3.2.2 alter the expected and actual SD of the linear encoders $\underline{\boldsymbol{\sigma}}_{\underline{\boldsymbol{l}}_i}$. Reducing the accuracy by a factor of ten in Runs 1.3.1.1 and 1.3.1.2 increases the error of the pose by a factor of about five in both the position as well the attitude of the SRS.

The HMD vehicle-fixed pose is less influenced by the reduction in accuracy of the accuracy of the linear encoders. Here, the total positional error increased around $50\,\%$ while the total attitude error increased around $15\,\%$ to Run 1.

The NEES values for Run 1.3.1.2, i.e. not taking into account the increased actual SD on the linear encoders, meaning the expected SD is smaller than the actual SD creates unfavorable conditions, resulting in sigma-points sampled closer to the state than would be optimal. Moreover, the UKF algorithm would also trust the observation too much in this case.

In Runs 1.3.2.1 to 1.3.2.2 the accuracy of the linear encoders is improved by a factor of ten. The total error of the SRS, in the Run 1.3.2.1, is reduced by $43\,\%$ and $35\,\%$ for the position and attitude respectively. Again, the estimation improvement in the SRS subsystem are barely measurable in the HMD vehicle-fixed pose error.

Run 1.3.2.2 reduces the mean error on the pose by approximately $20\,\%$ when not taking into account the improved linear encoders. The lack of certainty in the sensors is evident from the NEES values in

Table 7.18. NEES values this small indicate that the algorithm expects the estimated state to change more than it will, i.e. the state covariance $\boldsymbol{P}_{\hat{x}}$ is to large. If the SRS subsystem would have had another input, say an IMU such as in Configuration 2 in Section 7.4.2.1, the NEES value of the innovation would have been small and resembled the discrepancy between the expected and actual SD on the sensors output.

In **Run 1.4** the error on the initial state $\hat{\underline{x}}_0$ was modified from the state given in Equation 7.66 to:

$$\hat{\underline{x}}_0 = \begin{Bmatrix} \hat{\underline{c}}_{P_0} \\ \hat{\dot{\underline{c}}}_{P_0} \\ \hat{\underline{e}}_{P_0} \\ \hat{\underline{\omega}}_{P_0} \\ \hat{\underline{c}}_{H_0} \\ \hat{\dot{\underline{c}}}_{H_0} \\ \hat{\underline{e}}_{H_0} \\ \hat{\underline{\lambda}}_{H_0} \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} 0.58682858 & -0.97092327 & -1.13218296 \end{pmatrix}^T \\ \begin{pmatrix} 1.22803562 & 0.46829199 & -2.37069145 \end{pmatrix}^T \\ \begin{pmatrix} 0.99364229 & 0.10614029 & -0.03567829 & -0.0116746 \end{pmatrix}^T \\ \begin{pmatrix} 0.12301066 & 0.71263982 & 0.9263408 \end{pmatrix}^T \\ \begin{pmatrix} -0.47145191 & -0.60514862 & -3.50374137 \end{pmatrix}^T \\ \begin{pmatrix} 1.2907381 & 0.99407434 & 1.91678943 \end{pmatrix}^T \\ \begin{pmatrix} 0.99311633 & -0.11389189 & 0.02575446 & -0.00923585 \end{pmatrix}^T \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \end{Bmatrix} \tag{7.77}$$

Here, alongside increasing the initial estimation errors on the position and attitude, estimation errors are introduced on the velocity and angular velocity. Again, these values are extreme and, in the case of the position of the HMD, physically impossible. But, this initial state $\hat{\underline{x}}_0$ was chosen as such to investigate if the algorithm is mathematically robust to deal with them as such. The initial estimated SRS state had a random positional error of 1.69 m, velocity error of 2.71 m/s, attitude error of $6.46°$, and angular rate of 67.33 °/s. The initial estimate state of the HMD in this run had a random error on the position of 0.92 m, velocity of 2.52 m/s, and attitude of $6.73°$.

The algorithm took longer to converge onto the HMD pose, in this case about 2 s. The SRS pose, as was the case in the base run in Figures 7.13 and 7.14, converges as soon as the first observation is sampled. The reason for the longer convergence duration is due to the algorithm initially solving the mismatch between the HMD's IMU input and observation by contributing the error to the HMD's IMU biases, shown in Figures 7.34 and 7.35.

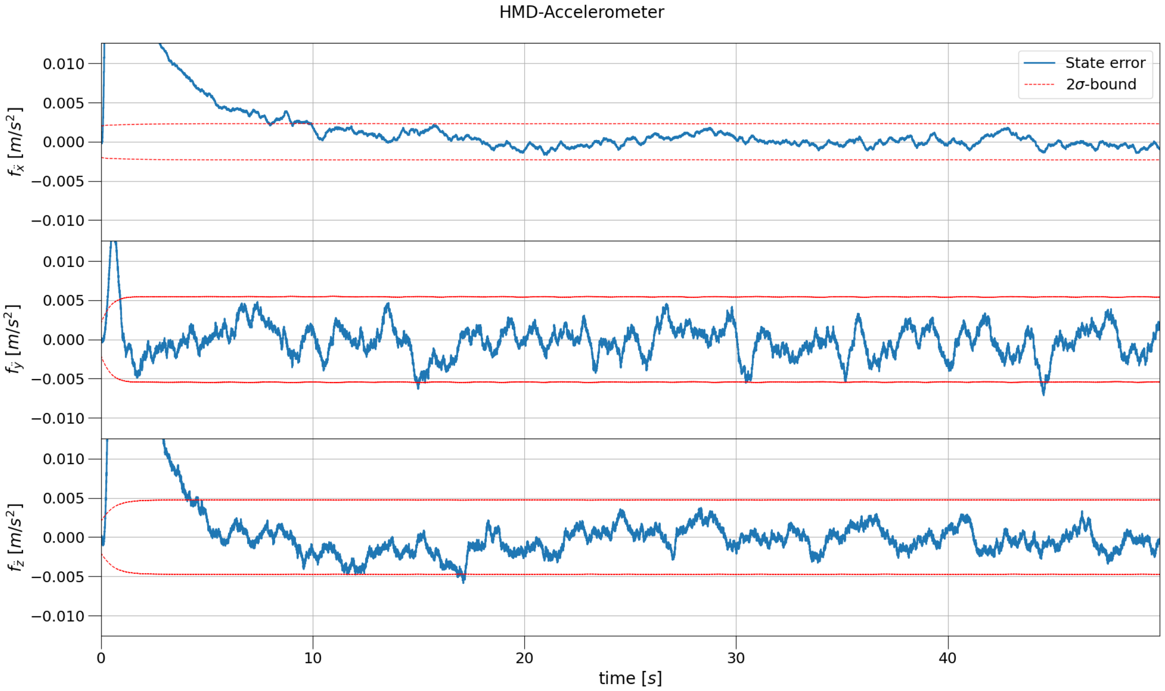


Figure 7.34: Sensitivity Run 1.4, the error of the accelerometer bias of the HMD per axis plotted against its covariance over time.

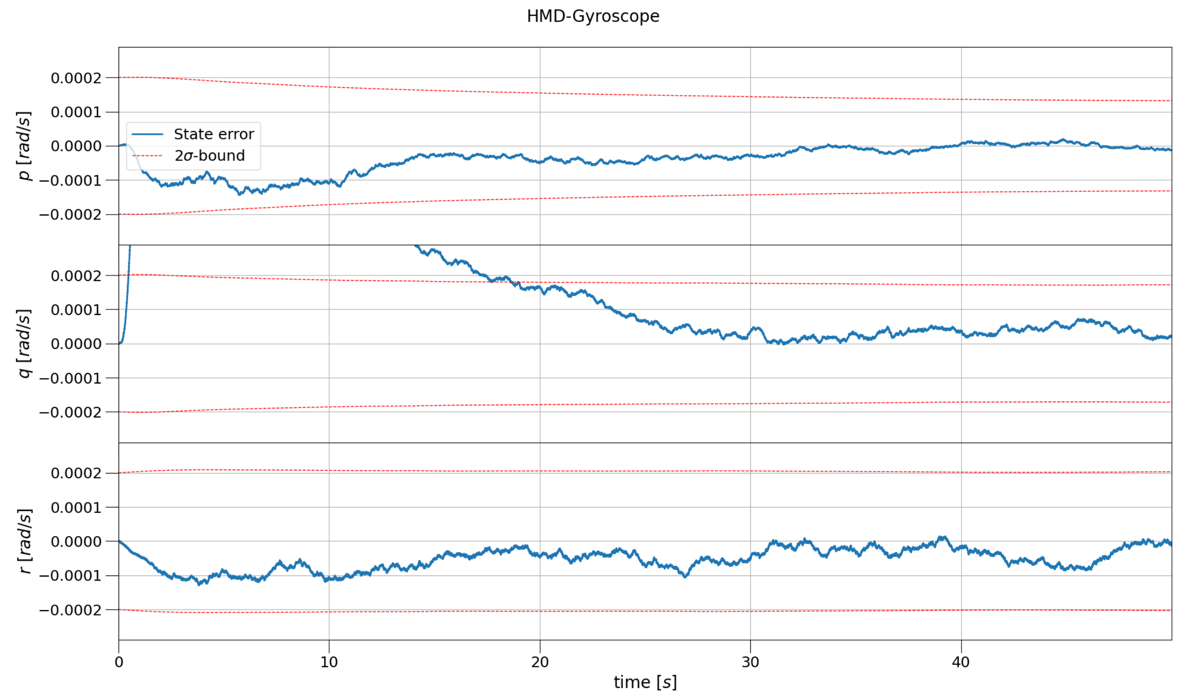


Figure 7.35: Sensitivity Run 1.4, the error of the gyroscope bias of the HMD per axis plotted against its covariance over time.

Next, **Run 1.5** decreases the computational frequency from 600 Hz to 100 Hz. In order to compare the errors found in Tables 7.17 and 7.19 the error of Run 1.5 was computed at 600 Hz. Meaning, that every state estimation was compared against the actual state sampled at 600 Hz, using every estimated state six times when comparing against the sampled actual state. It must be noted that the expected SD of the accelerometer of the HMD IMU was scaled by 1.5 as found in Section 7.2.4. Nonetheless, the scaling on the expected SD of the accelerometer the NEES value of the HMD increases further, indicating more motion present in the error. For the HMD vehicle-fixed pose the mean error increases by 3.3 and 7.4 for the position and attitude respectively. Compared against the base-line, Configuration 0, the improvements, especially in the positional case, are small, 5 to $25\,\%$, depending on the metric. Underlying the notion that with an IMU attached to the HMD the state should be estimated at a higher interval in order to generate benefits from having the IMU in the first place. The decrease in accuracy for the SRS is surprising given that the sampling rate of the linear encoders is 100 Hz. The real source of the increase in error has more to do with the fact that no interim estimations are provided by integration of the velocity and angular rate in the prediction phase.

Finally, **Run 1.6** scales the amplitudes and frequencies presented in Tables 7.1 and 7.2 in Section 7.2 by a factor of two. In order to compare Configuration 1 to 0, the baseline, Run 0.6 was created. In Run 0.6 the same motion profile, as used in Run 1.6, was observed by only the TrackIR 5 sensor. Following the increased error of the assumption that the IMU's input is a constant for the time-step's Δt duration due to the increased motion, the error on the HMD's vehicle-fixed pose has increased in Table 7.17 for both Runs 1.6 and 0.6 when compared to Runs 1 and 0 respectively.

However, analyzing the reduction of the average error of the HMD's vehicle-fixed pose per motion profile of Configuration 0 to 1, i.e. Run 1 to Run 0 and Run 1.6 to Run 0.6, hints at the fact that the addition of the HMD's IMU improves the HMD's pose estimation more when there is more motion. Due to the fact that for the original motion profile, i.e. Run 0 to 1,the average error reduced by a factor 4.3 and 15.1 for the position and attitude respectively, while for the increased motion profile, i.e. Run 0.6 to 1.6, the average error reduced by a factor 5.1 for the position and 23.5 for the attitude in Table 7.17. Even though the accuracy of the SRS's pose estimation was reduced by almost a factor of 10.

Table 7.17: The error of each Configuration 1 modified run in terms of the vehicle-fixed position and attitude of the HMD.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}
1	1.70×10^{-4}	8.22×10^{-5}	5.05×10^{-4}	6.87×10^{-3}	3.35×10^{-3}	2.01×10^{-2}
1.1.1	1.72×10^{-4}	8.26×10^{-5}	5.13×10^{-4}	6.82×10^{-3}	3.37×10^{-3}	2.00×10^{-2}
1.1.2	2.49×10^{-4}	1.17×10^{-4}	7.56×10^{-4}	9.11×10^{-3}	4.01×10^{-3}	2.21×10^{-2}
1.2.1.1	2.83×10^{-4}	1.40×10^{-4}	9.23×10^{-4}	2.41×10^{-2}	1.13×10^{-2}	7.26×10^{-2}
1.2.1.2	6.06×10^{-4}	3.44×10^{-4}	1.85×10^{-3}	4.92×10^{-2}	2.06×10^{-2}	9.90×10^{-2}
1.2.2.1	1.82×10^{-4}	9.94×10^{-5}	5.52×10^{-4}	6.76×10^{-3}	2.37×10^{-3}	1.63×10^{-2}
1.2.2.2	1.61×10^{-4}	8.13×10^{-5}	5.16×10^{-4}	6.96×10^{-3}	2.64×10^{-3}	1.48×10^{-2}
1.3.1.1	2.51×10^{-4}	1.26×10^{-4}	8.25×10^{-4}	7.70×10^{-3}	3.88×10^{-3}	2.37×10^{-2}
1.3.1.2	2.86×10^{-4}	1.33×10^{-4}	9.24×10^{-4}	7.95×10^{-3}	3.66×10^{-3}	2.55×10^{-2}
1.3.2.1	1.69×10^{-4}	8.21×10^{-5}	5.10×10^{-4}	6.81×10^{-3}	3.37×10^{-3}	2.02×10^{-2}
1.3.2.2	1.68×10^{-4}	8.18×10^{-5}	5.10×10^{-4}	6.84×10^{-3}	3.36×10^{-3}	2.03×10^{-2}
1.4	1.78×10^{-4}	8.98×10^{-5}	6.30×10^{-4}	7.68×10^{-3}	3.92×10^{-3}	1.94×10^{-2}
1.5	5.61×10^{-4}	3.47×10^{-4}	2.17×10^{-3}	5.07×10^{-2}	3.58×10^{-2}	1.52×10^{-1}
0.6	1.99×10^{-3}	1.41×10^{-3}	7.57×10^{-3}	2.09×10^{-1}	1.37×10^{-1}	6.84×10^{-1}
1.6	3.92×10^{-4}	1.64×10^{-4}	9.61×10^{-4}	8.89×10^{-3}	4.34×10^{-3}	2.88×10^{-2}

Table 7.18: Condition overview of the modified runs of Configuration 1, see Section 7.4.1 for the explanation on the colors.

	NEES					
	SRS			HMD		
Run	Position	Attitude	Lin. Enc. Inno.	Position	Attitude	TrackIR 5 Inno.
1	2.59	2.62	7.43	3.11	2.01	5.00
1.1.1	2.47	2.49	0.09	3.10	1.98	3.56
1.1.2	209.87	220.08	660.57	3.18	3.27	7.07
1.2.1.1	2.58	2.61	7.41	2.48	2.47	4.97
1.2.1.2	2.69	2.73	7.43	45.21	105.92	9.69
1.2.2.1	2.61	2.63	7.44	10.03	4.08	5.31
1.2.2.2	2.58	2.61	7.43	2.74	1.92	4.98
1.3.1.1	4.36	4.33	9.74	3.04	1.93	5.55
1.3.1.2	236.33	238.79	94.06	4.42	1.99	10.06
1.3.2.1	2.46	2.48	6.56	3.11	1.99	4.98
1.3.2.2	0.26	0.26	6.57	3.10	2.00	4.94
1.4	2.68	2.61	7.43	3.60	2.24	4.99
1.5	2.46	2.45	1.25	5.42	3.36	4.72
1.6	16.52	16.57	406.70	18.65	1.97	9.64

Table 7.19: The error of each Configuration 1 modified run in terms of the inertial position and attitude of the SRS.

	SRS					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
1	2.95×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.71×10^{-4}	5.52×10^{-4}	3.58×10^{-3}
1.1.1	2.74×10^{-5}	1.86×10^{-5}	1.37×10^{-4}	7.24×10^{-4}	5.10×10^{-4}	3.57×10^{-3}
1.1.2	1.45×10^{-4}	8.42×10^{-5}	4.86×10^{-4}	3.45×10^{-3}	1.88×10^{-3}	1.14×10^{-2}
1.2.1.1	2.94×10^{-5}	2.08×10^{-5}	1.38×10^{-4}	7.72×10^{-4}	5.53×10^{-4}	3.58×10^{-3}
1.2.1.2	2.98×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.78×10^{-4}	5.53×10^{-4}	3.60×10^{-3}
1.2.2.1	2.97×10^{-5}	2.10×10^{-5}	1.39×10^{-4}	7.68×10^{-4}	5.50×10^{-4}	3.57×10^{-3}
1.2.2.2	2.95×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.71×10^{-4}	5.52×10^{-4}	3.58×10^{-3}
1.3.1.1	1.59×10^{-4}	8.99×10^{-5}	5.99×10^{-4}	3.90×10^{-3}	1.97×10^{-3}	1.39×10^{-2}
1.3.1.2	1.67×10^{-4}	9.05×10^{-5}	5.98×10^{-4}	4.06×10^{-3}	2.15×10^{-3}	1.41×10^{-2}
1.3.2.1	1.69×10^{-5}	1.60×10^{-5}	8.39×10^{-5}	5.02×10^{-4}	4.76×10^{-4}	2.34×10^{-3}
1.3.2.2	2.38×10^{-5}	2.01×10^{-5}	1.08×10^{-4}	6.35×10^{-4}	5.44×10^{-4}	2.84×10^{-3}
1.4	2.98×10^{-5}	2.10×10^{-5}	1.49×10^{-4}	7.73×10^{-4}	5.52×10^{-4}	3.56×10^{-3}
1.5	8.38×10^{-4}	6.55×10^{-4}	3.06×10^{-3}	2.23×10^{-2}	1.68×10^{-2}	8.16×10^{-2}
1.6	1.91×10^{-4}	1.60×10^{-4}	9.11×10^{-4}	5.10×10^{-3}	4.37×10^{-3}	2.43×10^{-2}

7.5.2. Configuration 2: SRS's Linear Encoders and IMU + HMD's Visual Tracking and IMU

The sensitivity analysis of Configuration 2 focusses on the IMU of the SRS, as it is the main difference to Configuration 1. Moreover, the analysis looks into the impact of removing all the bias estimation without changing the underlying IMU model. In this configuration the impact of the noise present on the TrackIR 5 sensor is investigated. Next, the impact of reducing the computational frequency to 100 Hz is looked into. Following, the influence of increasing the SRS IMU sample rate to 600 Hz to match the HMD IMU. Last, the motion profile was changed by doubling the amplitudes and frequencies presented in Section 7.2 as was the case for Run 1.6 of Configuration 1.

Configuration 2

- **Process noise $\boldsymbol{Q}$**
  - 2.1 SRS: IMU $\underline{\boldsymbol{\sigma}}_{\boldsymbol{f}_P}$, $\underline{\boldsymbol{\sigma}}_{\boldsymbol{\omega}_P}$
    - 2.1.1 Increase actual SD: $\times 10$
      - 2.1.1.1 Change expected SD accordingly
      - 2.1.1.2 No change to expected SD
    - 2.1.2 Decrease actual SD: $\div 10$
      - 2.1.2.1 Change expected SD accordingly
      - 2.1.2.2 No change to expected SD
  - 2.2 Ignore induced error
- **Observation noise $\boldsymbol{R}$**
  - 2.3 HMD: TrackIR 5
    - 2.3.1 Increase actual SD: $\times 100$
      - 2.3.1.1 Change expected SD accordingly
      - 2.3.1.2 No change to expected SD
    - 2.3.2 Increase actual SD: $\times 10$
      - 2.3.2.1 Change expected SD accordingly
      - 2.3.2.2 No change to expected SD

2.3.3 Decrease actual SD: $\div 10$

2.3.3.1 Change expected SD accordingly

2.3.3.2 No change to expected SD

2.3.4 Decrease actual SD: $\div 100$

2.3.4.1 Change expected SD accordingly

2.3.4.2 No change to expected SD

- **Computational frequency**

  2.4 Decrease to 100 Hz

- **SRS IMU sample rate**

  2.5 increase from 100 Hz to 600 Hz

- **TrackIR 5 sample rate**

  2.6 increase from 100 Hz to 200 Hz

- **Motion profile**

  2.7 Double amplitude and frequency

First, **Runs 2.1.1.1 and 2.1.1.2** increase, while **Runs 2.1.2.1 and 2.1.2.2** decrease the actual SD on the IMU on the SRS. The modification of this parameter has a direct result on the performance of the SRS, as shown in Table 7.22. Here, the attitude error behaves as expected, the error increase or decreases following the IMU of the SRS. However, the positional error of these aforementioned runs decreases as the expected SD on the IMU increases. This is contrary to the expected outcome but in line with the outcome of Run 1.2.2.1. It must be noted that the new expected SD on the accelerometer of the SRS is scaled, according to Section 7.2.4, by a factor of 1.4 in Run 2.1.1.1, resulting in 14.0 times scaling to the original actual SD of the IMU. Also in Run 2.1.2.1, the new expected SD on both the accelerometer and gyroscope of the IMU on the SRS are rescaled by a factor of 106.6 and 8.6 respectively. Factoring in the reduction in actual SD results in a scaling of 10.66 to the original actual SD on the accelerometer and 0.86 scaling on the gyroscope.

Increasing the SD on the IMU output actually decreased the positional error of the SRS in Run 2.1.1.1. Indicating that the whiter output of the positional predictions improved the estimating performance of the algorithm concerning the SRS. These findings correlate with the NEES values in Table 7.21. Where, from Runs 2.1.1.1 to 2.1.2.2 every time the NEES values improve of the SRS position the estimation improves of the position.

The variance in SRS estimation performance do not translate themselves to the HMD vehicle-fixed estimation performance found in Table 7.20. The improvement in SRS position estimation of Run 2.1.1.1 in particular does not translate. From the viewpoint of the HMD vehicle-fixed pose, the changes to the SRS IMU noise profile the impact is small but expected.

In **Run 2.2** the expected SD scale was ignored on all IMU and set to the standard 1.1. Ignoring the forward Euler induced error, thereby reducing the expected SD on both IMU, i.e. reducing the uncertainty on the IMU readings, makes the UKF algorithm trust the prediction phase more. The positional error of the HMD vehicle-fixed pose increases. The impact on the attitude of the HMD is small as this factor did not change. The angular rate, computed for the gyroscope was based on the required rate not on the actual rate, as was the case for the acceleration, for the reasoning see Section 7.2.4. The NEES values increase in Table 7.21, indicating an increase in motion present in the output.

Next, **Runs 2.3.1.1 to 2.3.4.2** modify the accuracy for the TrackIR 5 sensor. Changing the accuracy of this sensor also changes the baseline performance of the sensor. The changing baseline performance of the TrackIR 5 in Configuration 0 is given in Table 7.20. Here, the first number in the run indicator, in this case 2 is replaced by the baseline run indicator 0. Thus every run that is labeled 0.3.x in Table 7.20 is the baseline performance of the TrackIR 5 in the aforementioned noise profile.

The TrackIR 5 performance by itself in Runs 0.3.1, 0.3.2, 0, 0.3.3, and 0.3.4, where in each the accuracy is improved by a factor of 10, i.e. the actual SD on the measurements decreases by $90\,\%$. Run 0.3.1 to Run 0.3.2 result in the expected decrease of $90\,\%$ in pose error across the board. The error decreases close to the expected value of $90\,\%$ from Run 0.3.2 to Run 0, averaging around $88\,\%$. However, comparing Runs 0 and 0.3.3 the error only decrease $48\,\%$ and $66\,\%$ for the average total position and attitude error respectively. The variability on the error improves even less in this comparison, $1\,\%$ and $23\,\%$ for the position and attitude respectively. Finally, comparing Runs 0.3.3 to 0.3.4 the error decreases only percentage points. Moreover, the error variance even increases.

This observed behavior, the declining rate of improvement, is not unlike the behavior observed in Section 7.2.4. The performance of the TrackIR 5 is computed at 600 Hz in order to compare the other values in Table 7.20. This is also the source of the declining rate of improvement, the TrackIR 5 sensor measures the pose but until the next measurement the actual object will have moved. The subsequent error is computed five times, i.e. at 600 Hz, per measurement. This source of error stays constant run to run and is the reason for the non-linear error response.

Comparing Runs 2.3.1.1, 2.3.2.1, 2, 2.3.3.1, and 2.3.4.1 to another, i.e. each run the TrackIR 5 accuracy improves by a factor of 10 and is taken into account, reveals the position improving less while the attitude improves more each run. The mean positional error decrease by $85\,\%$, $84\,\%$, $82\,\%$, and $63\,\%$. While the mean total attitude error decreases by $25\,\%$, $70\,\%$, $71\,\%$, $61\,\%$. Furthermore, the improvements of the SD and max total error are in the same range for the position. This is not true for Run 2.3.4.1 where the SD and max total error decreased less when compared to the mean percentage wise. Also noteworthy is the comparison between Runs 2.3.1.1 and 2.3.2.1, here the attitude SD and max error increases by around $10\,\%$.

Even though the TrackIR sensor by itself in Run 0.3.4 started to converge on a maximum performance for the motion at hand. Including the IMU of the HMD to the system pushed out the aforementioned maximal performance as it provided updates in between observation improving the HMD vehicle-fixed pose estimation performance of the system.

Analyzing Runs 2.3.1.1 to 2.3.4.2 from the SRS point of view in Table 7.22. The error on the SRS pose does not increase when the expected SD on the TrackIR is set correctly, i.e. in Runs 2.3.x.1. Moreover, in Run 2.3.4.1, i.e. the accuracy of the TrackIR 5 improves by a factor of 100, the estimation accuracy on the SRS improves notably. Indicating that the certainty on the HMD pose has improved enough for the TrackIR 5 readings extrapolate through to the SRS pose estimation by state covariance $\boldsymbol{P}_{\hat{x}}$.

In **Run 2.4** the computational frequency of the overall system was reduced from 600 Hz to 100 Hz. A similar pattern to Run 1.5 can be observed where the computational frequency was also reduced. Again here the performance is still computed at 600 Hz in order to make comparisons to the other runs in Tables 7.20 and 7.22. Comparing the HMD and the SRS pose errors in Tables 7.20 and 7.22 of Run 2.4, the accuracy of the SRS seemingly deteriorates more, $\approx \times 25$, than the HMD, $\approx \times 3 - 5$, when compared against Run 2. This statement is also true for Run 1.5. The SRS accuracy seems to deteriorate more due to the accuracy original achieved in combination with the error computed at 600 Hz.

Increasing the sample rate of the SRS IMU improves the SRS pose as expected in **Run 2.5**. Increasing the sample rate also reduces the forward Euler induced integration error on the SRS IMU, reducing the accelerometer and gyroscope expected SD scaling to 3.0 and 1.1 respectively, according to Section 7.2.4. Interestingly, increasing the sample rate improves the accuracy more than improving the sensor itself by a factor of 10 as shown in Run 2.1.2.1. The improvements on the accuracy of the SRS subsystem improve the HMD vehicle-fixed pose error, found in Table 7.20. The HMD pose improvement are the result of the combination of the increased certainty on the SRS estimated state and accuracy in general.

Increasing the sample rate this run of the TrackIR 5 result in two data point in Table 7.20, **Runs 0.6 and 2.6**. Run 0.6 is the result of Configuration 0, while, Run 2.6 is the result of Configuration 2. As expected, increasing the sample-rate of the TrackIR 5 sensors would improve the HMD vehicle-fixed pose error when comparing Runs 0 to 0.6 and 2 to 2.6. The improvements on the accuracy of the SRS where small if any except on both the max positional error and attitude error improving by $2\,\%$ and $15\,\%$ respectively.

As hinted at by comparing the reduction in average error of the HMD's vehicle-fixed pose average error between the baseline and Configuration 1 with different motion profiles, the IMU contribution is increased due to the faster and larger motion profile. This is again confirmed by comparing the comparison between Run 0 to Run 2 to the comparison between **Run 0.7 to 2.7** in Table 7.20. However, the contribution of the SRS's IMU in case of the increased motion profile in Run 2.7 in Table 7.22 is clear when compared to Run 1.6 in Table 7.19. In Run 2.7 the increase in motion causes the SRS's pose average error to only almost double.

Table 7.20: The error of each Configuration 2 modified run in terms of the vehicle-fixed position and attitude of the HMD.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}
2	1.93×10^{-4}	9.56×10^{-5}	5.80×10^{-4}	9.43×10^{-3}	4.78×10^{-3}	2.47×10^{-2}
2.1.1.1	2.32×10^{-4}	1.12×10^{-4}	8.42×10^{-4}	1.18×10^{-2}	5.92×10^{-3}	5.01×10^{-2}
2.1.1.2	2.33×10^{-4}	1.12×10^{-4}	8.34×10^{-4}	1.18×10^{-2}	5.91×10^{-3}	4.98×10^{-2}
2.1.2.1	1.91×10^{-4}	9.54×10^{-5}	5.76×10^{-4}	9.35×10^{-3}	4.82×10^{-3}	2.49×10^{-2}
2.1.2.2	1.92×10^{-4}	9.55×10^{-5}	5.78×10^{-4}	9.39×10^{-3}	4.83×10^{-3}	2.49×10^{-2}
2.2	2.34×10^{-4}	1.12×10^{-4}	6.53×10^{-4}	9.41×10^{-3}	5.15×10^{-3}	2.62×10^{-2}
0.3.1	5.36×10^{-2}	2.88×10^{-2}	2.01×10^{-1}	8.72	3.94	2.75×10^{1}
2.3.1.1	7.62×10^{-3}	3.80×10^{-3}	1.88×10^{-2}	4.14×10^{-2}	1.31×10^{-2}	6.86×10^{-2}
2.3.1.2	1.53×10^{-2}	8.83×10^{-3}	5.28×10^{-2}	7.46×10^{-1}	3.56×10^{-1}	1.84
0.3.2	5.38×10^{-3}	2.88×10^{-3}	2.01×10^{-2}	8.74×10^{-1}	3.94×10^{-1}	2.75
2.3.2.1	1.19×10^{-3}	5.27×10^{-4}	3.11×10^{-3}	3.11×10^{-2}	1.42×10^{-2}	7.70×10^{-2}
2.3.2.2	1.55×10^{-3}	8.80×10^{-4}	5.34×10^{-3}	7.46×10^{-2}	3.66×10^{-2}	1.90×10^{-1}
0.3.3	4.58×10^{-4}	3.62×10^{-4}	1.78×10^{-3}	4.60×10^{-2}	3.68×10^{-2}	1.44×10^{-1}
2.3.3.1	3.51×10^{-5}	1.71×10^{-5}	1.25×10^{-4}	2.77×10^{-3}	1.33×10^{-3}	8.21×10^{-3}
2.3.3.2	1.01×10^{-4}	4.64×10^{-5}	2.55×10^{-4}	5.73×10^{-3}	2.13×10^{-3}	1.24×10^{-2}
0.3.4	4.46×10^{-4}	3.72×10^{-4}	1.77×10^{-3}	4.36×10^{-2}	3.84×10^{-2}	1.35×10^{-1}
2.3.4.1	1.30×10^{-5}	9.96×10^{-6}	9.52×10^{-5}	1.09×10^{-3}	6.03×10^{-4}	4.62×10^{-3}
2.3.4.2	9.91×10^{-5}	4.39×10^{-5}	2.43×10^{-4}	5.62×10^{-3}	2.03×10^{-3}	1.19×10^{-2}
2.4	5.58×10^{-4}	3.44×10^{-4}	2.05×10^{-3}	4.65×10^{-2}	3.56×10^{-2}	1.44×10^{-1}
2.5	1.70×10^{-4}	9.59×10^{-5}	7.62×10^{-4}	6.25×10^{-3}	2.43×10^{-3}	1.30×10^{-2}
0.6	5.82×10^{-4}	2.87×10^{-4}	1.87×10^{-3}	9.16×10^{-2}	4.12×10^{-2}	2.75×10^{-1}
2.6	1.33×10^{-4}	6.22×10^{-5}	3.78×10^{-4}	6.43×10^{-3}	2.61×10^{-3}	1.75×10^{-2}
0.7	1.99×10^{-3}	1.41×10^{-3}	7.57×10^{-3}	2.09×10^{-1}	1.37×10^{-1}	6.84×10^{-1}
2.7	4.83×10^{-4}	1.98×10^{-4}	1.27×10^{-3}	9.59×10^{-3}	4.44×10^{-3}	2.54×10^{-2}

Table 7.21: Condition overview of the modified runs of Configuration 2, see Section 7.4.1 for the explanation on the colors.

	NEES					
	SRS			HMD		
Run	Position	Attitude	Lin. Enc. Inno.	Position	Attitude	TrackIR 5 Inno.
2	26.40	15.92	26.59	3.90	3.79	5.18
2.1.1.1	16.68	10.47	26.36	3.86	3.75	7.00
2.1.1.2	32.97	40.90	740.90	3.92	3.76	9.71
2.1.2.1	28.67	17.96	26.60	3.90	3.75	5.17
2.1.2.2	26.28	15.60	19.47	3.90	3.78	5.15
2.2	3910.63	1184.91	702.97	13.60	3.52	6.07
2.3.1.1	26.24	15.82	26.57	3.08	1.09	5.11
2.3.1.2	80.91	59.03	43.94	21 097.55	22 004.89	49 857.31
2.3.2.1	26.23	15.82	26.58	3.01	2.47	5.10
2.3.2.2	27.09	16.46	26.82	214.26	220.68	498.50
2.3.3.1	26.70	15.78	26.18	9.04	2.86	6.76
2.3.3.2	26.38	15.90	26.61	1.65	1.49	0.28
2.3.4.1	24.69	13.44	24.65	20.13	3.28	10.54
2.3.4.2	26.37	15.89	26.59	1.62	1.45	0.24
2.4	4.32	3.70	7.25	5.51	2.27	5.42
2.5	48.88	8.02	17.26	3.63	1.64	5.00
2.6	26.35	15.81	26.70	3.53	2.44	5.06
2.7	5.46	4.71	24.51	19.31	3.94	8.61

Table 7.22: The error of each Configuration 2 modified run in terms of the inertial position and attitude of the SRS.

	SRS					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
2	2.98×10^{-5}	1.71×10^{-5}	9.23×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.54×10^{-3}
2.1.1.1	2.55×10^{-5}	1.41×10^{-5}	7.96×10^{-5}	6.31×10^{-3}	5.31×10^{-3}	3.78×10^{-2}
2.1.1.2	3.31×10^{-5}	1.94×10^{-5}	1.19×10^{-4}	6.41×10^{-3}	5.23×10^{-3}	3.83×10^{-2}
2.1.2.1	3.07×10^{-5}	1.72×10^{-5}	9.40×10^{-5}	6.68×10^{-4}	3.34×10^{-4}	1.93×10^{-3}
2.1.2.2	2.98×10^{-5}	1.70×10^{-5}	9.16×10^{-5}	6.19×10^{-4}	3.18×10^{-4}	1.86×10^{-3}
2.2	1.89×10^{-4}	1.10×10^{-4}	5.64×10^{-4}	3.54×10^{-3}	2.23×10^{-3}	1.12×10^{-2}
2.3.1.1	2.98×10^{-5}	1.71×10^{-5}	9.15×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.53×10^{-3}
2.3.1.2	5.21×10^{-5}	2.92×10^{-5}	1.81×10^{-4}	1.30×10^{-3}	6.88×10^{-4}	6.16×10^{-3}
2.3.2.1	2.98×10^{-5}	1.71×10^{-5}	9.13×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.55×10^{-3}
2.3.2.2	3.03×10^{-5}	1.72×10^{-5}	9.59×10^{-5}	9.37×10^{-4}	5.53×10^{-4}	4.45×10^{-3}
2.3.3.1	2.92×10^{-5}	1.68×10^{-5}	9.28×10^{-5}	9.20×10^{-4}	5.53×10^{-4}	4.54×10^{-3}
2.3.3.2	2.98×10^{-5}	1.71×10^{-5}	9.18×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.54×10^{-3}
2.3.4.1	2.70×10^{-5}	1.53×10^{-5}	8.67×10^{-5}	8.72×10^{-4}	5.44×10^{-4}	4.53×10^{-3}
2.3.4.2	2.98×10^{-5}	1.71×10^{-5}	9.18×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.54×10^{-3}
2.4	8.35×10^{-4}	6.53×10^{-4}	3.04×10^{-3}	2.23×10^{-2}	1.68×10^{-2}	8.14×10^{-2}
2.5	2.63×10^{-5}	1.48×10^{-5}	7.46×10^{-5}	6.33×10^{-4}	3.20×10^{-4}	2.21×10^{-3}
2.6	2.98×10^{-5}	1.70×10^{-5}	9.09×10^{-5}	9.37×10^{-4}	5.62×10^{-4}	3.85×10^{-3}
2.7	3.81×10^{-5}	2.59×10^{-5}	1.53×10^{-4}	1.77×10^{-3}	1.77×10^{-3}	1.04×10^{-2}

7.5.3. Configuration 3: SRS's Linear Encoders and Setpoints + HMD's Visual Tracking and IMU

The sensitivity analysis of Configuration 3 focusses on the setpoint integration to improve the vehicle-fixed pose estimation of the HMD. First, the impact of changing the expected variance on the lag $\underline{\boldsymbol{\sigma}}_{\mathcal{T}}$ is investigated, followed by looking into the influence of the acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and the angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$ variances.

This configuration is not as robust to initial conditions as its predecessors, i.e. Configurations 1 and 2. To establish the sensitivity of Configuration 3 to initial conditions, the initial state $\hat{\underline{\boldsymbol{x}}}_0$ as well as the initial state covariance $\boldsymbol{P}_{\hat{x}_0}$ were altered and tested. The initial expected lag $\hat{\underline{\tau}}$ for run 3.3 was varied from 0.0001 to 1.00, testing how well the configuration deals with $\div 100$ to $\times 100$ wrong initial estimation of said lag. Moreover, the initial random error of the position, velocity, attitude and angular velocity were set to similar initial errors to Run 1.4. Furthermore, the initial lag covariance of the lag was varied to see the impact on the initial convergence and robustness. Next, the SUT parameters were varied in run 3.7.1 to 3.10.3, as this configuration is the most sensitive to initial conditions. Last, the motion profile itself was changed.

Configuration 3

- **Process noise Q**
  - 3.1 SRS: lag $\underline{\boldsymbol{\sigma}}_{\mathcal{T}}$
    - 3.1.1 Increase expected SD: $\times 10$
    - 3.1.2 Decrease expected SD: $\div 10$
  - 3.2 SRS: acceleration $\underline{\boldsymbol{\sigma}}_{\ddot{c}_P}$ and angular acceleration $\underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}$
    - 3.2.1 Increase expected SD: $\times 10$
    - 3.2.2 Decrease expected SD: $\div 10$
- **Initial state $\hat{\underline{\boldsymbol{x}}}_0$**
  - 3.3 Set initial lag
    - 3.3.1 $\forall = 0.0001$

3.3.2 $\forall = 0.001$

3.3.3 $\forall = 0.01$

3.3.4 $\forall = 0.10$

3.3.5 $\forall = 1.0$

3.3.6 $\forall = 10.0$

3.4 Increase error position, velocity, attitude, and angular velocity

3.4.1 Similar to Run 1.4

3.4.2 Half the initial error of Run 1.4

- **Initial state covariance $\boldsymbol{P}_{\hat{x}_0}$**

3.5 Scale variance lag

3.5.1 $\times 1000$

3.5.2 $\times 100$

3.5.3 $\times 10$

3.5.4 $\div 10$

3.5.5 $\div 100$

3.5.6 $\div 1000$

- **Computational frequency**

3.6 Decrease to 100 Hz

- **SUT parameters**

3.7 α

3.7.1 0.001

3.7.2 0.1

3.8 β

3.8.1 0

3.8.2 10

3.9 κ

3.9.1 1

3.9.2 10

- **Motion profile**

3.10 Double amplitude and frequency

Any failed run is indicated in the tables below by '-'. The reason for the failure will be indicated in the accompanying text.

In **Runs 3.1.1 and 3.1.2** the expected SD is scaled up or down respectively by a factor of 10. The translation and rotation estimated lag are plotted in Figures 7.36, 7.37, 7.38, and 7.39 over time. Comparing the lag in Figures 7.36 and 7.36 of the increased expected SD to Figures 7.30 and 7.31 the reduced time to converge onto the correct value is evident. Here it takes about 2 s for the algorithm to get close to the correct value.

Also in this simulation it was assumed for the actual lag to be constant over a period of time. Therefore, a larger expected SD on the lag, as in Run 3.1.1, could be preferred if the actual lag was changing over time. However, care should be taken that the expected lag SD should be small relative to the lag itself, so sampling one of the sigma-points with a lag almost or less than zero is not possible.

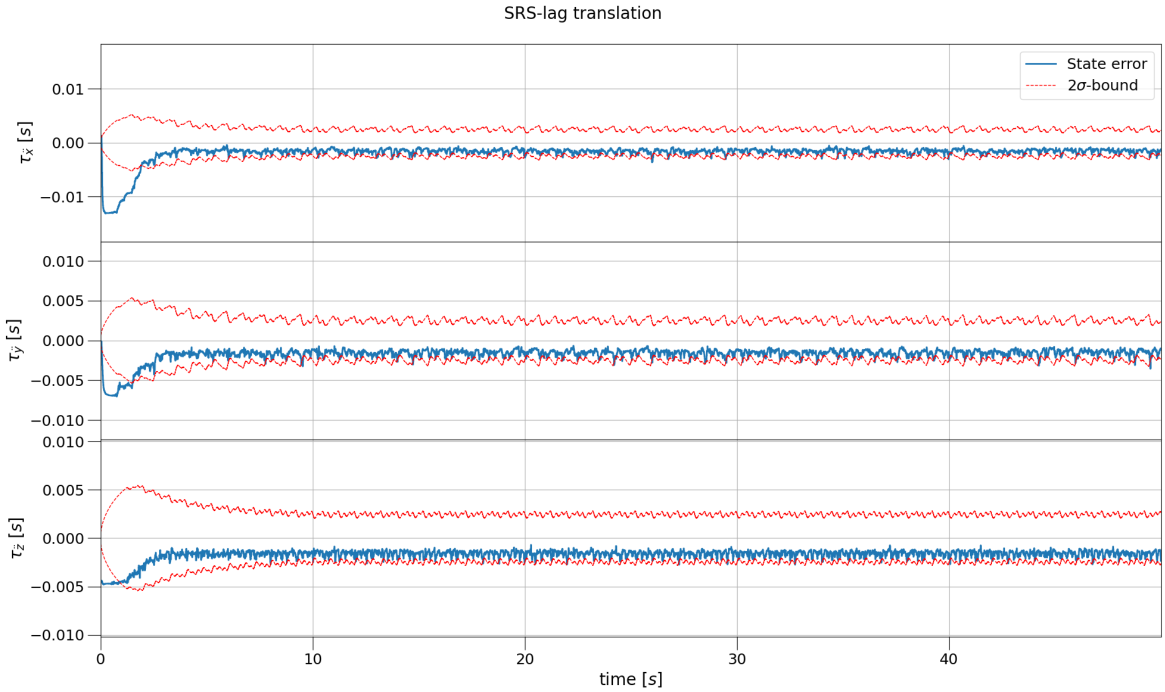


Figure 7.36: Sensitivity Run 3.1.1, the error on the translational lags the SRS per axis plotted against its covariance over time.

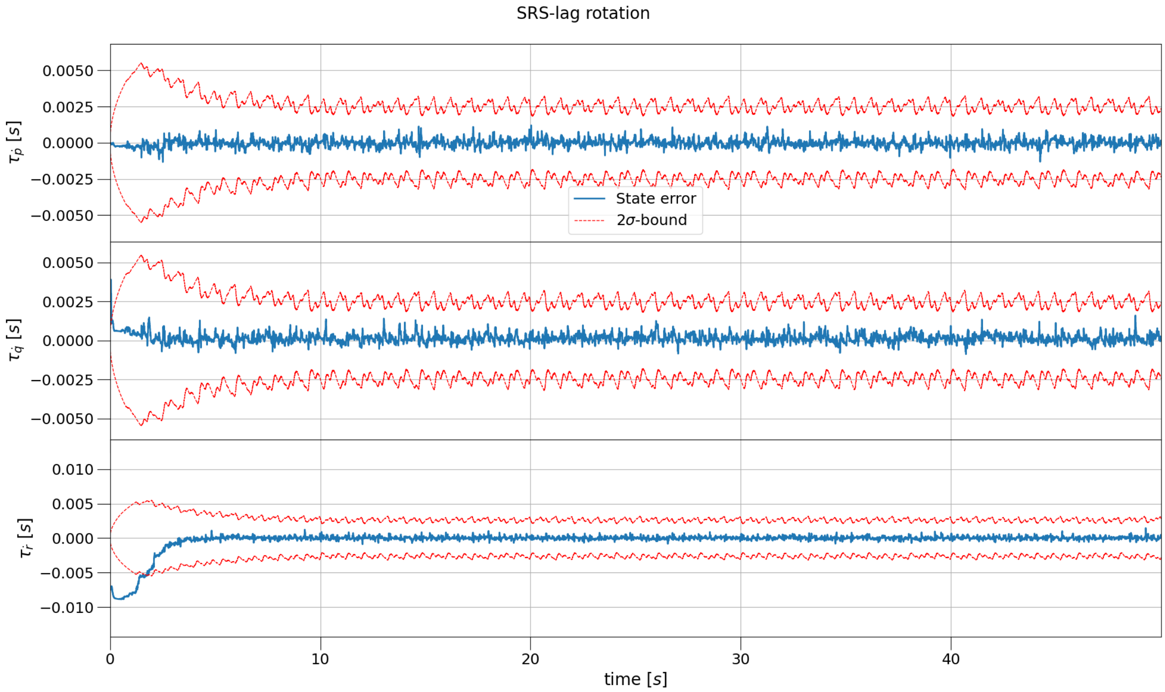


Figure 7.37: Sensitivity Run 3.1.1, the error on the rotational lags the SRS per axis plotted against its covariance over time.

Whereby Run 3.1.2 in Figures 7.38, and 7.39 take upward of 50 s to converge on to the correct value.

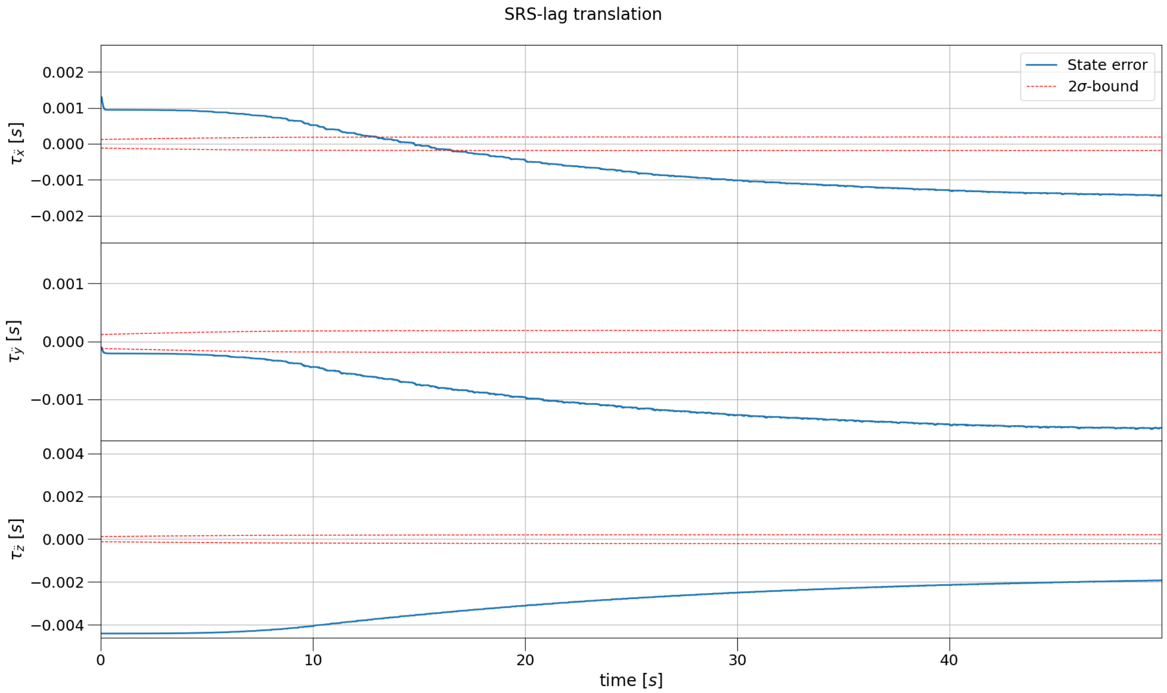


Figure 7.38: Sensitivity Run 3.1.2, the error on the translational lags the SRS per axis plotted against its covariance over time.

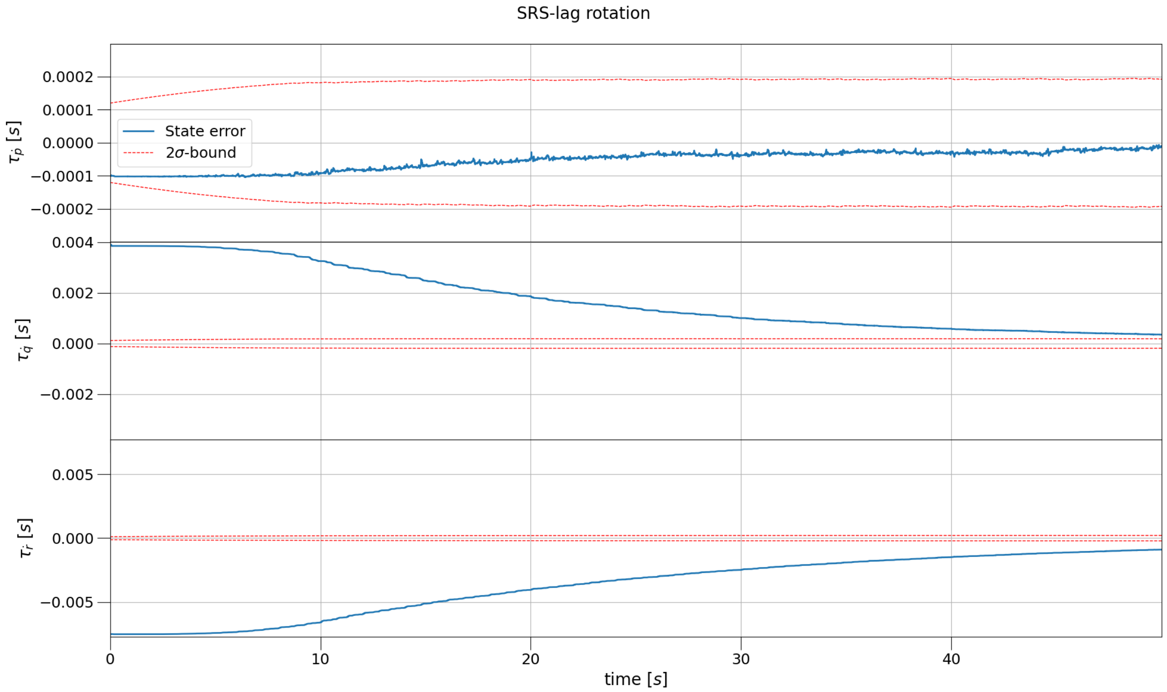


Figure 7.39: Sensitivity Run 3.1.2, the error on the rotational lags the SRS per axis plotted against its covariance over time.

The impact of the expected variance in the lag on the SRS pose in Table 7.25 and the vehicle-fixed HMD pose in Table 7.23 is minimal. This is expected as both Run 3.1.1 and Run 3.1.2 got close, eventually, to the actual lag.

Runs 3.2.1 and 3.2.2 investigate the impact of the expected variance on the acceleration and angular acceleration in a similar manner to Runs 1.1.1 and 1.1.2. The difference is that in this configuration the expected variance on the acceleration and angular acceleration is not the main source of input to the pose itself. As explained in Section 5.4, the expected variance on the acceleration and angular acceleration introduces uncertainty into the state covariance $\boldsymbol{P}_{\hat{x}}$. This in turn allows the algorithm to deal with the fact that the lag is an estimation and not an exact delay. Both runs decrease the accuracy on the SRS platform, hinting at the expected variance of the acceleration and angular acceleration in Run 3 being close to a local optimum, further evidenced by the NEES values in Table 7.24. The NEES values of Run 3 are already on the small side, indicating that the system might be more optimal with slightly less variance. Run 3.2.1 increase the expected variance, resulting in an even lower NEES value, even on the linear encoders innovation. Run 3.2.2 on the other hand, decreases the expected variance to much for this simulation. With NEES value well above the optimal 3 and, in the case of the linear encoders, 6. The innovation of the linear encoders can be a good indication of the expected variance in a real life system.

The impact of the expected variance on the acceleration and angular acceleration on the lag is of interest, in particular the convergence rate. Increasing the expected variance on the acceleration and angular acceleration reduced the converge rate of the lag, pictured in Figures 7.40 and 7.41. Increasing the variance, i.e. increasing the uncertainty, reduces the systems ability to find the lag. With the lag taking up to 40 s to converge.

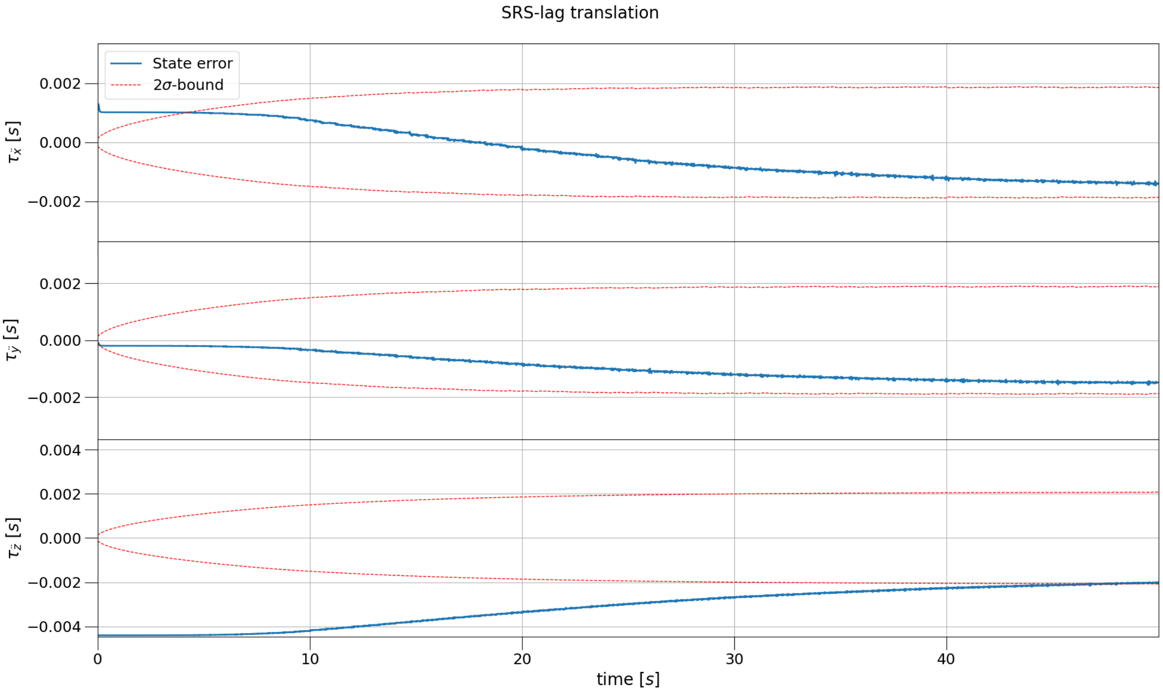


Figure 7.40: Sensitivity Run 3.2.1, the error on the translational lags the SRS per axis plotted against its covariance over time.

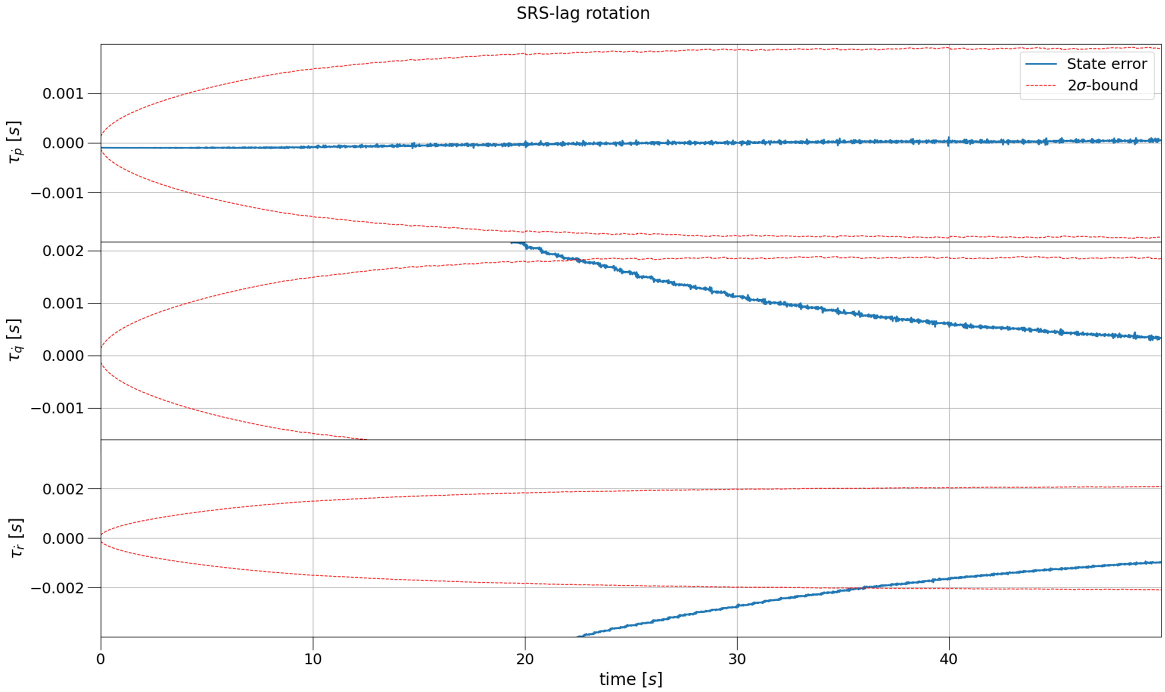


Figure 7.41: Sensitivity Run 3.2.1, the error on the rotational lags the SRS per axis plotted against its covariance over time.

Reducing the expected variance on the acceleration and angular acceleration has the opposite effect as depicted in Figures 7.42 and 7.43. The error on the lag is more volatile, possibly picking up other sensor errors within the system. In this configuration, the dynamics between the expected variances of the lag and the acceleration and angular acceleration are of note.

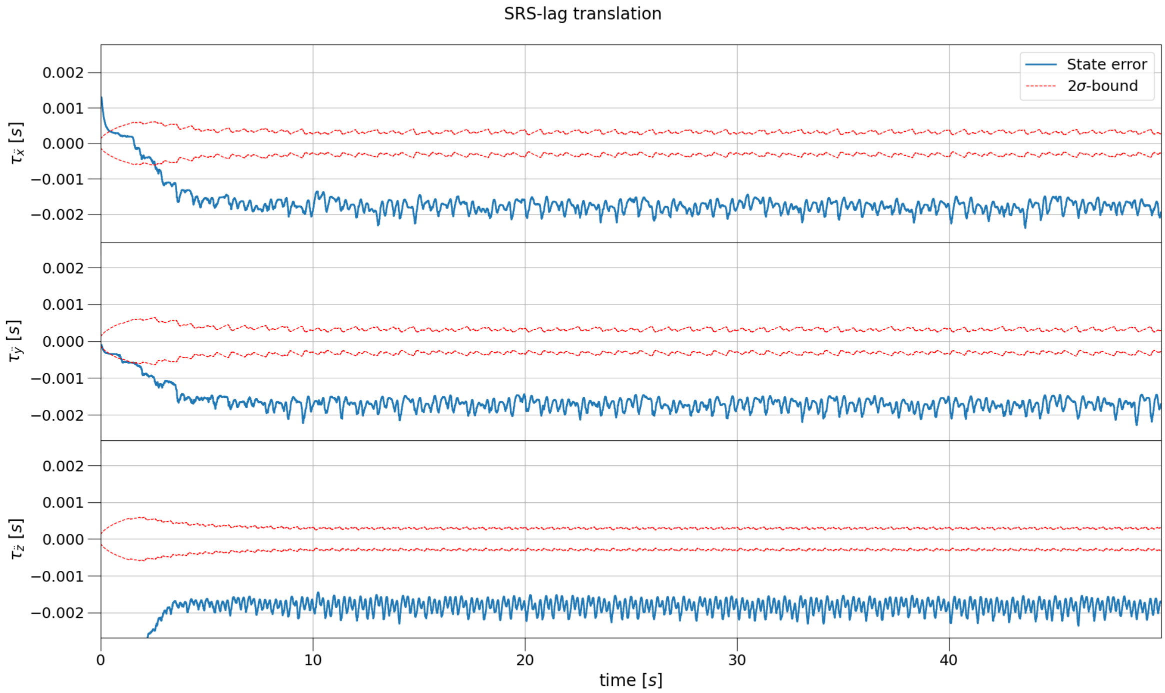


Figure 7.42: Sensitivity Run 3.2.2, the error on the translational lags the SRS per axis plotted against its covariance over time.

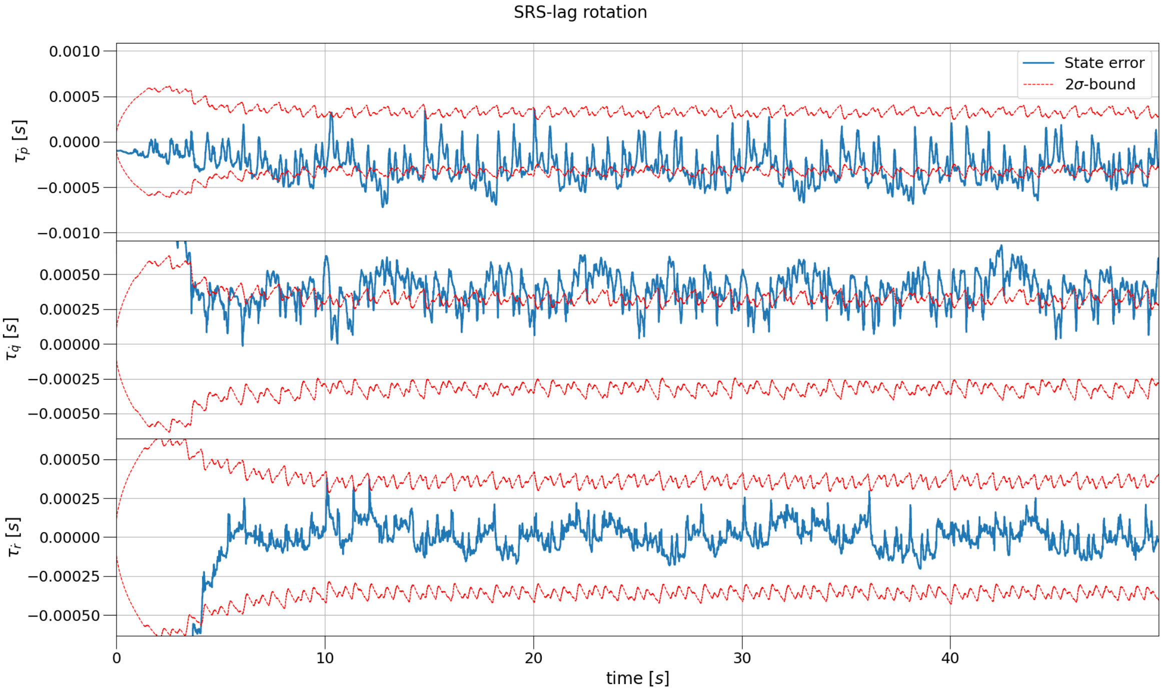


Figure 7.43: Sensitivity Run 3.2.2, the error on the rotational lags the SRS per axis plotted against its covariance over time.

Runs 3.3.x all modified the initial estimated lag in the state covariance $\boldsymbol{P}_{\hat{x}}$, ranging from 10^{-4} to 10^{1} and with the correct lag around 6×10^{-2}, found in Section 6.2.3.

The first run, Run 3.3.1, failed due to the lag getting smaller than zero and consequently causing the state covariance $\boldsymbol{P}_{\hat{x}}$ to be no longer positive definite, which is necessary for the Cholesky decomposition [10] and thus failing Equation 3.19. This failure is indicated in Tables 7.23, 7.24, and 7.25 as '-'.

The sixth run, Run 3.3.6, did not fail as none of its sigma-points samples lag estimates were smaller than zero. However, the system did not converge onto the correct lag due to scale difference between the error and the expected SD on the lag, and, therefore, the estimation failed. The errors and NEES values of this run are still presented in Tables 7.23, 7.24, and 7.25, but the run is considered a failure. Given this failure to estimate the correct lag, the accuracy presented in Table 7.25 is the result of the variance present on the acceleration and angular acceleration in combination with the accurate linear encoders. The impact on the vehicle-fixed pose error of the HMD is relatively small as the pose of the SRS only impact the measurement of the IMU of the HMD.

The estimated lags of Runs 3.3.2 and 3.3.5 are plotted in Figures 7.44, 7.45, 7.46, and 7.47. The lags of these two runs were plotted as they represented the two extremes simulated.

Interestingly, Run 3.3.2 takes roughly the same time to converge as Run 3 in Figures 7.30 and 7.31 even though the initial error was larger.

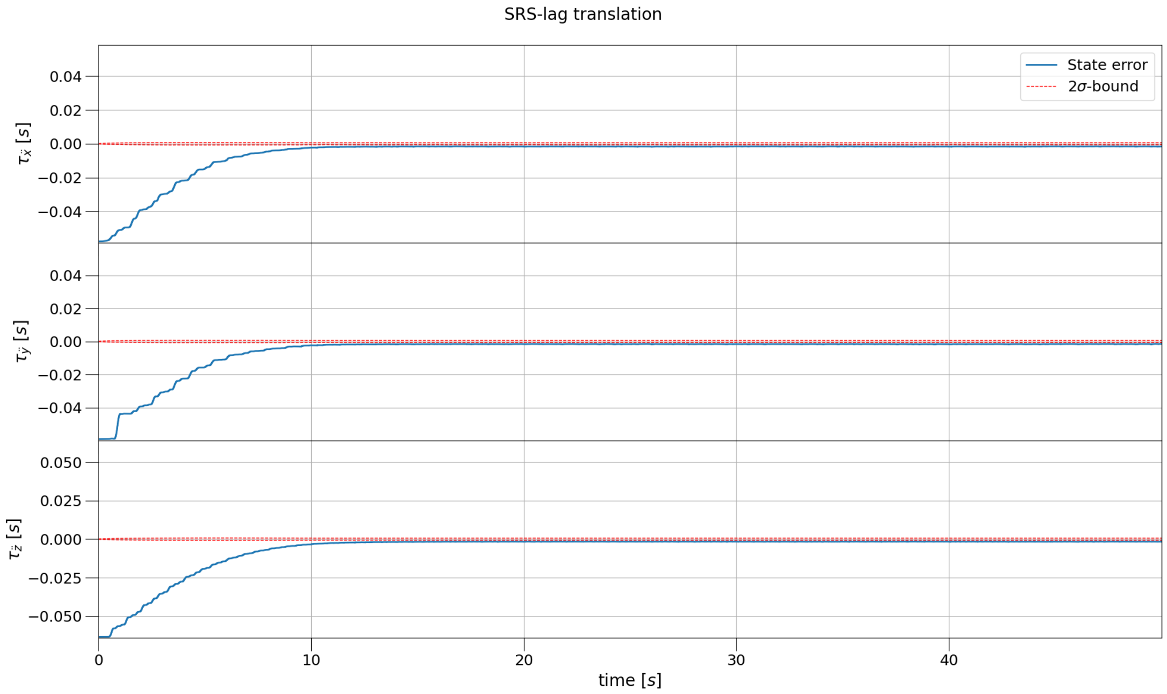


Figure 7.44: Sensitivity Run 3.3.2, the error on the translational lags the SRS per axis plotted against its covariance over time.

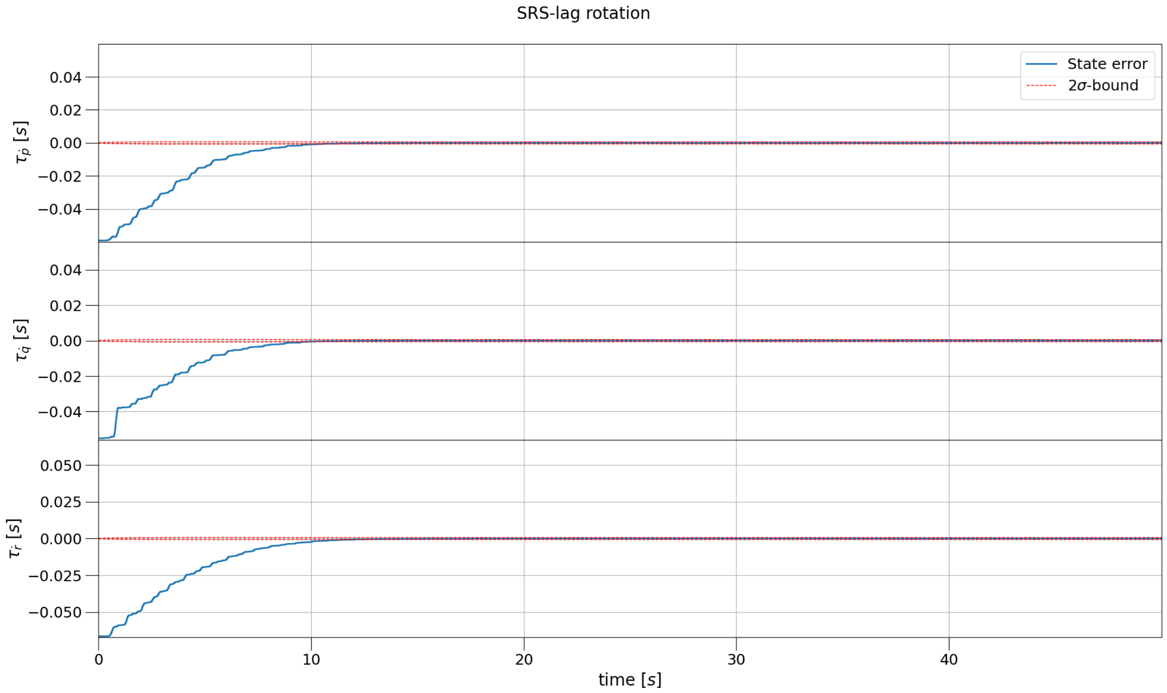


Figure 7.45: Sensitivity Run 3.3.2, the error on the rotational lags the SRS per axis plotted against its covariance over time.

Run 3.3.5 was getting close to converging onto the correct lag near the end of the run at 50 s. For this reason the run duration was extended in this case to 100 s. The convergence rate of the lag increase near the actual value, due to the nonlinear nature of the underlying model, i.e. Equations 5.20 and 5.21.

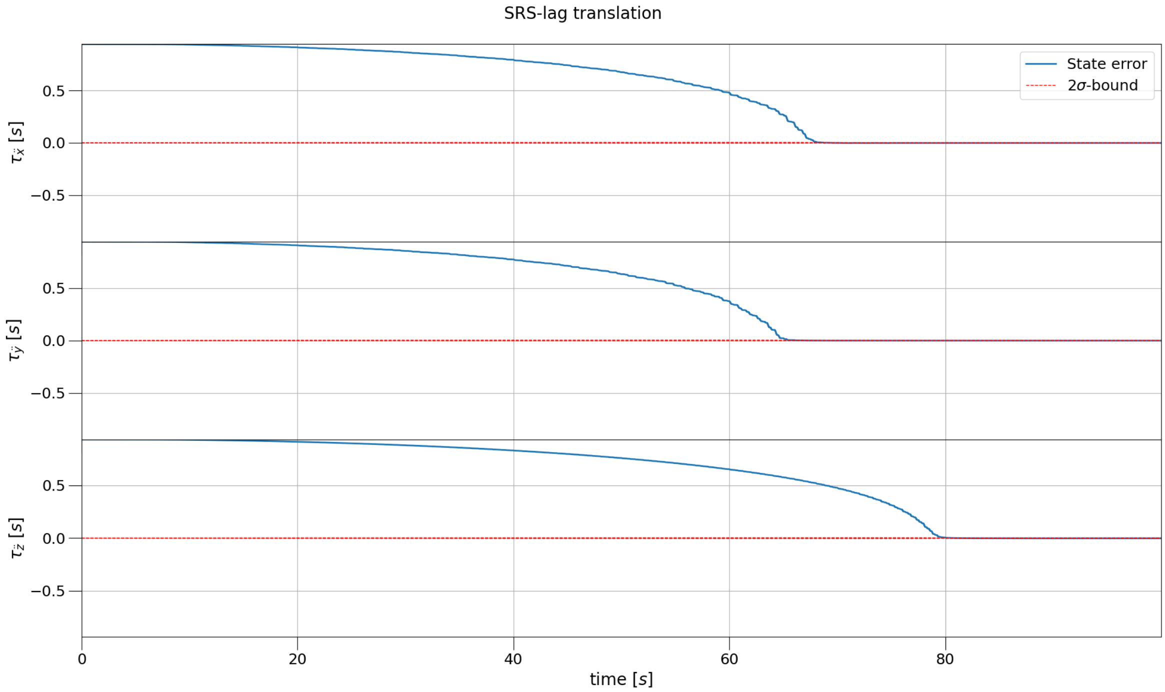


Figure 7.46: Sensitivity Run 3.3.5, the error on the translational lags the SRS per axis plotted against its covariance over time.

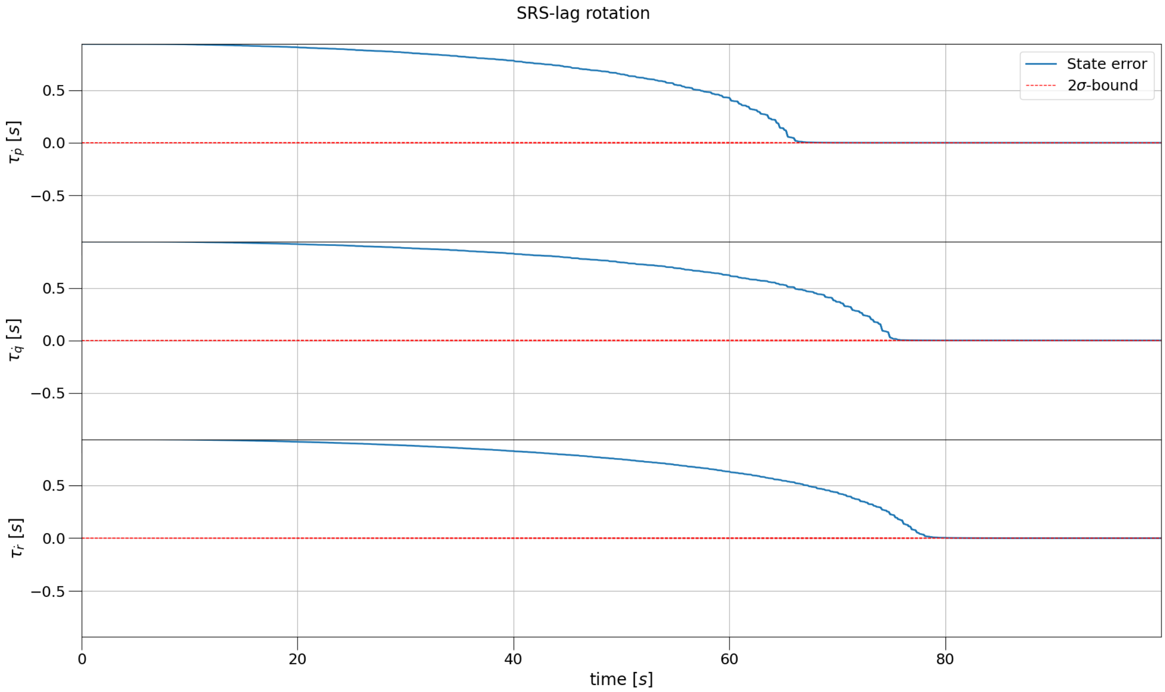


Figure 7.47: Sensitivity Run 3.3.5, the error on the rotational lags the SRS per axis plotted against its covariance over time.

In Figure 7.48 the innovation on the linear encoders of the SRS per hydraulic actuator plotted against its covariance over time. Here, the innovation starts to reduce onwards from 70 s to nominal values, relative to their covariance, from 80 s onward. This is the time-stamp at which all lags, plotted in Figures 7.46 and 7.47, are converged onto their actual value.

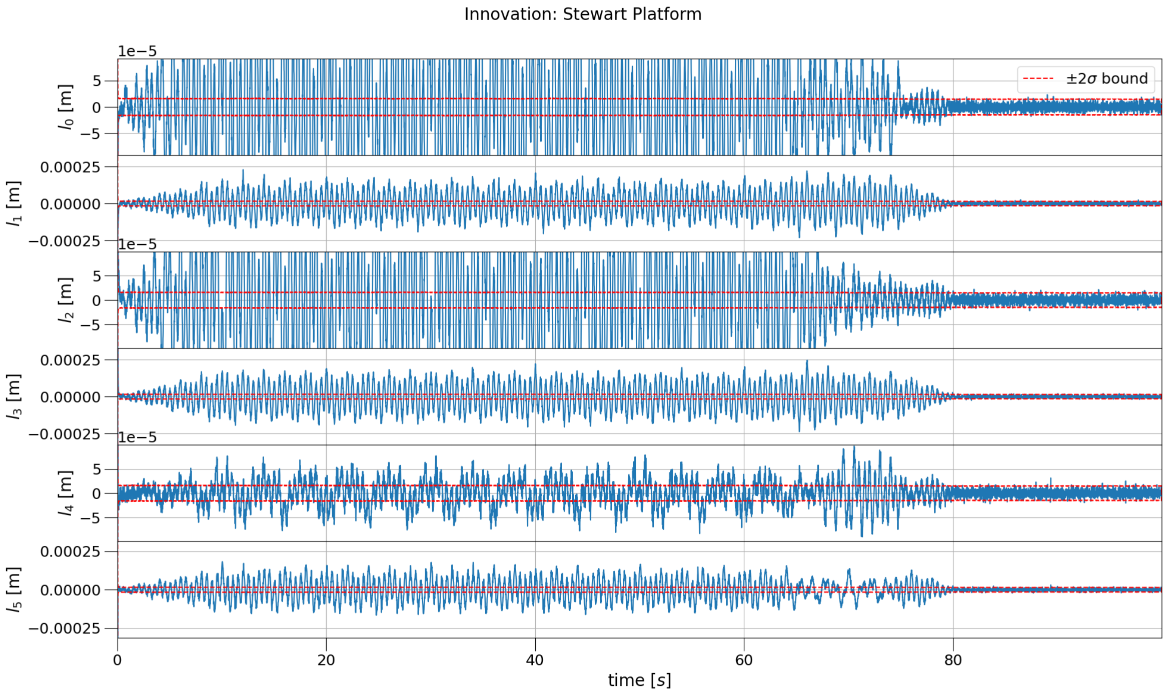


Figure 7.48: Sensitivity Run 3.3.5, the innovation on the linear encoders of the SRS per hydraulic actuator plotted against its covariance over time.

Run 3.4.1 the initial estimated state $\underline{\hat{\boldsymbol{x}}}_0$ is modified from Equation 7.74 to an initial error exact to that added in Equation 7.77, i.e.,

$$\underline{\hat{\boldsymbol{x}}}_0 = \begin{Bmatrix} \underline{\hat{\boldsymbol{c}}}_{P_0} \\ \underline{\hat{\dot{\boldsymbol{c}}}}_{P_0} \\ \underline{\hat{\ddot{\boldsymbol{c}}}}_{P_0} \\ \underline{\hat{\boldsymbol{e}}}_{P_0} \\ \underline{\hat{\boldsymbol{\omega}}}_{P_0} \\ \underline{\hat{\dot{\boldsymbol{\omega}}}}_{P_0} \\ \underline{\hat{\boldsymbol{\tau}}}_{\ddot{c}_0} \\ \underline{\hat{\boldsymbol{\tau}}}_{\dot{\omega}_0} \\ \underline{\hat{\boldsymbol{c}}}_{H_0} \\ \underline{\hat{\dot{\boldsymbol{c}}}}_{H_0} \\ \underline{\hat{\boldsymbol{e}}}_{H_0} \\ \underline{\hat{\boldsymbol{\lambda}}}_{H_0} \end{Bmatrix} = \left\{ \begin{matrix} \begin{pmatrix} 0.58682858 & -0.97092327 & -1.13218296 \end{pmatrix}^T \\ \begin{pmatrix} 1.22803562 & 0.46829199 & -2.37069145 \end{pmatrix}^T \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \\ \begin{pmatrix} 0.99364229 & 0.10614029 & -0.03567829 & -0.0116746 \end{pmatrix}^T \\ \begin{pmatrix} 0.12301066 & 0.71263982 & 0.9263408 \end{pmatrix}^T \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \\ \begin{pmatrix} 0.06 & 0.06 & 0.06 \end{pmatrix}^T \\ \begin{pmatrix} 0.06 & 0.06 & 0.06 \end{pmatrix}^T \\ \begin{pmatrix} -0.47145191 & -0.60514862 & -3.50374137 \end{pmatrix}^T \\ \begin{pmatrix} 1.2907381 & 0.99407434 & 1.91678943 \end{pmatrix}^T \\ \begin{pmatrix} 0.99311633 & -0.11389189 & 0.02575446 & -0.00923585 \end{pmatrix}^T \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \end{matrix} \right\} \tag{7.78}$$

Here, alongside increasing the initial estimation errors on the position and attitude, errors are introduced on the velocity and angular velocity. Again, these values are extreme and, in the case of the position of the HMD, physically impossible. But, this initial state $\underline{\hat{\boldsymbol{x}}}_0$ was chosen as such to investigate if the algorithm is mathematically robust to deal with them as such. The initial estimated SRS state had a random total positional error of 1.69 m, velocity error of 2.71 m/s, attitude error of $6.46°$, and angular rate of 67.33 °/s. Moreover, the initial estimate state of the HMD in this run had a random total error on the position of 0.92 m, velocity of 2.52 m/s, and attitude of $6.73°$.

Run 3.4.1 failed after 36 iteration due to a sigma-point sample lag dropping below zero. The algorithm trusts the linear encoders of the SRS such that it assumes that the error is on the input side, i.e. the lag variable.

This behavior is visible in **Run 3.4.2**, where the initial estimated errors halved and resulted in the new initial estimated state $\underline{\hat{\boldsymbol{x}}}_0$:

$$\underline{\hat{\boldsymbol{x}}}_0 = \begin{Bmatrix} \underline{\hat{\boldsymbol{c}}}_{P_0} \\ \underline{\hat{\dot{\boldsymbol{c}}}}_{P_0} \\ \underline{\hat{\ddot{\boldsymbol{c}}}}_{P_0} \\ \underline{\hat{\boldsymbol{e}}}_{P_0} \\ \underline{\hat{\boldsymbol{\omega}}}_{P_0} \\ \underline{\hat{\dot{\boldsymbol{\omega}}}}_{P_0} \\ \underline{\hat{\boldsymbol{\tau}}}_{\ddot{c}_0} \\ \underline{\hat{\boldsymbol{\tau}}}_{\dot{\omega}_0} \\ \underline{\hat{\boldsymbol{c}}}_{H_0} \\ \underline{\hat{\dot{\boldsymbol{c}}}}_{H_0} \\ \underline{\hat{\boldsymbol{e}}}_{H_0} \\ \underline{\hat{\boldsymbol{\lambda}}}_{H_0} \end{Bmatrix} = \begin{Bmatrix} \begin{pmatrix} 0.29341429 & -0.48546163 & -1.76109148 \end{pmatrix}^T \\ \begin{pmatrix} 0.61401781 & 0.234146 & -1.18534573 \end{pmatrix}^T \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \\ \begin{pmatrix} 0.99840931 & 0.0531547 & -0.01786757 & -0.0058466 \end{pmatrix}^T \\ \begin{pmatrix} 0.06150533 & 0.35631991 & 0.4631704 \end{pmatrix}^T \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \\ \begin{pmatrix} 0.06 & 0.06 & 0.06 \end{pmatrix}^T \\ \begin{pmatrix} 0.06 & 0.06 & 0.06 \end{pmatrix}^T \\ \begin{pmatrix} -0.23572596 & -0.30257431 & -3.25187068 \end{pmatrix}^T \\ \begin{pmatrix} 0.64536905 & 0.49703717 & 0.95839471 \end{pmatrix}^T \\ \begin{pmatrix} 0.9982776 & -0.0570442 & 0.01289945 & -0.00462589 \end{pmatrix}^T \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \end{Bmatrix} \tag{7.79}$$

This causes the initial state $\underline{\hat{\boldsymbol{x}}}_0$ of the SRS to have a total error on the position of 0.85 m, velocity of 1.36 m/s, attitude of $3.23°$, and angular rate of 33.67 °/s. Moreover, the HMD has a total error on the position of 0.46 m, velocity of 1.26 m/s, and attitude of $3.36°$.

In Figures 7.49 and 7.50 the error on the translational and rotational lags of the SRS are plotted per axis against their covariance over time. Here, the focus is on the initial error on the lag. Which, in the cases of $\tau_{\ddot{y}}$ and $\tau_{\dot{r}}$ reduce sharply due the initial overall state error. Furthermore, Tables 7.23, 7.24, and 7.25 show that once the algorithm overcomes the initial estimated state error, the impact is minimal.

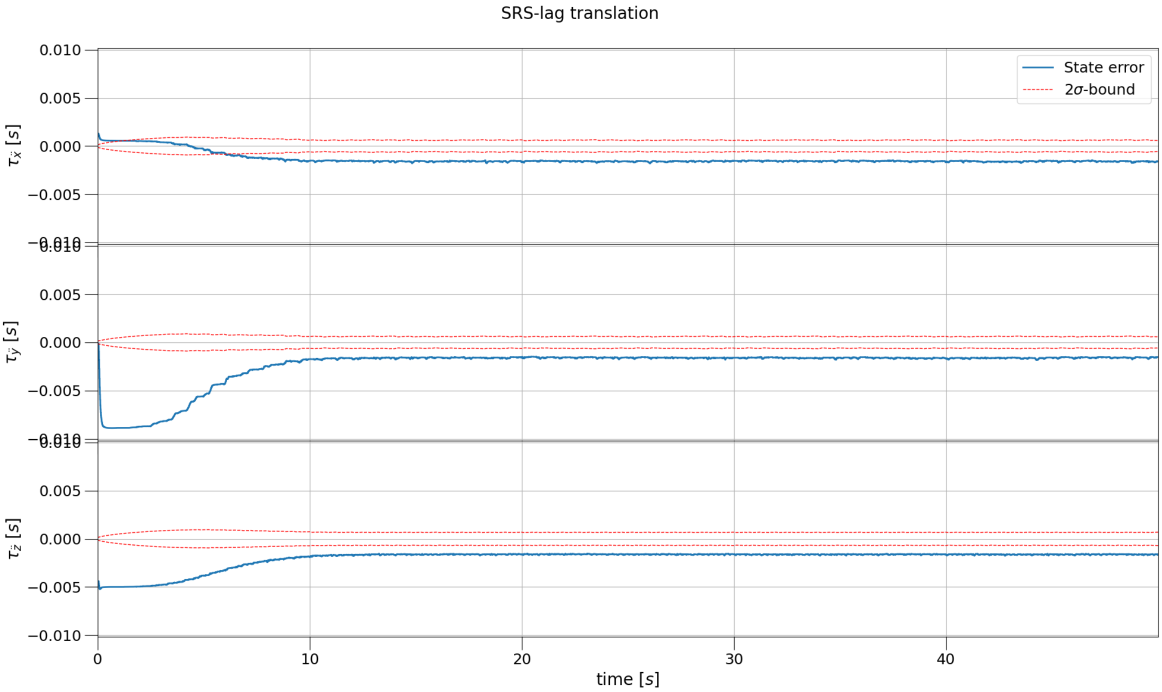


Figure 7.49: Sensitivity Run 3.4.2, the error on the translational lags the SRS per axis plotted against its covariance over time.

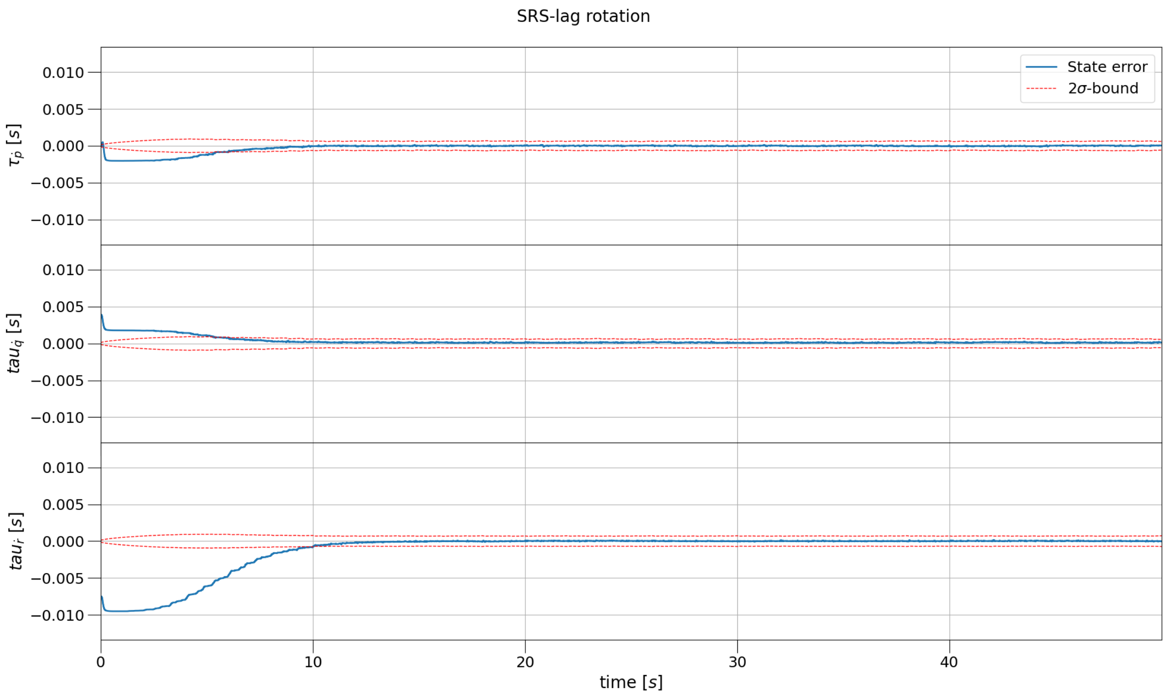


Figure 7.50: Sensitivity Run 3.4.2, the error on the rotational lags the SRS per axis plotted against its covariance over time.

In **Runs 3.5.x** the initial estimated lag variance, part of the state covariance $\boldsymbol{P}_{\hat{x}}$, is scaled ranging from 10^3 to 10^{-3}. Resulting in an estimated lag variance, ranging from 10^{-2} to 10^{-8}. These runs test the robustness of this configuration to initial conditions regarding the estimated lag on the setpoints. The first 2 runs fail, Run 3.5.1 fails after 24 iterations and Run 3.5.2 after 30 iterations. In both cases the failures were one of the lags being less than zero. All the failures in this configuration have in common that the failures occur after the linear encoder information have corrected the estimated state, i.e. an iteration count with a modulus of 6.

The two most extreme cases of Runs 3.5.x are highlighted by plotting there estimated lags of the SRS in Figures 7.51 to 7.54.

In case of Run 3.5.3, the initial excursion of the lag on the x-axis $\tau_{\ddot{x}}$ is larger than any in Run 3.4.2. As the error plot is centered around 0.0587 s, the -0.015 s excursion reduces the estimated lag by $25\,\%$. This explains how the lag of one of the sigma-point samples of Runs 3.5.1 and 3.5.2 end up below zero. Furthermore, the error and NEES values in Tables 7.23, 7.24, and 7.25 show that once the algorithm overcomes the initial estimated state error, the impact is minimal.

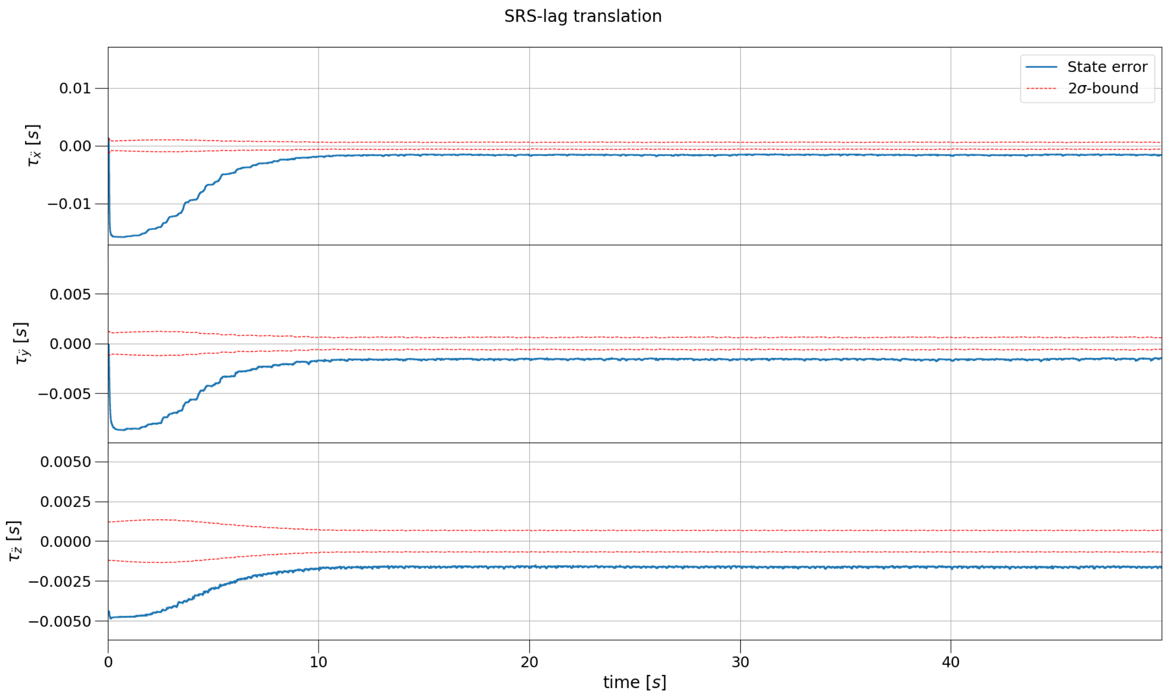


Figure 7.51: Sensitivity Run 3.5.3, the error on the translational lags the SRS per axis plotted against its covariance over time.

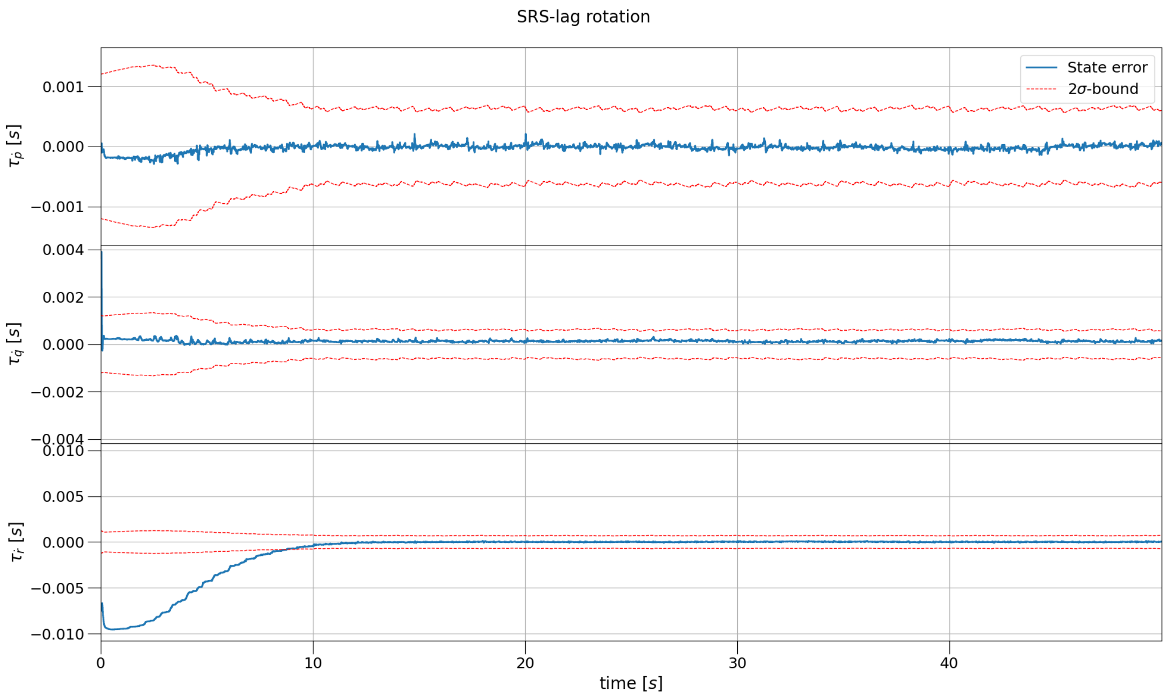


Figure 7.52: Sensitivity Run 3.5.3, the error on the rotational lags the SRS per axis plotted against its covariance over time.

The difference in the initial reaction of the estimated lag to the initial estimated error because of the reduced initial expected variance of the lag can be seen in Figures 7.53 and 7.54. Due to the small initial variance on the lag, the initial lag is almost constant in the initial iteration of the algorithm. Only when the expected variance, indicated by the red dotted lines, grows large enough, because of the influence of the process noise on the lag, the estimated lags start converging onto the actual value. This is also further helped by the fact that by the time the covariance on the lag increases the estimated state has converged. Causing smaller changes in the estimated state. Reducing the initial state covariance $\boldsymbol{P}_{\hat{x}_0}$ on sensitive states is a way to force the algorithm to estimate other states first.

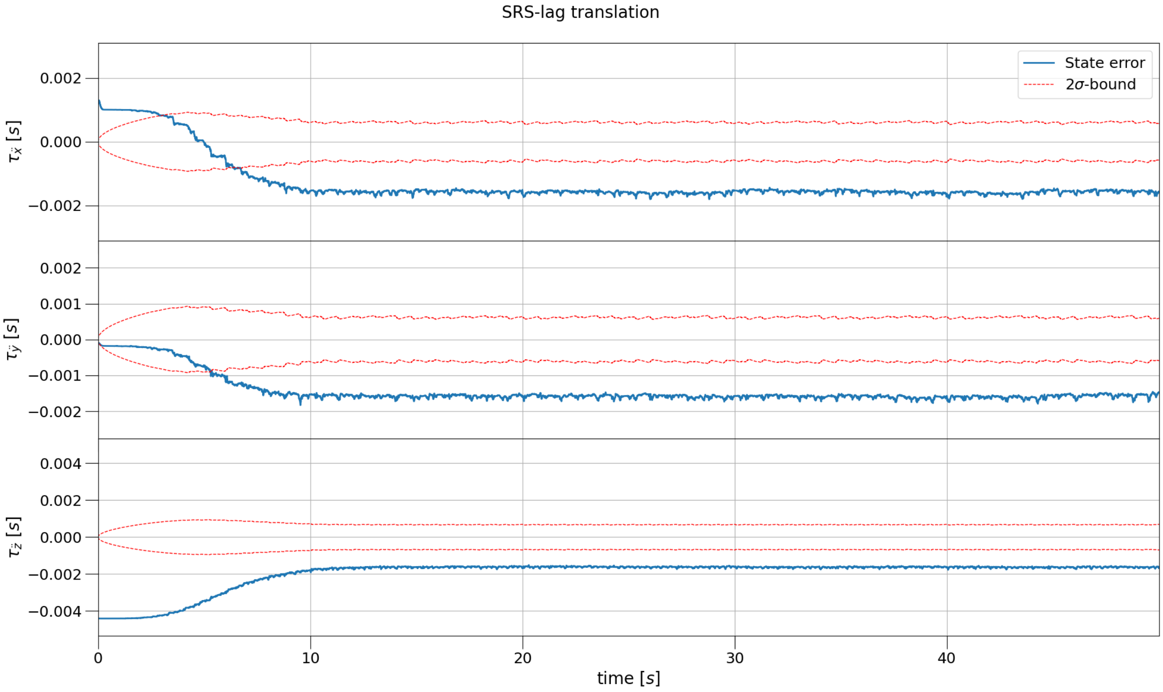


Figure 7.53: Sensitivity Run 3.5.6, the error on the translational lags the SRS per axis plotted against its covariance over time.

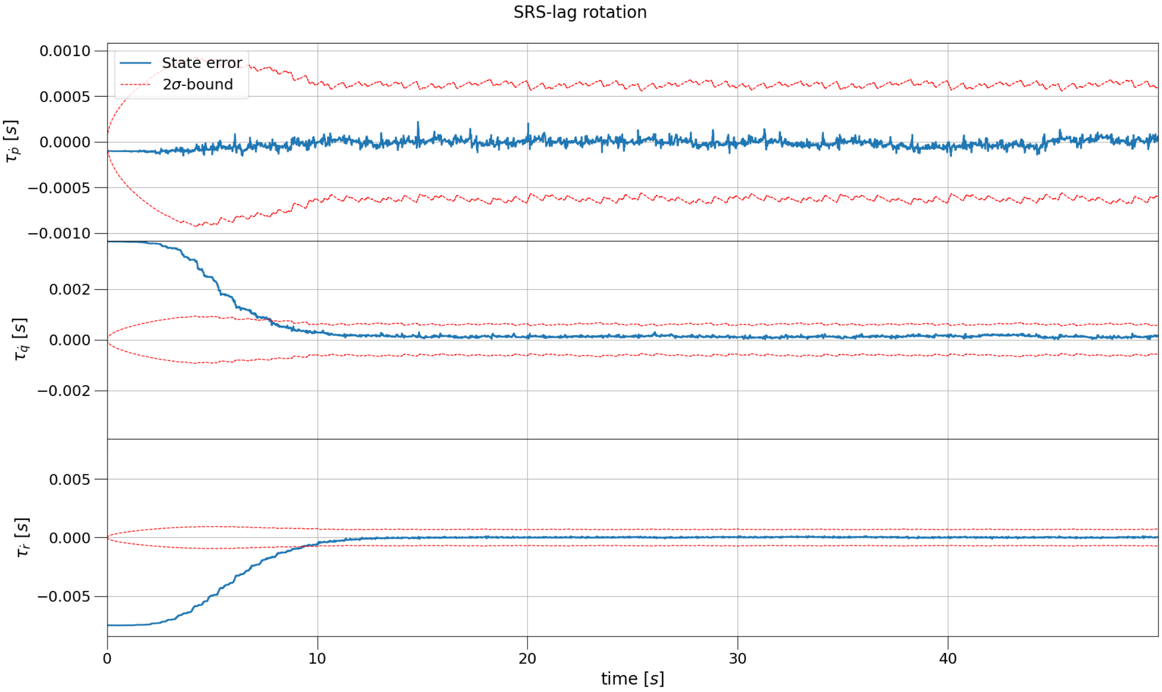


Figure 7.54: Sensitivity Run 3.5.6, the error on the rotational lags the SRS per axis plotted against its covariance over time.

Reducing the computational frequency to $100\,$Hz from $600\,$Hz in **Run 3.6**, just as in Runs 1.5 and

2.4, reduces the accuracy of the whole system. However, the estimated first order lags of the SRS subsystem are of note and shown in Figures 7.55 and 7.56.

The translational lag again converge onto -0.01, i.e. the negative time-step of the prediction. The sigma-bounds on both the translational and rotational lags are twice as large. As the sigma-bound are $\approx \sqrt{\boldsymbol{P}_{\hat{x}}}$, the covariance is roughly four times as large as in base Run 3.

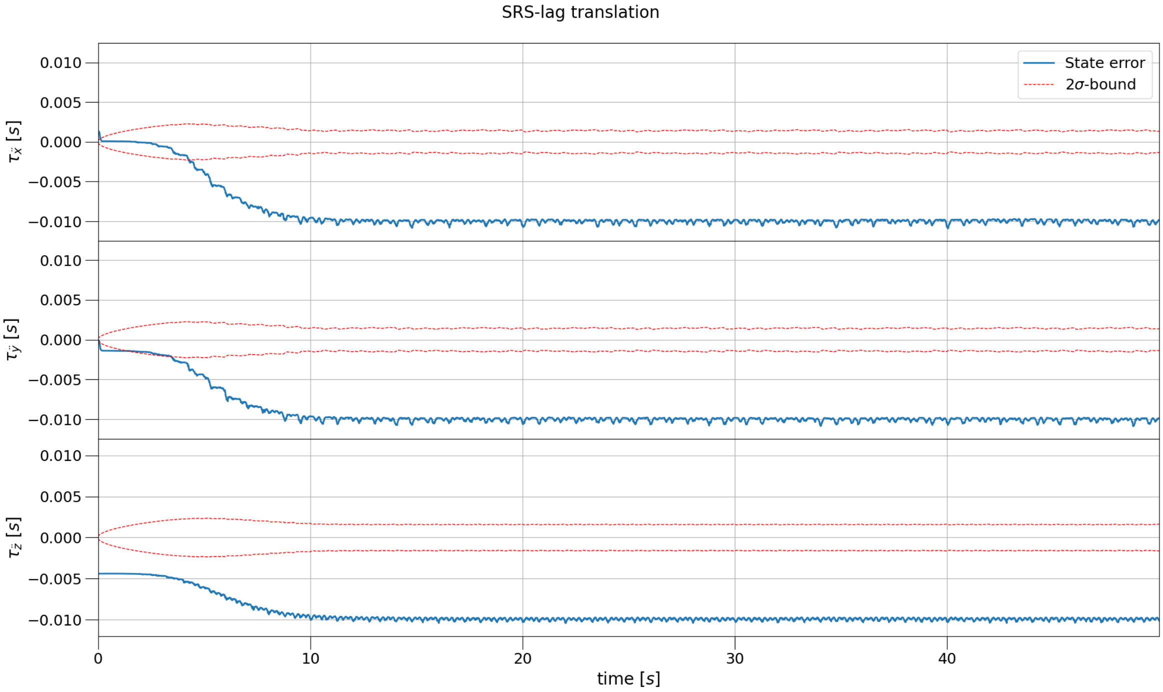


Figure 7.55: Sensitivity Run 3.6, the error on the translational lags the SRS per axis plotted against its covariance over time.

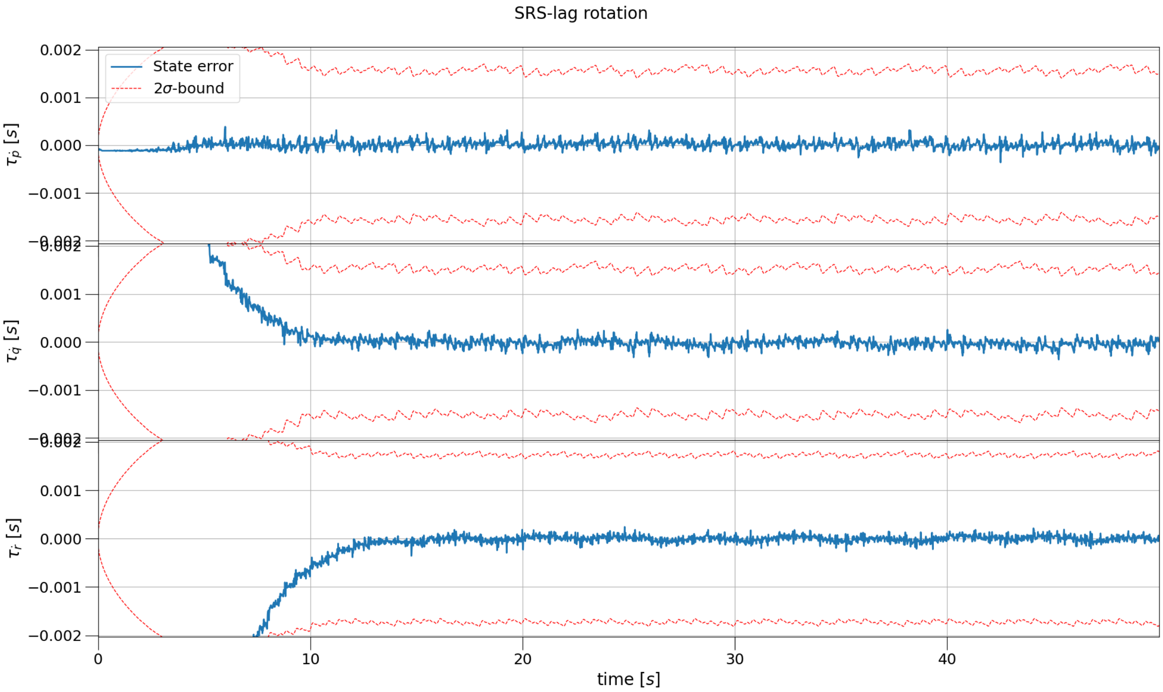


Figure 7.56: Sensitivity Run 3.6, the error on the rotational lags the SRS per axis plotted against its covariance over time.

Runs 3.7.1 to 3.9.2 focus on the impact of the sigma-point variables (α, β, κ). First α is varied in **Runs 3.7.1 and 3.7.2.** α scales the 'distance' between the sigma-point sample and the estimated state. As working the term, $L + \zeta$, scaling the state covariance $\boldsymbol{P}_{\hat{x}}$, in Equation 3.9, results in $\alpha^2 \left(L + \kappa\right)$.

Run 3.7.1, with α set to 10^{-3}, fails due to the state covariance matrix no longer being positive definite, necessary for the Cholesky decomposition [10] in Equation 3.19. Smaller α are preferred to capture local non-linearities into the sigma-points [41].

Increasing α to 10^{-1} in Run 3.7.2 causes no stability issues and even manages to slightly improve the accuracy by a couple of percentage points. Increasing α, i.e. increasing the distance to the sampled sigma-points, could increase the risk of sampling a negative lag. In this configuration, setting an α of 0.95 would make the algorithm sample negative lag sigma-point.

In **Runs 3.8.1 and 3.8.2** varied β, the sigma-point variable that only influences the computation of the covariance weights in Equation 3.7. β was varied from zero to ten, but the impact was small given the data in Tables 7.23, 7.24, and 7.25.

Runs 3.9.1 and 3.9.2 modified κ, the sigma-point variable responsible for guaranteeing positive semi-definiteness of the state covariance matrix $\boldsymbol{P}_{\hat{x}}$. If problems arise with positive-definiteness of the state covariance matrix, the value of κ could be increased to guarantee again said characteristic of the state covariance matrix $\boldsymbol{P}_{\hat{x}}$. For example, in the case of Run 3.7.1, setting κ to 10^3 resolved the positive-definiteness issue, with again similar performance to Run 3.

Run 3.10 scales the amplitudes and frequencies presented in Tables 7.1 and 7.2 in Section 7.2 by a factor of two. This causes the forces to be scaled by a factor of eight, deducted from Equation 7.23. Figures 7.57 and 7.58 show again that an increase in motion amplitude and or frequency causes the lag to converge faster. This is due to Equations 7.19 and 7.20, the equations for computing the accelerations from the setpoints. In these two equations, an acceleration of zero would cause the lag to become unobservable. The opposite is also true, larger forces cause the lag to become more observable and hence increase the lag convergence rate.

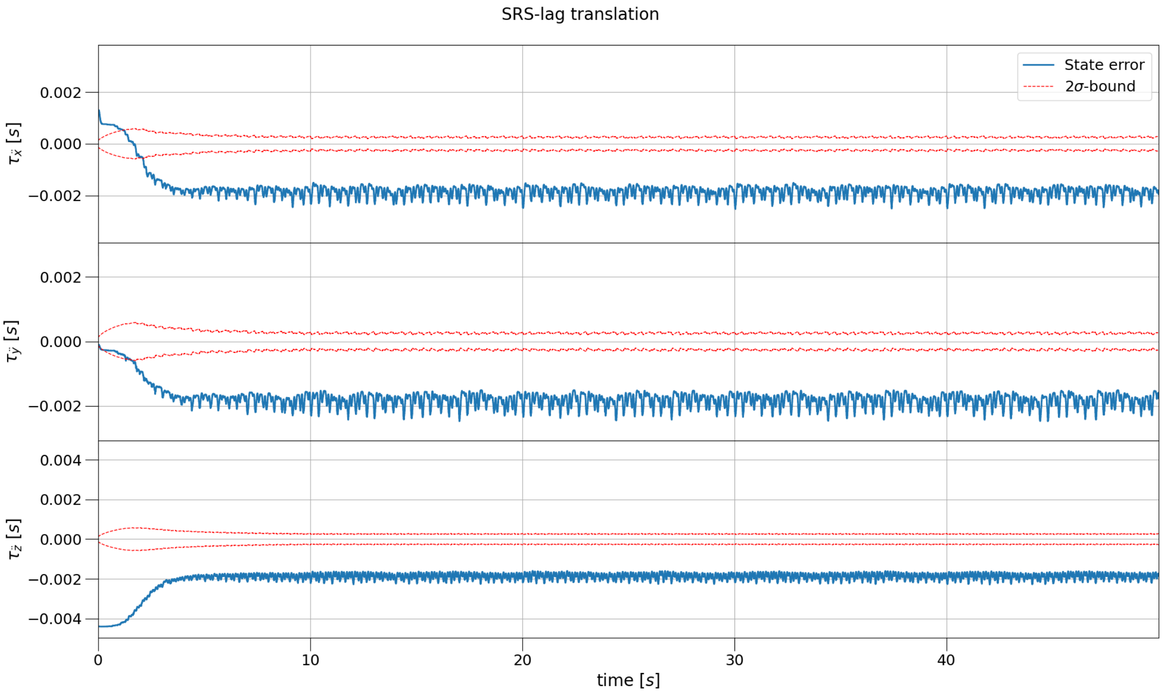


Figure 7.57: Sensitivity Run 3.10, the error on the translational lags the SRS per axis plotted against its covariance over time.

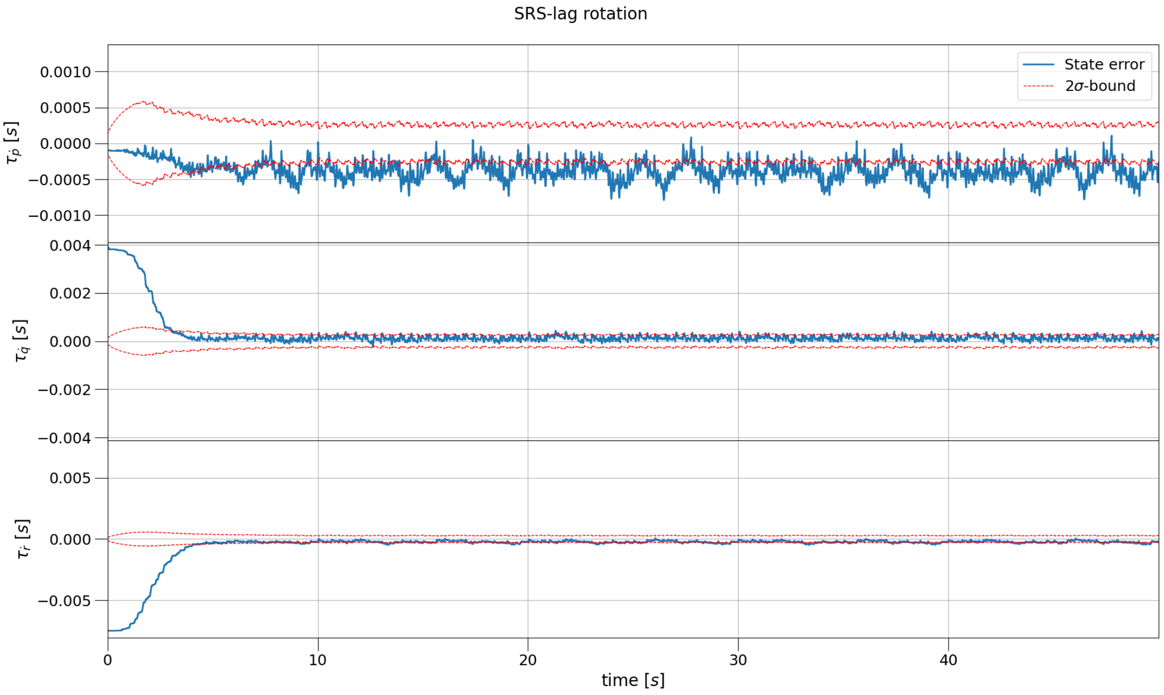


Figure 7.58: Sensitivity Run 3.10, the error on the rotational lags the SRS per axis plotted against its covariance over time.

Moreover, the expected SD on IMU was adapted to the motion at hand, resulting in scaling the expected SD on the accelerometer by 3.2. The accuracy of Run 3.10 is reduced because of the increase in jerk, angular rate, and angular acceleration increasing the error by integrating these values as a constant per prediction step. On first glance at Table 7.23 the accuracy impact seems worse than it actually is. The actual comparison is to Run 0.11 in the same table. Run 0.11 is the performance at $600\,$Hz of the TrackIR 5 by itself, i.e. Configuration 0, on the motion of Run 3.10. Comparing these two runs, Run 3.10 and 0.10, shows a higher accuracy improvement from Run 3 to Run 0 due to the increase in motion already mentioned.

The NEES values in Table 7.24 show that the motion is present in the resulting states and innovations. However, due to the method of integration this is expected.

Table 7.23: The error of each Configuration 3 modified run in terms of the vehicle-fixed position and attitude of the HMD.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}
3	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.76×10^{-3}	3.29×10^{-3}	1.95×10^{-2}
3.1.1	1.73×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.74×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.1.2	1.72×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.76×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.2.1	1.72×10^{-4}	8.34×10^{-5}	5.25×10^{-4}	6.74×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.2.2	1.74×10^{-4}	8.53×10^{-5}	5.41×10^{-4}	6.80×10^{-3}	3.33×10^{-3}	1.99×10^{-2}
3.3.1	-	-	-	-	-	-
3.3.2	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.75×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.3.3	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.75×10^{-3}	3.29×10^{-3}	1.95×10^{-2}
3.3.4	1.72×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.75×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.3.5	1.96×10^{-4}	1.05×10^{-4}	6.25×10^{-4}	7.92×10^{-3}	3.35×10^{-3}	2.12×10^{-2}
3.3.6	3.40×10^{-4}	1.72×10^{-4}	1.01×10^{-3}	8.40×10^{-3}	4.02×10^{-3}	2.49×10^{-2}
3.4.1	-	-	-	-	-	-
3.4.2	1.83×10^{-4}	9.08×10^{-5}	5.93×10^{-4}	7.65×10^{-3}	3.91×10^{-3}	1.96×10^{-2}
3.5.1	-	-	-	-	-	-
3.5.2	-	-	-	-	-	-
3.5.3	1.72×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.76×10^{-3}	3.31×10^{-3}	1.95×10^{-2}
3.5.4	1.72×10^{-4}	8.35×10^{-5}	5.29×10^{-4}	6.76×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.5.5	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.76×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.5.6	1.72×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.72×10^{-3}	3.29×10^{-3}	1.94×10^{-2}
3.6	5.62×10^{-4}	3.47×10^{-4}	2.17×10^{-3}	5.07×10^{-2}	3.59×10^{-2}	1.52×10^{-1}
3.7.1	-	-	-	-	-	-
3.7.2	1.72×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.68×10^{-3}	3.34×10^{-3}	1.94×10^{-2}
3.8.1	1.72×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.73×10^{-3}	3.30×10^{-3}	1.94×10^{-2}
3.8.2	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.74×10^{-3}	3.30×10^{-3}	1.95×10^{-2}
3.9.1	1.73×10^{-4}	8.34×10^{-5}	5.28×10^{-4}	6.75×10^{-3}	3.31×10^{-3}	1.95×10^{-2}
3.9.2	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.73×10^{-3}	3.29×10^{-3}	1.94×10^{-2}
0.10	1.99×10^{-3}	1.41×10^{-3}	7.57×10^{-3}	2.09×10^{-1}	1.37×10^{-1}	6.84×10^{-1}
3.10	4.97×10^{-4}	2.07×10^{-4}	1.20×10^{-3}	6.79×10^{-3}	2.74×10^{-3}	1.54×10^{-2}

Table 7.24: Condition overview of the modified runs of Configuration 3, see Section 7.4.1 for the explanation on the colors.

	NEES					
	SRS			HMD		
Run	Position	Attitude	Lin. Enc. Inno.	Position	Attitude	TrackIR 5 Inno.
3	2.35	2.31	4.64	3.11	1.97	5.11
3.1.1	2.29	2.25	4.58	3.11	1.97	5.12
3.1.2	2.34	2.35	4.70	3.11	1.98	5.11
3.2.1	2.20	2.25	3.75	3.10	1.97	5.10
3.2.2	31.58	12.38	13.89	3.25	1.99	5.11
3.3.1	-	-	-	-	-	-
3.3.2	2.35	2.31	4.63	3.11	1.97	5.11
3.3.3	2.33	2.29	4.63	3.11	1.97	5.11
3.3.4	2.34	2.31	4.61	3.11	1.96	5.11
3.3.5	18.63	10.82	4.53	3.58	2.61	5.03
3.3.6	1007.62	962.30	1460.70	3.00	2.14	7.36
3.4.1	-	-	-	-	-	-
3.4.2	2.45	2.29	4.61	3.61	2.26	5.11
3.5.1	-	-	-	-	-	-
3.5.2	-	-	-	-	-	-
3.5.3	2.34	2.28	4.62	3.11	1.98	5.11
3.5.4	2.33	2.29	4.62	3.11	1.98	5.11
3.5.5	2.34	2.29	4.63	3.11	1.98	5.11
3.5.6	2.34	2.31	4.63	3.11	1.96	5.11
3.6	3.53	2.84	5.70	5.45	3.34	5.59
3.7	2.47	2.28	4.70	3.11	1.97	5.11
3.7.1	-	-	-	-	-	-
3.7.2	2.30	2.25	4.62	3.11	1.95	5.11
3.8.1	2.34	2.29	4.63	3.11	1.96	5.11
3.8.2	2.33	2.29	4.63	3.11	1.96	5.11
3.9.1	2.33	2.29	4.63	3.11	1.97	5.11
3.9.2	2.33	2.28	4.61	3.11	1.96	5.11
3.10	40.58	21.71	45.91	19.10	1.99	9.53

Table 7.25: The error of each Configuration 3 modified run in terms of the inertial position and attitude of the SRS.

	SRS					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
3	7.61×10^{-6}	3.46×10^{-6}	2.12×10^{-5}	1.80×10^{-4}	7.92×10^{-5}	5.16×10^{-4}
3.1.1	7.86×10^{-6}	3.64×10^{-6}	2.44×10^{-5}	1.87×10^{-4}	8.36×10^{-5}	5.25×10^{-4}
3.1.2	7.54×10^{-6}	3.41×10^{-6}	2.12×10^{-5}	1.80×10^{-4}	7.82×10^{-5}	5.09×10^{-4}
3.2.1	1.08×10^{-5}	5.30×10^{-6}	4.11×10^{-5}	2.66×10^{-4}	1.26×10^{-4}	9.42×10^{-4}
3.2.2	1.70×10^{-5}	8.23×10^{-6}	4.35×10^{-5}	2.86×10^{-4}	1.54×10^{-4}	7.85×10^{-4}
3.3.1	-	-	-	-	-	-
3.3.2	7.61×10^{-6}	3.44×10^{-6}	2.10×10^{-5}	1.80×10^{-4}	7.88×10^{-5}	5.37×10^{-4}
3.3.3	7.56×10^{-6}	3.41×10^{-6}	2.11×10^{-5}	1.79×10^{-4}	7.84×10^{-5}	5.13×10^{-4}
3.3.4	7.59×10^{-6}	3.44×10^{-6}	2.10×10^{-5}	1.80×10^{-4}	7.91×10^{-5}	5.28×10^{-4}
3.3.5	1.19×10^{-5}	1.27×10^{-5}	9.51×10^{-5}	2.76×10^{-4}	3.38×10^{-4}	3.31×10^{-3}
3.3.6	2.11×10^{-4}	1.28×10^{-4}	6.21×10^{-4}	4.86×10^{-3}	2.80×10^{-3}	1.45×10^{-2}
3.4.1	-	-	-	-	-	-
3.4.2	7.69×10^{-6}	3.66×10^{-6}	2.44×10^{-5}	1.79×10^{-4}	7.77×10^{-5}	4.94×10^{-4}
3.5.1	-	-	-	-	-	-
3.5.2	-	-	-	-	-	-
3.5.3	7.58×10^{-6}	3.43×10^{-6}	2.12×10^{-5}	1.79×10^{-4}	7.84×10^{-5}	5.20×10^{-4}
3.5.4	7.55×10^{-6}	3.43×10^{-6}	2.10×10^{-5}	1.79×10^{-4}	7.86×10^{-5}	5.10×10^{-4}
3.5.5	7.58×10^{-6}	3.42×10^{-6}	2.14×10^{-5}	1.79×10^{-4}	7.83×10^{-5}	5.17×10^{-4}
3.5.6	7.58×10^{-6}	3.47×10^{-6}	2.11×10^{-5}	1.79×10^{-4}	7.96×10^{-5}	5.13×10^{-4}
3.6	8.37×10^{-4}	6.55×10^{-4}	3.05×10^{-3}	2.23×10^{-2}	1.68×10^{-2}	8.16×10^{-2}
3.7	7.84×10^{-6}	3.58×10^{-6}	2.21×10^{-5}	1.82×10^{-4}	8.02×10^{-5}	5.42×10^{-4}
3.7.1	-	-	-	-	-	-
3.7.2	7.47×10^{-6}	3.39×10^{-6}	2.06×10^{-5}	1.77×10^{-4}	7.77×10^{-5}	5.06×10^{-4}
3.8.1	7.58×10^{-6}	3.43×10^{-6}	2.11×10^{-5}	1.79×10^{-4}	7.90×10^{-5}	5.11×10^{-4}
3.8.2	7.55×10^{-6}	3.45×10^{-6}	2.14×10^{-5}	1.79×10^{-4}	7.89×10^{-5}	4.98×10^{-4}
3.9.1	7.57×10^{-6}	3.43×10^{-6}	2.11×10^{-5}	1.79×10^{-4}	7.83×10^{-5}	5.20×10^{-4}
3.9.2	7.55×10^{-6}	3.44×10^{-6}	2.09×10^{-5}	1.79×10^{-4}	7.85×10^{-5}	5.10×10^{-4}
3.10	4.45×10^{-5}	2.25×10^{-5}	1.21×10^{-4}	7.76×10^{-4}	4.05×10^{-4}	2.35×10^{-3}

In general, the Stewart platform of the SRS couples the measurement of the position and attitude. This causes the forward Euler induced error, not present on the attitude, to spread to the attitude. This is most evident from the NEES values in Tables 7.18, 7.21, and 7.24. Here, a regression in positional condition is coupled to a regression in attitude. This is not true for the NEES values of the HMD. Due to their observation models behave independent of one another.

7.5.4. Latency Compensation

In this section the impact of a latency of 50 ms on the linear encoder and its proposed mitigation is investigated. Here, the definition of latency is the time between the physical measurement and the arrival of the corresponding data. The mitigation used is the SDUKF proposed in Section 3.3.2.

First the impact of the latency is investigated. The combination of a latency of 50 ms and a sample rate of 100 Hz causes the linear encoders to buffer five measurements at a time in this case.

In all three configurations the impact of the latency in the non-mitigated case, indicated as 'Run x.D', is such that the accuracy of Configuration 0 is superior when looking at Table 7.28. Also, the impact on the accuracy of the SRS, compiled in Table 7.27 is similar for all three configuration.

Interestingly, the NEES values, in Table 7.26, of the linear encoders innovation in Run 1.D do not change from Run 1. In Configuration 1, the SRS subsystem prediction was designed to inject uncertainty into the acceleration and angular acceleration, the uncertainty to be integrated into the position and attitude, and for the linear encoders of the Stewart platform to estimate the state. Lacking any other input and the uncertainty, i.e. the observation noise $\boldsymbol{R}$, being small, the innovations of the linear encoders seem nominal compared to a non-delayed run.

This behavior is reaffirmed by comparing Figures 7.16 and 7.59.

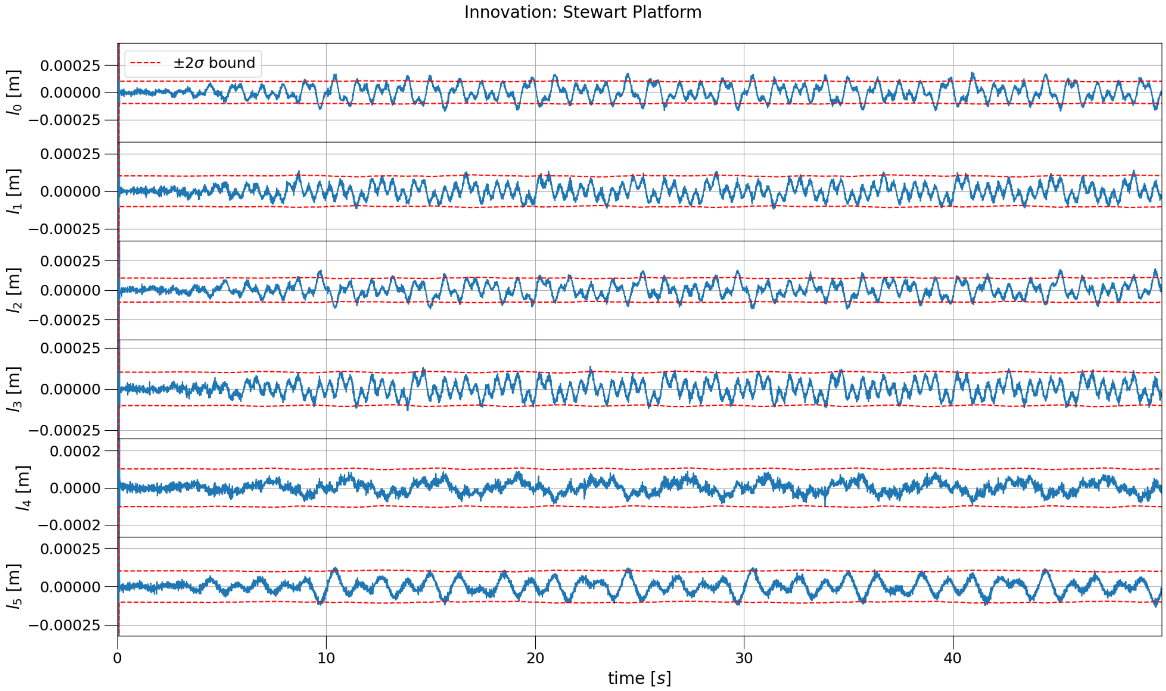


Figure 7.59: Delayed Run Configuration 1, the innovation of linear encoders measuring the length of the hydraulic actuators of the SRS plotted against its covariance over time.

However, this is not true for the HMD subsystem. Here, the TrackIR 5 innovation NEES, in Table 7.26, shows the disagreement with the IMU of the HMD and true for all subsequent configuration. The TrackIR 5 measurements, sampled in the upper moving platform reference frame $\mathbb{E}_P$, introduce the corruption due to the latency on the linear encoders. Comparing Figures 7.17 and 7.60 shows the detrimental effect of the linear encoders latency, the difference between the prediction, by the HMD IMU, and the correction, by the TrackIR 5.

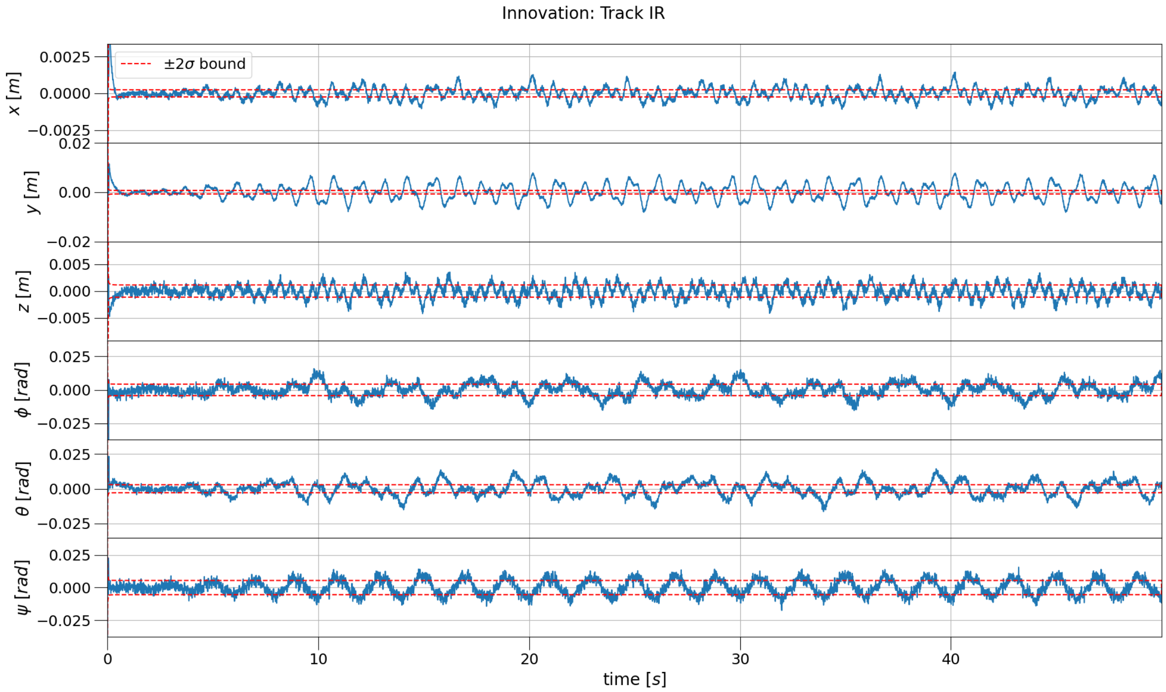


Figure 7.60: Delayed Run Configuration 1, the innovation of Track IR 5 measuring the position and attitude (Euler attitude formulation) of the HMD attached to the SRS plotted against its covariance over time.

Table 7.26: Condition overview of each run through the computation of the NEES, see Section 7.4.1 for the explanation on the color use. x.D indicator for the Delayed Runs. x.M indicator for the Mitigated Runs, i.e. were the delay is mitigated by use of the SDUKF, found in Section 3.3.2.

	NEES					
	SRS			HMD		
Run	Position	Attitude	Lin. Enc. Inno.	Position	Attitude	TrackIR 5 Inno.
1.SRS	2.56	2.60	7.41			
1.SRS.D	1.95×10^6	2.58×10^6	7.51			
1.SRS.M	12.30	13.30	7.48			
1	2.59	2.62	7.43	3.11	2.01	5.00
1.D	1.96×10^6	2.59×10^6	7.26	4.68×10^4	371.48	130.66
1.M	11.78	13.54	5.11	4.70	2.01	8.00
2	26.40	15.92	26.59	3.90	3.79	5.18
2.D	4.71×10^6	3.81×10^6	1187.86	4.67×10^4	375.52	115.99
2.M	85.63	21.68	25.61	6.02	3.70	5.44
3	2.35	2.31	4.64	3.11	1.97	5.11
3.D	4.53×10^6	5.81×10^6	263.66	4.68×10^4	373.50	137.89
3.M	1.86	1.31	3.49	3.17	2.00	5.10

Moreover, the algorithm solves the difference partly between the measurements of the TrackIR 5 and the HMD IMU as the bias of the IMU, evidenced in Figures 7.61 and 7.62. Comparing the following two figures, Figures 7.61 and 7.62, to Figures 7.11 and 7.12 highlights the motion in the bias error.

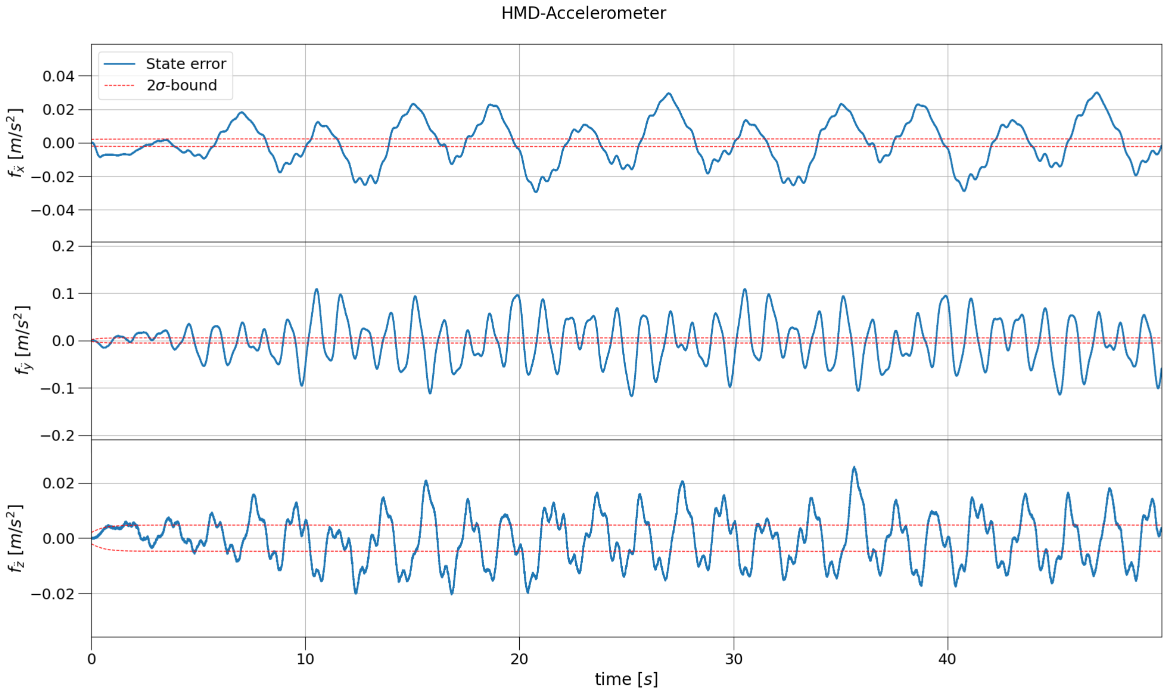


Figure 7.61: Delayed Run Configuration 1, the error of the accelerometer bias of the HMD per axis plotted against its covariance over time.

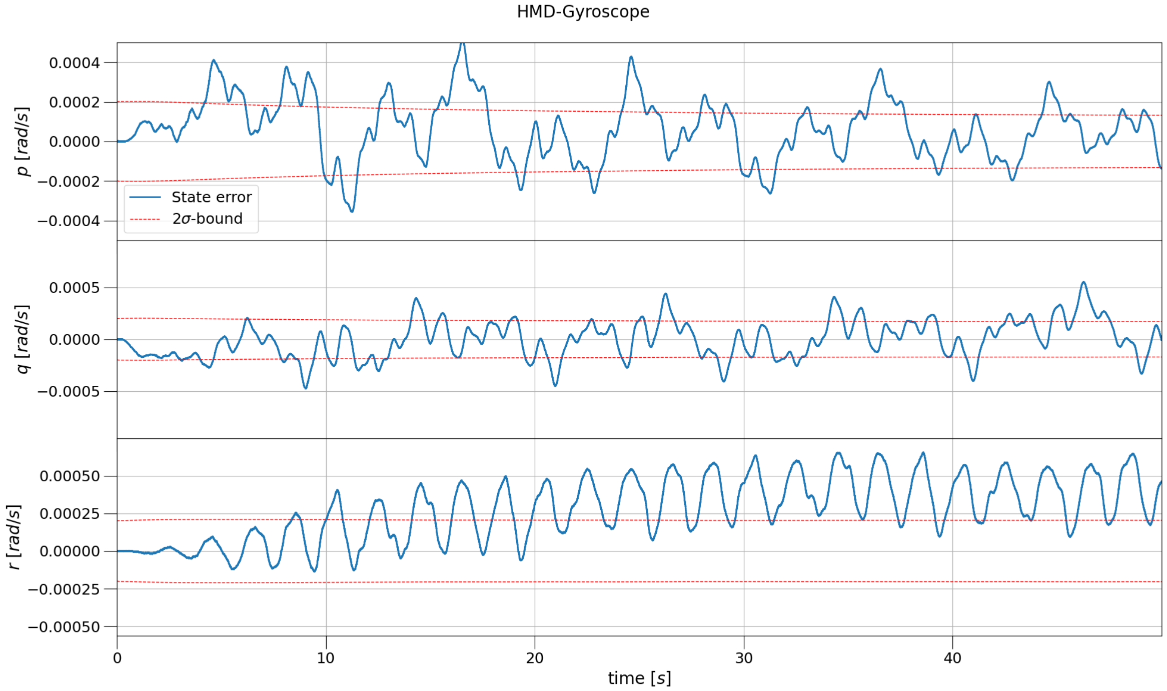


Figure 7.62: Delayed Run Configuration 1, the error of the gyroscope bias of the HMD per axis plotted against its covariance over time.

Further of note in the delayed runs are the lags on the setpoints of Configuration 3 and plotted in Figures 7.63 and 7.64. Both the translational and rotational lags are estimated 50 ms above their correct value. This is equal to the latency introduced in the linear encoders.

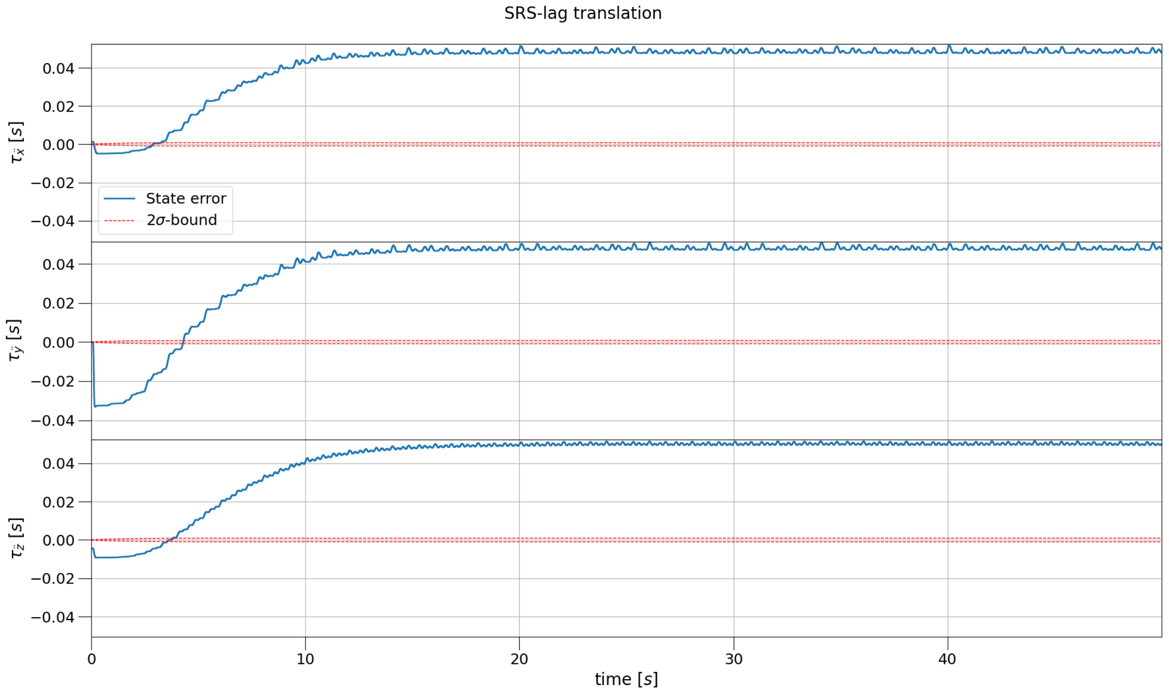


Figure 7.63: Delayed Run Configuration 3, the error on the translational lags the SRS per axis plotted against its covariance over time.

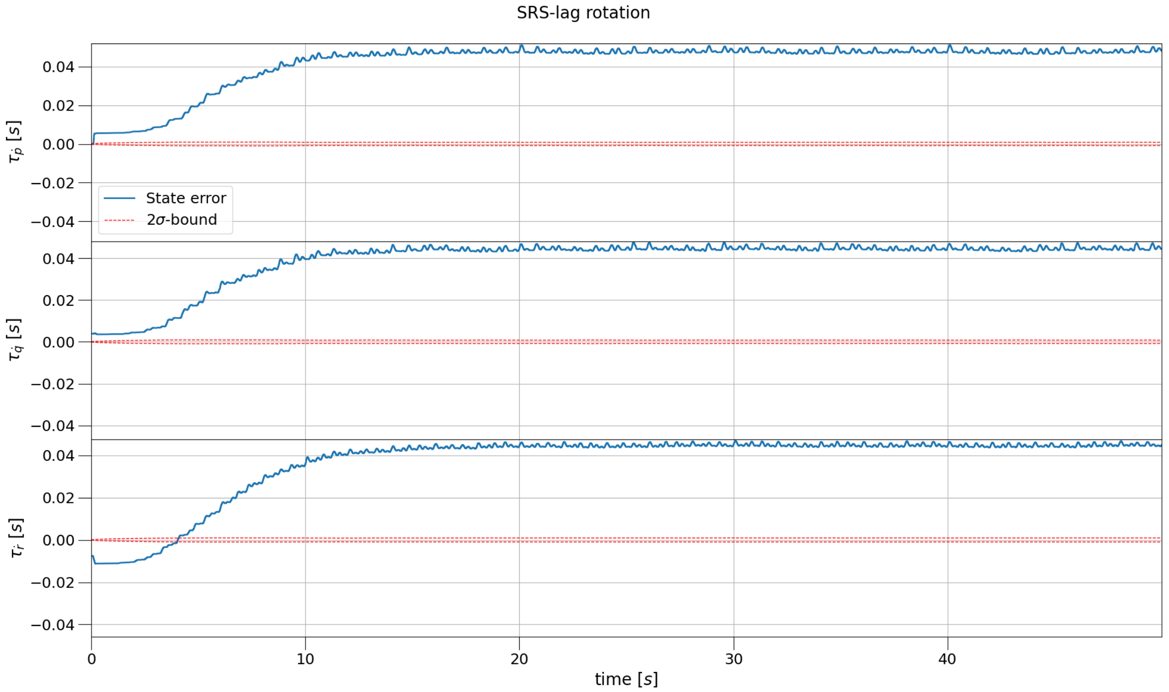


Figure 7.64: Delayed Run Configuration 3, the error on the rotational lags the SRS per axis plotted against its covariance over time.

In order to try and mitigate the negative impact of the latency present on the linear encoders the SDUKF is applied, as introduced in Section 3.3.2. The runs where SDUKF is applied are labeled x.M, x being the configuration.

Analyzing the SRS accuracy of the x.M runs, compiled in Table 7.27, Run 1.M recovers its accuracy the least. This is due to the fact that Configuration 1 has no direct input on the SRS subsystem. In this case, Configuration 1 is predicting the pose of the SRS based on the SRS state of 50 ms ago and the HMD IMU measurements indirectly through the TrackIR 5 sensor observation. In order to prove this statement, three extra runs were made with only the SRS as per Configuration 1. This means that only the SRS state $\hat{\boldsymbol{x}}_P$ was estimated, with a process F_P and observation model F_P as in Equations 5.8 and 4.44 respectively. These three runs are identified by 'Run 1.SRS' in Tables 7.26 and 7.27. Comparing Runs 1.SRS.M to 1.M, it is evident that the HMD IMU is beneficial in this case, reducing the error on the position and attitude by $35\,\%$ and $10\,\%$ respectively.

Mitigated Configurations 2 and 3 fare better due to their direct input into the SRS subsystem. Following, this input is integrated over the next 10 ms while receiving corrections from 50 ms through the past state.

Table 7.27: The error of each run in terms of the inertial position and attitude of the SRS. x.D indicator for the Delayed Runs. x.M indicator for the Mitigated Runs, i.e. were the delay is mitigated by use of the SDUKF, found in Section 3.3.2.

	SRS					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
1.SRS	2.95×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.73×10^{-4}	5.52×10^{-4}	3.75×10^{-3}
1.SRS.D	1.00×10^{-2}	3.20×10^{-3}	1.83×10^{-2}	2.66×10^{-1}	7.20×10^{-2}	4.88×10^{-1}
1.SRS.M	1.12×10^{-3}	4.75×10^{-4}	2.79×10^{-3}	3.32×10^{-2}	1.37×10^{-2}	7.67×10^{-2}
1	2.95×10^{-5}	2.09×10^{-5}	1.38×10^{-4}	7.71×10^{-4}	5.52×10^{-4}	3.58×10^{-3}
1.D	1.00×10^{-2}	3.20×10^{-3}	1.83×10^{-2}	2.66×10^{-1}	7.20×10^{-2}	4.88×10^{-1}
1.M	7.26×10^{-4}	3.12×10^{-4}	1.83×10^{-3}	3.00×10^{-2}	1.21×10^{-2}	6.67×10^{-2}
2	2.98×10^{-5}	1.71×10^{-5}	9.23×10^{-5}	9.30×10^{-4}	5.54×10^{-4}	4.54×10^{-3}
2.D	1.01×10^{-2}	3.25×10^{-3}	1.86×10^{-2}	2.67×10^{-1}	7.25×10^{-2}	4.86×10^{-1}
2.M	1.30×10^{-4}	6.87×10^{-5}	3.62×10^{-4}	5.31×10^{-3}	2.22×10^{-3}	1.53×10^{-2}
3	7.61×10^{-6}	3.46×10^{-6}	2.12×10^{-5}	1.80×10^{-4}	7.92×10^{-5}	5.16×10^{-4}
3.D	1.00×10^{-2}	3.21×10^{-3}	1.84×10^{-2}	2.67×10^{-1}	7.23×10^{-2}	4.89×10^{-1}
3.M	1.88×10^{-5}	8.23×10^{-6}	4.72×10^{-5}	3.76×10^{-4}	1.66×10^{-4}	9.90×10^{-4}

Analyzing the performance metrics on the errors of the HMD vehicle-fixed pose in Table 7.28 show similar behavior to the performance metrics on the SRS pose, compiled in Table 7.27. The difference between no-latency and latency mitigated runs are not as large as the case for the SRS. The TrackIR 5 would measure the correct pose after all, in the case of the latency it is the IMU that reduces the accuracy.

Table 7.28: The error of each run in terms of the vehicle-fixed position and attitude of the HMD. x.D indicator for the Delayed Runs. x.M indicator for the Mitigated Runs, i.e. were the delay is mitigated by use of the SDUKF, found in Section 3.3.2.

	HMD Vehicle-Fixed					
	Position Error m			Attitude Error °		
Run	Mean	SD	Max	Mean	SD	Max
0	7.36×10^{-4}	3.65×10^{-4}	2.37×10^{-3}	1.04×10^{-1}	4.74×10^{-2}	3.04×10^{-1}
1	1.70×10^{-4}	8.22×10^{-5}	5.05×10^{-4}	6.87×10^{-3}	3.35×10^{-3}	2.01×10^{-2}
1.D	2.90×10^{-3}	1.60×10^{-3}	7.76×10^{-3}	2.50×10^{-1}	6.42×10^{-2}	4.20×10^{-1}
1.M	5.65×10^{-4}	2.77×10^{-4}	1.77×10^{-3}	3.04×10^{-2}	1.22×10^{-2}	7.28×10^{-2}
2	1.93×10^{-4}	9.56×10^{-5}	5.80×10^{-4}	9.43×10^{-3}	4.78×10^{-3}	2.47×10^{-2}
2.D	2.71×10^{-3}	1.48×10^{-3}	6.97×10^{-3}	2.49×10^{-1}	6.46×10^{-2}	4.19×10^{-1}
2.M	2.04×10^{-4}	9.40×10^{-5}	5.75×10^{-4}	1.06×10^{-2}	4.52×10^{-3}	2.83×10^{-2}
3	1.72×10^{-4}	8.35×10^{-5}	5.28×10^{-4}	6.76×10^{-3}	3.29×10^{-3}	1.95×10^{-2}
3.D	2.94×10^{-3}	1.64×10^{-3}	7.87×10^{-3}	2.50×10^{-1}	6.45×10^{-2}	4.21×10^{-1}
3.M	1.76×10^{-4}	8.39×10^{-5}	5.19×10^{-4}	6.82×10^{-3}	3.35×10^{-3}	2.05×10^{-2}

Overall, the latency mitigation shows satisfactory performance, managing in Configuration 2 and 3 to overcome the latency imposed on the system almost completely. Only in Configuration 1, the improvements were less noticeable.

7.6. Conclusion

Keeping the experiment in mind, the innovation based NEES values are the most important measurable data in real life on which to base the success of the implementation [6] as the actual state is not available.

However, from the data presented in Tables 7.15, 7.18, 7.21, 7.24, and 7.26 it is concluded that the theoretical NEES values of one per DoF, as mentioned in Section 7.4.1, are not realistic in the case of the SRS in Configuration 2, where the combination of the relative low update rate, the motion profile, and discrete integration methods hurt the optimal theoretical performance of the observational NEES values, causing them to increase to values between 25 and 30 as indicated by Table 7.21.

Moreover, due the discrete method with which the angular rate and angular acceleration data were approximated for the simulation, see Section 7.2.1, the innovation based NEES values are not per se showing a realistic picture. Realistic values in Tables 7.15, 7.18, 7.21, 7.24 and 7.26 would be slightly higher in value, because the angular rate and acceleration used in this Thesis are approximations of reality.

Furthermore, Configuration 1, thus only linear encoders for the SRS, is sensitive to the balance between the SRS's expected acceleration and angular acceleration on the one hand, and the motion profile of the SRS on the other hand. The SRS pose estimation in Configuration 2 and 3 are less sensitive to a change in motion profile due to the increase in accuracy of their prediction, which is in turn caused by their addition of an IMU and setpoints respectively. As such, Configurations 2 and 3 are more robust to changes in the motion profile than Configuration 1, and therefore would yield a more optimal outcome.

Also, the cases were the expected noise variance for a sensor would be of a different order then the actual noise variance, the innovation based NEES values should be an indicator. However, the relevant sensor should be found by trial and error.

From Runs 1.6, 2.7, and 3.10, which doubled the amplitude and frequencies of the motion profile, show for all three configurations a sensitivity to the scale of the motion at hand. For this reason all three configurations should be tested with a motion profile of the SRS close to the maximum acceleration and angular acceleration possible.

Analyzing Equations 7.19 and 7.20 in combination with the convergence speed in Run 3.10 also hint at sensitivity to a lack of acceleration and angular acceleration of Configuration 3.

8

Experiment Plan

In order to test the success of the implementation proposed throughout this Thesis of a Virtual Reality (VR)/Augmented Reality (AR) system in an enclosed Stewart platform, an experiment is to be designed and performed. As mentioned in the conclusion of the sensitivity analysis, the success of the implementation will be evaluated by measuring the innovation based Normalized Estimated Error Squared (NEES) values. The layout and setup of this experiment will be presented in this chapter.

8.1. System Layout

The experiment will be conducted with the systems in the layout depicted in Figure 8.1. The sensors depicted in Figure 8.1 are the the sensors first introduced in Chapter 6.

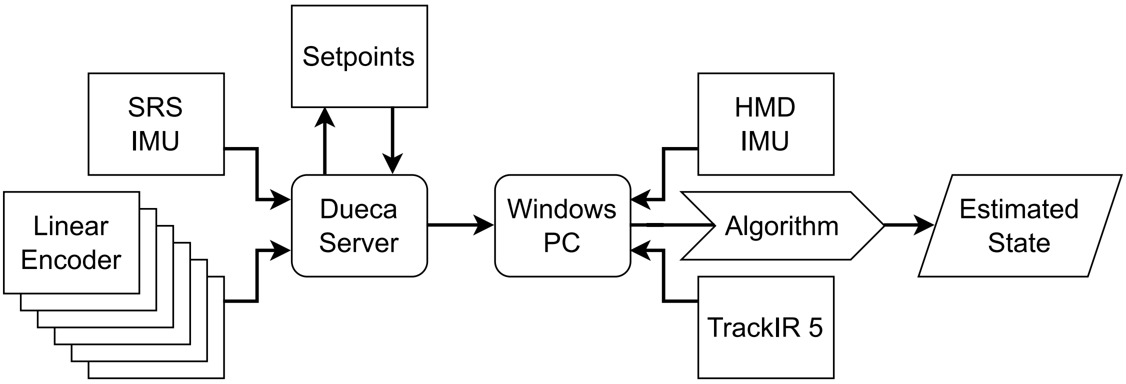


Figure 8.1: Experiment setup layout.

Starting on the left of Figure 8.1 with the six HeidenHain LC 415 [19] linear encoders of the SIMONA Research Simulator (SRS) transmitting their data to a server running DUECA.

DUECA, the abbreviation for Delft University Environment for Communication and Activation, is a middleware that enables real-time simulations by using both inter-module communication as well as communication between different modules on a distributed network of computers [50]. Moreover, DUECA uses a publish and subscribe design that enables modules, a self-contained possible real-time computational element, to activate when receiving or sending data. In design and use, it is similar to other middlewares like enhanced Communication Abstraction Layer (eCAL) [11], Message Queuing Telemetry Transport (MQTT) [48] and Robot Operating System (ROS) [49].

Also, the SRS's Inertial Measurement Unit (IMU) publishes its data onto DUECA. The setpoints of the SRS are generated by the motion limiter and filter algorithm, mentioned in Section 2.3, already running on a DUECA module and thus available for transmitting.

When any of the above mentioned data is available, it is sent to a Personal Computer (PC) running a recent Microsoft Windows operating system. The Microsoft Windows operating system is used since it is required by the proprietary software of TrackIR 5, which is connected to said PC over Universal Standard Bus (USB).

Furthermore, the measurements of the MPU-6500, the IMU onboard the Pimax 8k-x [52], are read by a modified version of the open-source tool libsurvive [35], which in turn also runs on the Microsoft Windows PC.

In addition, also running on the Microsoft Windows PC is the Unscented Kalman Filter (UKF) algorithm, described throughout this Thesis. The UKF predicts and corrects the estimate state depending on the data received, as well as the configuration selected.

8.2. Experiment Run

The experiment itself will consist out of two runs per motion profile per configuration. Both runs on a moving platform will last $300\,\mathrm{s}$. The first run will have the Head-Mounted Display (HMD) strapped in place inside the cockpit, i.e. no motion relative to the upper motion platform reference frame $\mathbb{E}_P$, and the second run will introduce movement relative to the upper motion platform reference frame $\mathbb{E}_P$, resembling a realistic use case. In the latter run, a user will track a virtual sphere trough the HMD to validate the working of the configuration in a realistic use case.

Two motion profiles for the SRS will be generated to stress the configurations in the manners discussed in Section 7.6. The first motion profile exist to stress all three configurations by creating a motion profile similar in design to the motion profile presented in Section 7.2, but takes into account the physical limitations of the SRS in terms of acceleration and angular acceleration.

The second motion profile will stress Configuration 3, whose lag's convergence speed is dependant on the absolute values of acceleration and angular acceleration setpoints, as shown in Section 7.6. Following, the second motion profile will consist of long durations with the SRS at a standstill interspersed with relative small movements.

8.3. Data Collection

In order to validate the implementation and verify its success the data incoming to the Microsoft Windows PC will be recorded. This data will consist of:

- DUECA data packages send to the Microsoft Windows PC
- IMU measurement data of the HMD read by the libsurvive [35] program
- The HMD pose interpreted by the TrackIR 5 proprietary software

As mentioned in Section 2.3, the state of the Stewart platform is already estimated via a Gauss-Newton (GN) method solely based on the linear encoders in order to compute the setpoints for said Stewart platform. This estimated state of the Stewart platform is recorded for validation purposes to be compared against the UKF estimated state.

During the runtime of the algorithm the following data is recorded at every prediction/correction phase:

- Estimated state $\underline{\hat{\boldsymbol{x}}}$
- Estimated covariance $\boldsymbol{P}_{\hat{x}}$
- Observational covariance $\boldsymbol{P}_{\hat{y}_k}$
- Innovation $\underline{\boldsymbol{\epsilon}}$

Using the innovation and its covariance $\boldsymbol{P}_{\hat{y}_k}$ the NEES ν values can be computed afterwards via Equation 7.64 to ascertain the success of the implementation.

Also, recording all this data has the added benefit that one recording can serve for testing/debugging of other Configurations offline.

8.4. Hypothesis

The aim set out by this Thesis was to present a solution on the implementation of a HMD with both an IMU and visual tracking inside an enclosed motion platform by developing a sensor fusion algorithm that generates a proper relative pose estimate. In order to quantify every implementation the innovation based NEES ν should be used [6].

The theoretical NEES values for both the set of linear encoders and the TrackIR 5 should, according to theory [6], equal to 6. Corresponding to the sensitivity analysis in Section 7.5 values between 4 and 9 are deemed acceptable for all innovation based NEES values, except for the set of linear encoders in Configuration 2. Due to the combination of the motion profile, discrete integration methods used, and the relative low update rate of the SRS subsystem colors the innovation values. Due to the aforementioned reasons it is not realistic to expect the theoretical optimal results of six, but rather according to Table 7.22 and also taking into account the fact that the angular simulation is a discrete approximation of reality the expected NEES values should be between 25 and 30.

9 Conclusion

To sum up, this report presented the preliminary findings of the research aiming to designing an effective fusion algorithm that generates a proper relative pose estimation of a Head-Mounted Display (HMD) of a Virtual Reality (VR)/Augmented Reality (AR) system within an enclosed motion platform.

For the sensor fusion algorithm the Unscented Kalman Filter (UKF) was chosen and adapted further into the Smoothed Delayed Unscented Kalman Filter (SDUKF) to deal with a possible delayed observation sensor.

Three configurations were identified from which its estimated state the proper, vehicle-fixed, pose of the HMD of a VR/AR system could be computed.

These three configuration differ very little to one another but for the sensor layout of the SIMONA Research Simulator (SRS), a Stewart platform. The first configuration, Configuration 1, only uses the linear encoders measuring the length of the actuators to estimate the state of the SRS. Next, Configuration 2 also used the linear encoders in addition to an Inertial Measurement Unit (IMU) attached to the upper moving motion platform. Lastly, Configuration 3 uses the future information encapsulated within the setpoints to be fused with the linear encoders. These setpoints are generated by the motion limiter and filter algorithm meant as motion commands for the motion control computer, which directs the pose of the Stewart platform itself.

The sensor layout, and thus the mathematical model, of the HMD is the same in all three configurations. The measurements stemming from the HMD's built-in IMU are fused together with the outside-in visual tracking.

Next to these three configurations a baseline configuration was presented where only the visual outside-in tracking, already measuring the vehicle-fixed pose, would be used. By forgoing the fast and frequent pose predictions enabled by the IMU of the HMD the need to estimate the Stewart platform state was removed. This baseline configuration will serve as a benchmark to the three just mentioned configurations.

The HMD's IMU was identified and parametrized by a Allan Variation (AV) method. Moreover, it was found difficult to generate a noise model for the measurements coming from the proprietary software of the visual tracking sensor. Further validation of said noise model must be done.

A sensitivity analysis showed that Configuration 1, the configuration with only the linear encoders to estimate the pose of the SRS, showed sensitivity to changes made that upset the balance between the motion profile and the SRS's expected process noise values.

The sensitivity analysis also showed that the innovation based Normalized Estimated Error Squared (NEES) values of the linear encoders in case of Configuration 2, where the IMU attached to the upper moving motion platform was enabled, were higher then the theoretical expected value. This was due to a combination of three factors namely the relative low selected update rate of 100 Hz, the motion profile, and the integration methods.

The sensitivity analysis of Configuration 3, where the setpoints were used to help predict the SRS's future state, showed promising results where the algorithm designed was capable of estimating the approximate first order lag online. It must be noted, however, that the setpoints for the simulation were computed exactly according to the first order system response, thus reducing the uncertainty in the simulated system. The experiment, using the innovation based NEES values, will give an indication about the optimality of Configuration 3.

The method used for mitigating the delay on observation sensors in all three configurations, in this case the linear encoders, was almost capable of removing the impact of said delay given exact knowledge of the delay.

In order to validate the three configurations as solutions in combination with the proposed UKF, and thus find a conclusive answer to the aforementioned research question, the following Hypothesis was formulated:

> The innovation based NEES values should be between 4 and 9 for both innovation sets, i.e. linear encoders and TrackIR 5 innovations, except for the set of linear encoders in Configuration 2 where values are expected between 25 and 30 on average during the runtime of the sensor fusion algorithm.

The Hypothesis will be tested according to the experiment plan setup in the previous chapter, Chapter 8.

References

[1] Sunjoo Kan Advani. “The Kinematic design of flight simulator motion bases”. PhD thesis. Delft, Netherlands: Delft University, Apr. 1998. ISBN: 90-407-1672-2.

[2] Harold L. Alexander. “State estimation for distributed systems with sensing delay”. In: *Data Structures and Target Classification*. Ed. by Vibeke Libby. Vol. 1470. International Society for Optics and Photonics. SPIE, 1991, pp. 103–111. DOI: 10.1117/12.44843. URL: https://doi.org/10.1117/12.44843.

[3] C. Anthes et al. “State of the art of virtual reality technology”. In: *2016 IEEE Aerospace Conference*. Mar. 2016, pp. 1–19. DOI: 10.1109/AERO.2016.7500674.

[4] Advanced Realtime Tracking GmbH & Co. KG (ART). *ART*. Accessed: 01-02-2022. URL: https://ar-tracking.com/.

[5] M. Bak et al. “Location estimation using delayed measurements”. In: *AMC’98 - Coimbra. 1998 5th International Workshop on Advanced Motion Control. Proceedings (Cat. No.98TH8354)*. June 1998, pp. 180–185. DOI: 10.1109/AMC.1998.743533.

[6] Yaakov bar-shalom, X. Rong Li, and Thia Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Jan. 2004. ISBN: 047141655X. DOI: 10.1002/0471221279.ch11.

[7] Miguel Borges et al. “HTC Vive: Analysis and Accuracy Improvement”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2018, pp. 2610–2615. DOI: 10.1109/IROS.2018.8593707.

[8] Stephen Butterworth. “On the Theory of Filter Amplifiers”. In: *Experimental Wireless & The Wireless Engineer* (Oct. 1930), pp. 536–541.

[9] Antônio C. B. Chiella, Bruno O. S. Teixeira, and Guilherme A. S. Pereira. “Quaternion-Based Robust Attitude Estimation Using an Adaptive Unscented Kalman Filter”. In: *Sensors* 19.10 (2019). ISSN: 1424-8220. DOI: 10.3390/s19102372. URL: https://www.mdpi.com/1424-8220/19/10/2372.

[10] André-Louis Cholesky. “Note Sur Une Méthode de Résolution des équations Normales Provenant de L’Application de la MéThode des Moindres Carrés a un Système D’équations Linéaires en Nombre Inférieur a Celui des Inconnues. — Application de la Méthode a la Résolution D’un Système Defini D’éQuations LinéAires”. In: *Bulletin géodésique* 2.1 (Apr. 1924), pp. 67–77. ISSN: 1432-1394. DOI: 10.1007/BF03031308. URL: https://doi.org/10.1007/BF03031308.

[11] Continental AG. *eCAL Documentation*. 2021. URL: https://continental.github.io/ecal/index.html (visited on 02/15/2022).

[12] Fabrizio Cutolo et al. “Ambiguity-Free Optical–Inertial Tracking for Augmented Reality Headsets”. In: *Sensors* 20.5 (Mar. 2020), p. 1444. ISSN: 1424-8220. DOI: 10.3390/s20051444. URL: http://dx.doi.org/10.3390/s20051444.

[13] James E. Dieudonne, Russell V. Parrish, and Richard E. Bardusch. “An Actuator Extension Transformation for a Motion Simulator and an Inverse Transformation Applying Newton-Raphson’s Method”. In: *Technical Reports: NASA-TN-D-7067* (Nov. 1972).

[14] A. O. Ercan and A. T. Erdem. “On sensor fusion for head tracking in augmented reality applications”. In: *Proceedings of the 2011 American Control Conference*. June 2011, pp. 1286–1291. DOI: 10.1109/ACC.2011.5991077.

[15] Jay A Farrell et al. “IMU Error Modeling Tutorial: INS state estimation with real-time sensor calibration”. In: Oct. 2021.

[16] E.A. Gerber and A. Ballato. *Precision Frequency Control: Oscillators and standards*. Precision Frequency Control. Academic Press, 1985. ISBN: 9780122806025. URL: https://books.google.nl/books?id=PQ2EzgEACAAJ.

[17] Gourlay et al. “Head-Mounted-Display Tracking for Augmented and Virtual Reality”. In: *Information Display* 33.1 (2017), pp. 6–10. ISSN: 0362-0972. DOI: 10.1002/j.2637-496X.2017.tb00962.x. URL: https://doi.org/10.1002/j.2637-496X.2017.tb00962.x.

[18] S. Haykin and S.S. Haykin. *Adaptive Filter Theory*. Pearson, 2014. ISBN: 9780132671453. URL: https://books.google.nl/books?id=J4GRKQEACAAJ.

[19] *Heidenhain LC 400*. Accessed: 18-02-2022. URL: https://www.heidenhain.com/products/linear-encoders/sealed/lc-400.

[20] Dai Hong-de et al. “Performance Comparison of EKF/UKF/CKF for the Tracking of Ballistic Target”. In: *TELKOMNIKA Indonesian Journal of Electrical Engineering* 10 (Nov. 2012). DOI: 10.11591/telkomnika.v10i7.1564.

[21] “IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros”. In: *IEEE Std 952-1997* (Oct. 1998), pp. 1–84. DOI: 10.1109/IEEESTD.1998.86153.

[22] Simon J. Julier and Jeffrey K. Uhlmann. “New extension of the Kalman filter to nonlinear systems”. In: *Signal Processing, Sensor Fusion, and Target Recognition VI*. Ed. by Ivan Kadar. Vol. 3068. International Society for Optics and Photonics. SPIE, 1997, pp. 182–193. DOI: 10.1117/12.280797. URL: https://doi.org/10.1117/12.280797.

[23] Rudolph Emil Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[24] H. C. Kam, Y. K. Yu, and K. H. Wong. “An Improvement on ArUco Marker for Pose Tracking Using Kalman Filter”. In: *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. June 2018, pp. 65–69. DOI: 10.1109/SNPD.2018.8441049.

[25] Mara Kaufeld and Thomas Alexander. “The Impact of Motion on Individual Simulator Sickness in a Moving Base VR Simulator with Head-Mounted Display”. In: *Virtual, Augmented and Mixed Reality. Multimodal Interaction*. Ed. by Jessie Y.C. Chen and Gino Fragomeni. Cham: Springer International Publishing, 2019, pp. 461–472. ISBN: 978-3-030-21607-8.

[26] Andrew Kiruluta, Moshe Eizenman, and Subbarayan Pasupathy. “Predictive head movement tracking using a Kalman filter”. In: *IEEE Trans. Syst. Man Cybern. Part B* 27.2 (1997), pp. 326–331. DOI: 10.1109/3477.558841. URL: https://doi.org/10.1109/3477.558841.

[27] Nirmal Kj et al. “Noise modeling and analysis of an IMU-based attitude sensor: improvement of performance by filtering and sensor fusion”. In: July 2016, 99126W. DOI: 10.1117/12.2234255.

[28] G. Klein and D. Murray. “Parallel Tracking and Mapping for Small AR Workspaces”. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Nov. 2007, pp. 225–234. DOI: 10.1109/ISMAR.2007.4538852.

[29] Sjirk Koekebakker. “Model Based Control of a Flight Simulator Motion System”. PhD thesis. Delft, Netherlands: Delft University, Dec. 2001. ISBN: 90-370-0194-7.

[30] Stanislaw Konatowski, Piotr Kaniewski, and Jan Matuszewski. “Comparison of Estimation Accuracy of EKF, UKF and PF Filters”. In: *Annual of Navigation* 23 (Dec. 2016). DOI: 10.1515/aon-2016-0005.

[31] Roger R. Labbe. *Kalman and Bayesian Filters in Python*. Accessed: 25-10-2023. 2021. URL: https://drive.google.com/file/d/0By_SW19c1BfhSVFzNHc0SjduNzg/view?usp=sharing.

[32] Bob M. Lansdorp and Omar A. Saleh. “Power spectrum and Allan variance methods for calibrating single-molecule video-tracking instruments”. In: *Review of Scientific Instruments* 83.2 (2012), p. 025115. DOI: 10.1063/1.3687431. eprint: https://doi.org/10.1063/1.3687431. URL: https://doi.org/10.1063/1.3687431.

[33] T.D. Larsen et al. “Incorporation of time delayed measurements in a discrete-time Kalman filter”. In: *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*. Vol. 4. 1998, 3972–3977 vol.4. DOI: 10.1109/CDC.1998.761918.

[34] Steven M. LaValle et al. “Head Tracking for the Oculus Rift”. In: (2014). URL: http://lavalle.pl/papers/LavYerKatAnt14.pdf.

[35] *libsurvive*. Accessed: 15-02-2022. URL: https://github.com/cntools/libsurvive.

[36] Shyam M et al. “Introduction to the Kalman Filter and Tuning its Statistics for Near Optimal Estimates and Cramer Rao Bound”. In: (Mar. 2015).

[37] Landis Markley et al. “Averaging Quaternions”. In: *Journal of Guidance, Control, and Dynamics* 30 (July 2007), pp. 1193–1196. DOI: 10.2514/1.28949.

[38] matzman666. *OpenVR InputEmulator*. 2020. URL: https://github.com/matzman666/OpenVR-InputEmulator.

[39] Michael McCauley and Thomas Sharkey. “Cybersickness: Perception of Self-Motion in Virtual Environment.” In: *Presence* 1 (Jan. 1992), pp. 311–318. DOI: 10.1162/pres.1992.1.3.311.

[40] Mark McGill et al. “A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’15. Seoul, Republic of Korea: Association for Computing Machinery, 2015, pp. 2143–2152. ISBN: 9781450331456. DOI: 10.1145/2702123.2702382. URL: https://doi.org/10.1145/2702123.2702382.

[41] Rudolph Merwe and Eric Wan. “Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models”. In: *Proceedings of the Workshop on Advances in Machine Learning* (June 2003).

[42] Microsoft. *Overview of Packet Coalescing*. Accessed: 30-10-2023. 2023. URL: https://learn.microsoft.com/en-us/windows-hardware/drivers/network/overview-of-packet-coalescing.

[43] I. Miletović et al. “Improved Stewart platform state estimation using inertial and actuator position measurements”. In: *Control Engineering Practice* 62 (2017), pp. 102–115. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2017.03.006. URL: http://www.sciencedirect.com/science/article/pii/S0967066117300643.

[44] *MPU-6500 Product Specification Revision 1.3*. Accessed: 31-01-2022. URL: https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6500/%5C#product-documentation.

[45] Parag Narkhede et al. “Cascaded Complementary Filter Architecture for Sensor Fusion in Attitude Estimation”. In: *Sensors* 21.6 (2021). ISSN: 1424-8220. DOI: 10.3390/s21061937. URL: https://www.mdpi.com/1424-8220/21/6/1937.

[46] Diederick C. Niehorster, Li Li, and Markus Lappe. “The Accuracy and Precision of Position and Orientation Tracking in the HTC Vive Virtual Reality System for Scientific Research”. In: *i-Perception* 8.3 (2017), p. 2041669517708205. DOI: 10.1177/2041669517708205. URL: https://doi.org/10.1177/2041669517708205.

[47] Janosch Nikolic. “Characterisation, Calibration, and Design of Visual-Inertial Sensor Systems for Robot Navigation”. en. PhD thesis. Zürich: ETH Zurich, 2016. DOI: 10.3929/ethz-a-010874630.

[48] OASIS Message Queuing Telemetry Transport (MQTT) TC. *MQTT Version 5.0*. 2021. URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html (visited on 02/15/2022).

[49] Open Robotics. *Robot Operating System (ROS)*. 2021. URL: https://www.ros.org (visited on 02/15/2022).

[50] Marinus M. van Paassen, Olaf Stroosma, and Jerome Delatour. “DUECA - Data-driven activation in distributed real-time computation”. In: Aug. 2000. DOI: 10.2514/6.2000-4503.

[51] W. F. Phillips, C. E. Hailey, and G. A. Gebert. “Review of Attitude Representations Used for Aircraft Kinematics”. In: *Journal of Aircraft* 38.4 (2001), pp. 718–737. DOI: 10.2514/2.2824. eprint: https://doi.org/10.2514/2.2824. URL: https://doi.org/10.2514/2.2824.

[52] Pimax Technologies, LLC. *Pimax 8k-x*. Accessed: 30-01-2022. URL: https://pimax.com/product/vision-8k-x/.

[53] *P2 Pimax 8K Teardown Internal Photos Int Photos Pimax Technology*. Accessed: 30-01-2022. URL: https://fccid.io/2ARKN-P2/Internal-Photos/Int-Photos-4077345.

[54] D.M. Pool et al. "Optimal Reconstruction of Flight Simulator Motion Cues using Extended Kalman Filtering". In: *Proceedings of the AiAA Modeling and Simulation Technologies Conference and Exhibit*. Ed. by F. Weller and M. Duquette. United States: American Institute of Aeronautics and Astronautics Inc. (AIAA), 2008, pp. 1–20. ISBN: DVD978-1-56347-945-8.

[55] S. Rangelova and E. Andre. "A survey on simulation sickness in driving applications with virtual reality head-mounted displays". English. In: *Presence: Teleoperators and Virtual Environments* 27.1 (2019), pp. 15–31.

[56] A. Rhijn, Robert van Liere, and J.D. Mulder. "An analysis of orientation prediction and filtering methods for VR/AR". In: Apr. 2005, pp. 67–74. ISBN: 0-7803-8929-8. DOI: 10.1109/VR.2005.1492755.

[57] William Riley and David Howe. *Handbook of Frequency Stability Analysis*. en. July 2008. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50505.

[58] S. Ropelato et al. "Adaptive tutoring on a virtual reality driving simulator". English. In: *International series on information systems and management in creative eMedia*. Vol. 2017. 2017. Chap. 2, pp. 12–17.

[59] Guido van Rossum. *Python Tutorial*. Python Software Foundation, Sept. 2018.

[60] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. "Analysis and Modeling of Inertial Sensors Using Allan Variance". In: *Instrumentation and Measurement, IEEE Transactions on* 57 (Feb. 2008), pp. 140–149. DOI: 10.1109/TIM.2007.908635.

[61] Malcolm David Shuster. "Survey of attitude representations". In: *Journal of The Astronautical Sciences* (1993).

[62] *SIMONA Research Simulator*. Accessed: 06-07-2022. URL: https://cs.lr.tudelft.nl/simona/facility/visual-display-system/.

[63] D. Stewart. "A Platform with Six Degrees of Freedom". In: *Proceedings of the Institution of Mechanical Engineers* 180.1 (1965), pp. 371–386. DOI: 10.1243/PIME_PROC_1965_180_029_02. eprint: https://doi.org/10.1243/PIME_PROC_1965_180_029_02. URL: https://doi.org/10.1243/PIME%5C_PROC%5C_1965%5C_180%5C_029%5C_02.

[64] Olaf Stroosma, Marinus M. van Paassen, and Max Mulder. "Using the SIMONA Research Simulator for Human-machine Interaction Research". In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. 2012. DOI: 10.2514/6.2003-5525. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2003-5525. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2003-5525.

[65] Ivan E. Sutherland. "A Head-Mounted Three Dimensional Display". In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. AFIPS '68 (Fall, part I). San Francisco, California: Association for Computing Machinery, 1968, pp. 757–764. ISBN: 9781450378994. DOI: 10.1145/1476589.1476686. URL: https://doi.org/10.1145/1476589.1476686.

[66] Ivan E. Sutherland. "The Ultimate Display". In: *Proceedings of the IFIP Congress*. 1965, pp. 506–508. URL: https://my.eng.utah.edu/~cs6360/Readings/UltimateDisplay.pdf.

[67] Gregory K. Tharp et al. "Timing considerations of helmet-mounted display performance". In: *Human Vision, Visual Processing, and Digital Display III*. Ed. by Bernice E. Rogowitz. Vol. 1666. International Society for Optics and Photonics. SPIE, 1992, pp. 570–576. DOI: 10.1117/12.136003. URL: https://doi.org/10.1117/12.136003.

[68] E. Thöndel. "Virtual reality for a realistic simulation with dynamic motion platforms". English. In: *33rd Annual European Simulation and Modelling Conference 2019, ESM 2019*. 2019, pp. 259–263.

[69] NaturalPoint INC. *TrackIR 5*. Accessed: 30-01-2022. URL: https://www.naturalpoint.com/trackir/trackir5/.

[70] Roberto G. Valenti, Ivan Dryanovski, and Jizhong Xiao. “Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs”. In: *Sensors* 15.8 (2015), pp. 19302–19330. ISSN: 1424-8220. DOI: 10.3390/s150819302. URL: https://www.mdpi.com/1424-8220/15/8/19302.

[71] John Vince. *Quaternions for Computer Graphics*. 1st. Springer Publishing Company, Incorporated, 2011. ISBN: 0857297597.

[72] James R. Wertz. “Three-Axis Attitude Determination Methods”. In: *Spacecraft Attitude Determination and Control*. Dordrecht: Springer Netherlands, 1978, pp. 410–435. ISBN: 978-94-009-9907-7. DOI: 10.1007/978-94-009-9907-7_12. URL: https://doi.org/10.1007/978-94-009-9907-7_12.

[73] Oliver J. Woodman. *An introduction to inertial navigation*. Tech. rep. UCAM-CL-TR-696. University of Cambridge, Computer Laboratory, Aug. 2007. URL: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf.

[74] Jiann-Rong Wu and Ming Ouhyoung. “On latency compensation and its effects on head-motion trajectories in virtual environments”. In: *Vis. Comput.* 16.2 (2000), pp. 79–90. DOI: 10.1007/s003710050198. URL: https://doi.org/10.1007/s003710050198.

A

Overlapping Allan Deviation

In this section, the Overlapping Allan Deviation (OAD) is first briefly discussed, followed by an explanation via an example of the possible result of an Overlapping Allan Deviation (OAD) sensor analysis.

The Allan Deviation (AD) is a time-domain-analysis technique capable of determining both the derivation as well as the source of the Gaussian noise. Given that the noise profile is also assumed Gaussian in the AD, this noise profiling method is well suited to be used in the Kalman Filter (KF)-framework [73, 15, 27].

The OAD is a superior adaptation of the AD by exploring the available data in different size 'overlapping' bins [32]. Given M number of measured samples z, measured at a constant time step τ_0, arranged in varying sized bins of size m, named averaging factor, the variation σ^2 can be computed as:

$$\sigma^2(m\tau_0) = \frac{1}{2m^2\left(M-2m+1\right)} \sum_{j=1}^{M-2m+1} \left(\sum_{i=j}^{j+m-1} z_{i+m} - z_i \right)^2 \tag{A.1}$$

The deviation is simply the square-root of the variation, i.e.,

$$\sqrt{\sigma^2(m\tau_0)} = \sigma(m\tau_0) \tag{A.2}$$

Stated in [21, 16], the relationship between the Allan variance σ^2 and the two-sided Power Spectral Density (PSD) Ω is given in terms of frequency f as:

$$\sigma^2(m\tau_0) = 4 \int_0^\infty \Omega(f) \frac{\sin^4\left(\pi f m \tau_0\right)}{\left(\pi f m \tau_0\right)^2} df \tag{A.3}$$

When the deviation, defined in Equations A.1 and A.2, of an Inertial Measurement Unit (IMU) is plotted on a log-log scale the result will be equivalent to the example in Figure A.1. In this figure, the quantization noise was added for completeness, however, due its diminished contribution to the overall random process, they are not further discussed, see [21, 60] for more information on these topics. The angle/velocity random walk, bias instability and rate/acceleration random walk are further explained in Section A.1, A.3 and A.2 respectively.

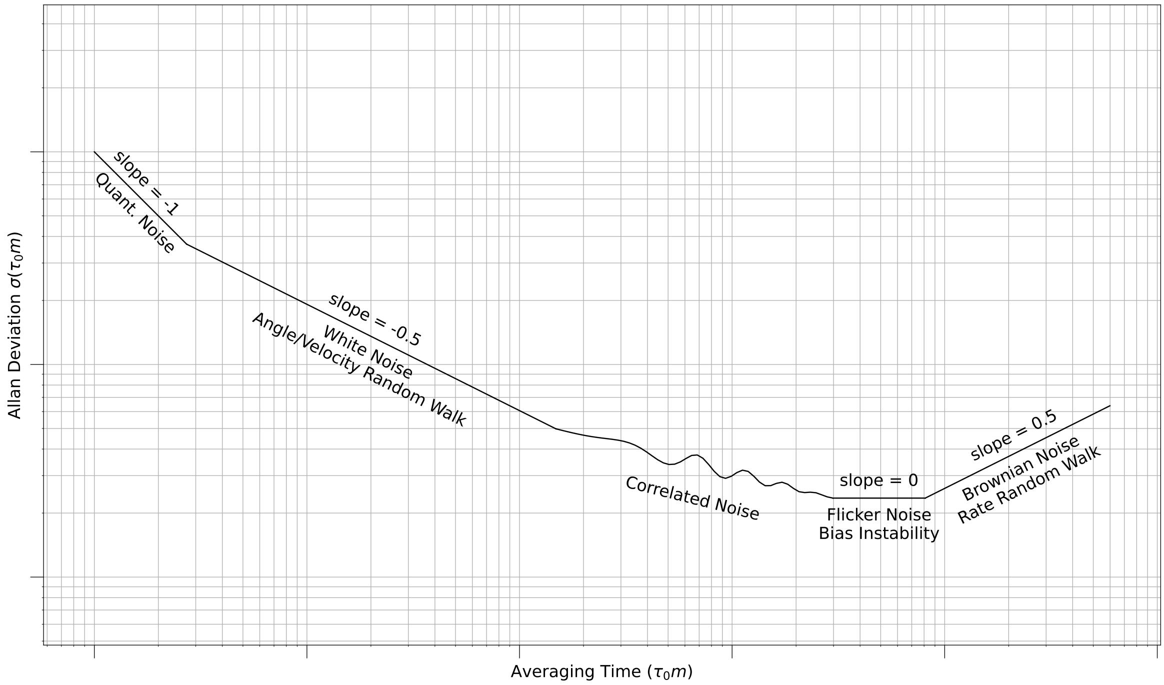


Figure A.1: An example of a log-log plot of an Allan Deviation sensor analysis [57]

Figure A.1 is plotted on a log-log scale as it helps to graphically indicate various noise sources and their derivation by fitting a certain sloped line through said plot. The result of substituting the known rate PSD Ω into Equation A.3 and solve for its integration will indicate the slope and averaging time $m\tau_0$ for a particular noise source.

A.1. Angle/Velocity Random Walk

The angle or velocity random walk is another term for the white random noise that is present on a IMU output and presents itself as Gaussian spread errors around the actual measurement. The white noise spectrum PSD Ω is defined by the angle/velocity random walk coefficient N as:

$$\Omega(f) = N^2 \tag{A.4}$$

Integrating Equation A.3 while substituting Equations A.4 results in:

$$\sigma^2(m\tau_0) = \frac{N^2}{m\tau_0} \tag{A.5}$$

Expressing Equation A.5 on a log-log scale proves that the slope of the fitting line is equal to $-1/2$, i.e.,

$$\sqrt{\sigma^2(m\tau_0)} = N\sqrt{\frac{1}{m\tau_0}}$$

$$\log\left(\sigma(m\tau_0)\right) = \frac{-1}{2}\log\left(m\tau_0\right) + \log\left(N\right) \tag{A.6}$$

After fitting a line with a slope of $-1/2$ graphically through the plotted Allan deviation σ versus the averaging time $m\tau_0$ on a log-log scale, the value for N can be found at the averaging time set to one, i.e. $m\tau_0 = 1$, a result of Equation A.5. The units of the angle or velocity random walk coefficient N are rad/s/$\sqrt{\text{Hz}}$ or m/s^2/$\sqrt{\text{Hz}}$ respectively depending on the sensor being a gyroscope or an accelerometer.

A.2. Rate/Acceleration Random Walk

The rate or acceleration random walk is also called Brownian noise and presents itself as the derivative of a gyroscope or accelerometer respectively. The PSD of the rate/acceleration random walk is defined by the rate/acceleration random walk factor K as:

$$\Omega(f) = \left(\frac{K^2}{2\pi}\right)^2 \frac{1}{f^2} \tag{A.7}$$

Substituting into Equation A.3 and solving the integration results in:

$$\sigma^2(m\tau_0) = K^2 \frac{m\tau_0}{3} \tag{A.8}$$

Expressing Equation A.8 on a log-log scale proves that the slope of the fitting line is equal to $1/2$, i.e.,

$$\begin{aligned}\sqrt{\sigma^2(m\tau_0)} &== K\sqrt{\frac{m\tau_0}{3}} \\ \log\left(\sigma(m\tau_0)\right) &= \frac{1}{2}\log\left(m\tau_0\right) + \log\left(K\right)\end{aligned} \tag{A.9}$$

After fitting a line with a slope of $1/2$ graphically through the plotted Allan deviation σ versus the averaging time $m\tau_0$ on a log-log scale, the value for K can be found at the averaging time set to one, i.e. $m\tau_0 = 3$, a result of Equation A.8. The units of the rate or acceleration random walk coefficient N are rad/s$\sqrt{\text{Hz}}$ or m/s$^2\sqrt{\text{Hz}}$ respectively depending on the sensor being a gyroscope or an accelerometer.

A.3. Bias Instability

Bias instability, or also called pink noise, presents itself trough random fluctuations in the bias itself. This noise factor is usually small, relative to the overall error modeling due to angle/velocity random walk and rate/acceleration random walk. The bias instability can be ignored for modeling purposes if the overlapping time at which the log-log plot is flat is relatively short. The PSD of the bias instability is defined by the bias instability term B as:

$$\Omega(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right)\frac{1}{f} & f \leq f_0 \\ 0 & f > f_0 \end{cases} \tag{A.10}$$

Substituting into Equation A.3 and solving the integration results in:

$$\sigma^2(m\tau_0) = \frac{2B^2}{\pi}\left(\ln 2 - \frac{\sin^3\Upsilon}{2\Upsilon^2}\left(\sin\Upsilon + 4\Upsilon\cos\Upsilon\right) + Ci(2\Upsilon) - Ci(4\Upsilon)\right) \tag{A.11}$$

Where, Υ represents and Ci the cosine-integral function. When the duration of the averaging time is much longer than the inverse of the cutoff frequency, i.e. $m\tau_0 \gg 1/f_0$, Equation A.11 can be simplified to:

$$\sigma^2(m\tau_0) = \frac{2B^2}{\pi}\ln 2 \tag{A.12}$$

This results in a line with slope of 0 when seen on the same log-log scale of averaging time vs Allan derivation, as presented in Figure A.1. This means that the value of the bias instability coefficient B can be graphically deduced where the Allan derivation σ bottoms out. The Allan derivation σ found at the lowest point in the graph should be scaled to find the bias instability coefficient B. i.e.,

$$\begin{aligned} B &= \sqrt{\frac{\pi}{2\ln 2}}\,\sigma(m\tau_0) \\ &\approx 1.505\,\sigma(m\tau_0)\end{aligned} \tag{A.13}$$

The units of the bias instability coefficient B are, depending on the sensor being a gyroscope or an accelerometer, rad/s or m/s^2 respectively.

III

Appendices to the Thesis Paper

B

Rotation Moment Linear Acceleration coupling

For Configuration 2, the one where both the Head-Mounted Display (HMD) and the SIMONA Research Simulator (SRS) have an Inertial Measurement Unit (IMU), the process model f_P of the SRS should be altered to improve the incorporation of its own IMU input $\underline{\boldsymbol{u}}_P$. The reason for this alteration is due to the fact that the fact that the IMU of the SRS is not located at the Upper Gimbal Point (UGP), i.e., the center of its reference frame $\mathbb{E}_P$. The position $\underline{\boldsymbol{c}}_{\text{IMU}}$ of the SRS's IMU to the UGP expressed in the upper moving reference frame $\mathbb{E}_P$ is written as:

$$\begin{aligned}\underline{\boldsymbol{c}}_{\text{IMU}} &= \begin{pmatrix} x_{\text{IMU}} & y_{\text{IMU}} & x_{\text{IMU}} \end{pmatrix} \{\mathbb{E}_P\} \\ &= \begin{pmatrix} 0.305 & 0 & 0.0105 \end{pmatrix} \{\mathbb{E}_P\} \end{aligned} \tag{B.1}$$

The IMU's distance to the center of rotation causes a rotation-moment-linear-acceleration coupling component $\underline{\boldsymbol{a}}_{\omega_P}$ [41, 43]:

$$\begin{aligned}\underline{\boldsymbol{a}}_{\omega_P} &= \begin{pmatrix} \ddot{x}_{\omega_P} & \ddot{y}_{\omega_P} & \ddot{z}_{\omega_P} \end{pmatrix} \{\mathbb{E}_P\} \\ &= \dot{\underline{\boldsymbol{\omega}}}_P \times \underline{\boldsymbol{c}}_{\text{IMU}} + \underline{\boldsymbol{\omega}}_P \times \left(\underline{\boldsymbol{\omega}}_P \times \underline{\boldsymbol{c}}_{\text{IMU}}\right) \end{aligned} \tag{B.2}$$

$$= \begin{Bmatrix} \dot{p}_P \\ \dot{q}_P \\ \dot{r}_P \end{Bmatrix} \times \begin{Bmatrix} x_{\text{IMU}} \\ y_{\text{IMU}} \\ x_{\text{IMU}} \end{Bmatrix} + \begin{Bmatrix} p_P \\ q_P \\ r_P \end{Bmatrix} \times \left(\begin{Bmatrix} p_P \\ q_P \\ r_P \end{Bmatrix} \times \begin{Bmatrix} x_{\text{IMU}} \\ y_{\text{IMU}} \\ x_{\text{IMU}} \end{Bmatrix} \right) \tag{B.3}$$

$$\begin{Bmatrix} \ddot{x}_{\omega_P} \\ \ddot{y}_{\omega_P} \\ \ddot{z}_{\omega_P} \end{Bmatrix} = \begin{Bmatrix} q_P(p_P y_{\text{IMU}} - q_P x_{\text{IMU}}) + \dot{q}_P z_{\text{IMU}} + r_P(p_P z_{\text{IMU}} - r_P x_{\text{IMU}}) - \dot{r}_P y_{\text{IMU}} \\ -p_P(p_P y_{\text{IMU}} - q_P x_{\text{IMU}}) - \dot{p}_P z_{\text{IMU}} + r_P(q_P z_{\text{IMU}} - r_P y_{\text{IMU}}) + \dot{r}_P x_{\text{IMU}} \\ -p_P(p_P z_{\text{IMU}} - r_P x_{\text{IMU}}) + \dot{p}_P y_{\text{IMU}} - q_P(q_P z_{\text{IMU}} - r_P y_{\text{IMU}}) - \dot{q}_P x_{\text{IMU}} \end{Bmatrix} \tag{B.4}$$

Here, × indicates the vector-cross-product, and $\dot{\underline{\boldsymbol{\omega}}}_P$ is the angular acceleration of $\mathbb{E}_P$ computed by differentiating the angular rate $\underline{\boldsymbol{\omega}}_P$ and taking the mean of all Sigma-points. In turn, the angular rate $\underline{\boldsymbol{\omega}}_P$ of $\mathbb{E}_P$ in this configuration is computed as:

$$\underline{\boldsymbol{\omega}}_P = \underline{\boldsymbol{\omega}}_{P_m} - \underline{\boldsymbol{\lambda}}_{\underline{\boldsymbol{\omega}}_P} - \underline{\boldsymbol{n}}_{\underline{\boldsymbol{\omega}}_P} \tag{B.5}$$

Finally, the rotation-moment-linear acceleration coupling component $\underline{\boldsymbol{a}}_{\omega_P}$ is included in the SRS's process model f_P in Equation 4.35 as:

$$
\begin{aligned}
&\underline{\dot{\boldsymbol{x}}}_P(t) = \mathrm{f}_P\left(\underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{u}}_P(t), \underline{\boldsymbol{\nu}}_P(t)\right) \\
&\begin{Bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \\ \ddot{x}_P \\ \ddot{y}_P \\ \ddot{z}_P \\ \dot{e}_{0_P} \\ \dot{e}_{x_P} \\ \dot{e}_{y_P} \\ \dot{e}_{z_P} \\ \dot{\lambda}_{\ddot{x}_P} \\ \dot{\lambda}_{\ddot{y}_P} \\ \dot{\lambda}_{\ddot{z}_P} \\ \dot{\lambda}_{p_P} \\ \dot{\lambda}_{q_P} \\ \dot{\lambda}_{r_P} \end{Bmatrix}
= \left\{ \begin{matrix}
\dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \\
\boldsymbol{T}_{PI}^T \begin{Bmatrix} f_{\ddot{x}_{P_m}} - \lambda_{\ddot{x}_P} - n_{\ddot{x}_P} + \ddot{x}_{\omega_P} \\ f_{\ddot{y}_{P_m}} - \lambda_{\ddot{y}_P} - n_{\ddot{y}_P} + \ddot{y}_{\omega_P} \\ f_{\ddot{z}_{P_m}} - \lambda_{\ddot{z}_P} - n_{\ddot{z}_P} + \ddot{z}_{\omega_P} \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ g \end{Bmatrix} \\
\frac{1}{2} \begin{bmatrix} -e_{x_P} & -e_{y_P} & -e_{z_P} \\ e_{0_P} & -e_{z_P} & e_{y_P} \\ e_{z_P} & e_{0_P} & -e_{x_P} \\ -e_{y_P} & e_{x_P} & e_{0_P} \end{bmatrix} \begin{Bmatrix} p_{P_m} - \lambda_{p_P} - n_{p_P} \\ q_{P_m} - \lambda_{q_P} - n_{q_P} \\ r_{P_m} - \lambda_{r_P} - n_{r_P} \end{Bmatrix} \\
\mu_{\ddot{x}_P} \\ \mu_{\ddot{y}_P} \\ \mu_{\ddot{z}_P} \\ \mu_{p_P} \\ \mu_{q_P} \\ \mu_{r_P}
\end{matrix} \right\}
\end{aligned}
\tag{B.6}
$$

C

Testing Procedure

In this chapter the testing procedure is presented. First, the system for recording the data is presented in Section C.1. Next, various motion profiles for the SIMONA Research Simulator (SRS) are discussed in Section C.2. The motion profiles presented range from standstill, in order to validate the various noise parameters, to very firm motion, in order to stress the system's non-linearities.

However, before the testing procedure can commence, a calibration procedure of the TrackIR 5 is necessary in order to create an pose offset. This is required as the visual-tracking system, TrackIR 5, presents a pose that is the offset from system initialization. The offset position is not an issue for the pose estimation of the Head-Mounted Display (HMD). However, the attitude offset is problematic due to its reliance to cancel out the influence of the gravity. The TrackIR 5 calibration process is presented in Appendix D.

C.1. Data Recording System

The data recording system is records the three systems shown in bold on top of Figure C.1. Below those, the software packages are given with a short fact list. The right most software package, enhanced Communication Abstraction Layer (eCAL) [11], is used for recording all the data.

It was opted to record the data using eCAL on a Windows Personal Computer (PC). For this reason, the data of the SRS is send by Delft University Environment for Communication and Activation (DUECA) over an ethernet connection using User Datagram Protocol (UDP) from a Linux to Windows environment, shown in Figure C.2. The Windows PC receives the UDP messages and publishes them in turn onto eCAL.

The TrackIR 5 sensor is directly connected to the Windows PC, which in turn runs the software that translates and publishes the TrackIR 5 measurements onto eCAL, also depicted in Figure C.2.

The HMD's Universal Standard Bus (USB) data was decoded using libsurvive, an open-source software

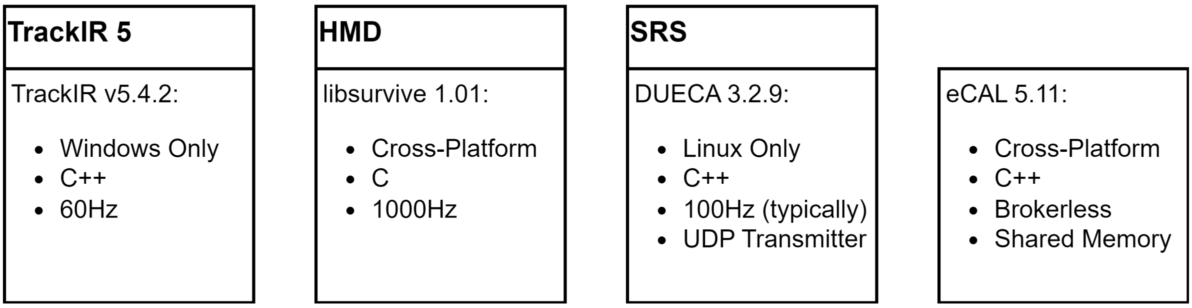


Figure C.1: An overview of the physical systems alongside their software packages.
The software packages from left to right: TrackIR v5 [69], libsurvive [35], DUECA [50], and eCAL [11].

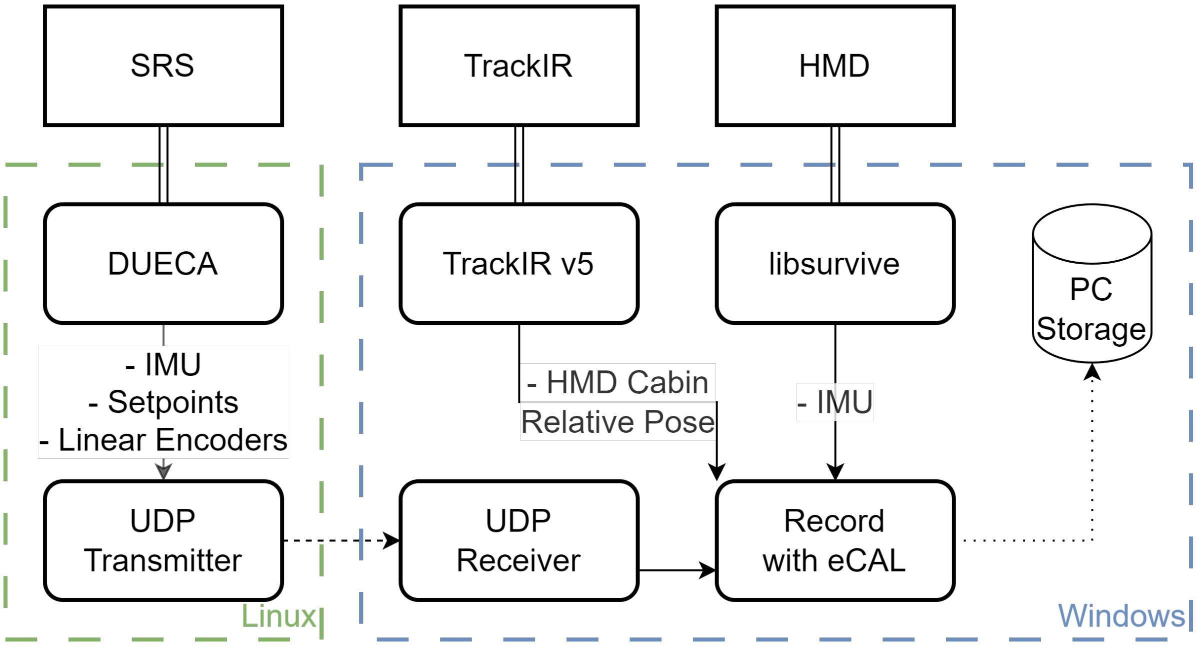


Figure C.2: Recording system data flow.

package, and published onto eCAL.

C.2. Motion Profiles

Each motion profile, except for standstill, consist of summed sinusoids per axis over its duration. The amplitudes and frequencies of each sinusoids will be given in tables. In order for the system to have ample time to perform the calibration shown in Section D, a stationary time of one minute was inserted before each motion profile. Furthermore, all motion profiles have a fade-in and fade-out duration of ten seconds. This was achieved by scaling thw whole motion profile with a hyperbolic tangent. Finally, for each motion profile, a short reasoning was given for its existence.

The motion profiles builds up from standstill to singular sinusoid on one axis to the original motion profile presented in Table 7.1 in order for potential problems to be isolated.

No Movement

- Duration: 10 minutes
- Description: Standstill
- Reasoning:
  - Configuration 3 could be problematic as it potentially divide by zero.
  - Validate that the real-time software implementation works.
  - Prior knowledge about both HMD and SRS pose.
  - Test the validity of the calibration duration.
  - Validate the noise parameters.

Simple Motion around a Single Axis

Let <rotation> → roll, pitch, yaw:

SRS <rotation> around DERP

- Duration: (1 + 5 minutes) × 3
- Description:

- SRS: Sinusoidal $<$rotation$>$ motion around Design Eye Reference Point (DERP), with an amplitude of $10°$ and frequency of 0.2 Hz.
- HMD: Stationary relative to cockpit.

- Reasoning:
  - HMD's Inertial Measurement Unit (IMU) should only measure $<$rotation$>$.
  - Prior knowledge about the HMD vehicle-fixed pose.
  - Comparisons can be made to other motion profiles with $<$rotation$>$ around the Upper Gimbal Point (UGP).

SRS $<$rotation$>$ around UGP

- Duration: (1 + 5 minutes) × 3
- Description:
  - SRS: Sinusoidal $<$rotation$>$ motion around UGP, ith an amplitude of $10°$ and frequency of 0.2 Hz.
  - HMD: Stationary relative to cockpit.
- Reasoning:
  - HMD's IMU should, in addition to the $<$rotation$>$, also measure the centripetal forces.
  - Prior knowledge about HMD relative pose.
  - Comparisons can be made to other motion profiles with $<$rotation$>$ around DERP.

SRS's Circular Translation around UGP in the xy-plane

- Duration: 1 + 5 minutes
- Description:
  - SRS: Circular motion in the xy plane, with an amplitude of 0.2 m and frequency of 0.2 Hz.
  - HMD: Stationary relative to cockpit.
- Reasoning:
  - Verify translational implementation of the algorithm while there's no rotation.
  - Verify that HMD stays static relative to cabin even though both systems are defined in the inertial frame.

Complex Motion around a Single Axis

The downside of moving at a singular amplitude and frequency combination is that this motion could hide or highlight aspects of the system, i.e. subsystem delay, the integration errors following the sinusoidal motion, convergence of the algorithm. In order to verify that the results found in the previous section are valid for a wide range of motion, the commanded motion, in this section, is the combination of multiple amplitudes and frequencies.

As the user moves their head, it is assumed that the resulting motion will be complex.

Let $<$rotation$>$ $\rightarrow$ roll, pitch, yaw:

HMD $<$rotation$>$

- Duration: (1 + 2 minutes) × 3
- Description:
  - SRS: Stationary.
  - HMD: $<$rotation$>$ motion with an amplitude of approximately $15°$ and a frequency of approximately 1 Hz.
- Reasoning:

- Prior knowledge about SRS's pose.
- Verify that the HMD's motion does not influence the SRS pose estimation.

SRS <rotation> around UGP

- Duration: (1 + 5 minutes) × 3
- Description:
  - HMD: Stationary relative to cockpit.
  - SRS: Sinusoidal <rotation> motion around UGP with:

Amplitude [°]	Frequency [Hz]
-4.01	0.1
4.01	0.25
0.705	0.65
0.917	0.85
0.115	2.0

- Reasoning: See section introduction.

Complex Motion

By increasing the total motion the total integration error will increase. This will stress the accuracy and thus possible the convergence of the system.

SRS Motion

- Duration: 1 + 5 minutes
- Description:
  - HMD: Stationary relative to cockpit.
  - SRS: Sinusoidal motion around UGP with:

Translation	Amplitude [m]	Frequency [Hz]
x:	-0.1	0.1
	0.1	0.25
	0.0123	0.65
	0.016	0.85
	0.002	2.0
y:	0.1	-0.1
	0.1	-0.25
	0.0123	0.65
	0.016	0.85
	0.002	2.0
z:	0.03	0.5
	0.002	2.0
Rotation	Amplitude [°]	Frequency [Hz]
roll:	-4.01	0.1
	4.01	0.25
	0.705	0.65
	0.917	0.85
	0.115	2.0
pitch:	4.01	-0.1
	4.01	-0.25
	0.705	0.65
	0.917	0.85
	0.115	2.0
yaw:	2.865	0.5
	0.115	2.0

- Reasoning:
  - By keeping the HMD stationary, its pose, and particularly its pose relative to the cockpit, can still be ascertained with some certainty.
  - The SRS based results can be compared to DUECA own SRS pose estimates for validation.

Full Motion

- Duration: 1 + 2 minutes
- Description:
  - HMD: Glance around cockpit, around x, y, z.
  - HMD: See previous motion
  - Reasoning:
    - The SRS results can still be compared to DUECA own SRS pose estimates for validation.
    - This run can be compared to the previous run, to possible observe how the extra superimposed HMD motion influences the SRS pose estimation.

D

TrackIR 5 Calibration Using the Head-Mounted Display's Inertial Measurement Unit

A calibration must be performed before the system can be used in order to create an attitude offset and a point of origin of the Head-Mounted Display (HMD) reference frame. This is necessary as the visual tracking system, TrackIR 5 presented in Section 6.1.2, does not present an **absolute** but a pose **relative** to initialization of the TrackIR 5 sensor.

D.1. Position

The relative position is not an issue for the pose estimation of the HMD as the offset position $\underline{\boldsymbol{c}}_{H_\delta}$ can be guessed at the expected head position, i.e., Design Eye Reference Point (DERP):

$$\left(\underline{\boldsymbol{c}}_{HP_\delta}\right)\{\mathbb{E}_P\} = \begin{pmatrix}0 & -0.550 & -1.2075\end{pmatrix}\{\mathbb{E}_P\} \tag{D.1}$$

This assumption has little negative consequences for the user and algorithm, as the error would be constant and can be easily adjusted by the user itself.

D.2. Attitude

However, the TrackIR's relative attitude is problematic for the pose estimation of the HMD. An initial attitude offset would influence the transformation of the Inertial Measurement Unit (IMU) measurements into the inertial reference frame $\mathbb{E}_I$ in Equations 4.30 and 4.33. Therefore, the potential attitude offset should preferably be found before starting the algorithm.

In order to find this attitude offset $\underline{\boldsymbol{e}}_{HP_\delta}$, the gravitational vector of the HMD's IMU is used to compute the tilt of the attitude offset $\breve{\underline{e}}_{HP}$ [70] and is compared to the relative attitude $\underline{\boldsymbol{e}}_{HP_m}$ provided by the TrackIR sensor.

At standstill the accelerometer measurements should only measure gravity $\underline{\boldsymbol{g}}$ in the HMD reference frame $\mathbb{E}_H$, resulting in,

$$f_{\ddot{x}_H}^2 + f_{\ddot{y}_H}^2 + f_{\ddot{z}_H}^2 = g^2 \tag{D.2}$$

Or written in their respective reference frames,

$$\begin{gathered}\left(\underline{\boldsymbol{f}}_H\right)\{\mathbb{E}_H\} = \left(\underline{\boldsymbol{g}}\right)\{\mathbb{E}_I\} \\ \begin{pmatrix}f_{\ddot{x}_H} & f_{\ddot{y}_H} & f_{\ddot{z}_H}\end{pmatrix}\{\mathbb{E}_H\} = \begin{pmatrix}0 & 0 & g\end{pmatrix}\{\mathbb{E}_I\}\end{gathered}$$

At initialization it is assumed that the upper platform is at rest, i.e. the upper platform $\mathbb{E}_P$ and inertial $\mathbb{E}_I$ reference frames align,

$$= \begin{pmatrix}0 & 0 & g\end{pmatrix}\{\mathbb{E}_P\}$$

Using the transformation matrix $\boldsymbol{T}_{HP}^T$, found in Equation 4.3, to write the whole equation into the HMD reference frame $\mathbb{E}_H$,

$$= \begin{pmatrix}0 & 0 & g\end{pmatrix}\boldsymbol{T}_{HP}^T\{\mathbb{E}_H\} \tag{D.3}$$

Transposing Equation D.3 and dividing by gravity g leads to,

$$\begin{Bmatrix} f_{\ddot{x}_H}/g \\ f_{\ddot{y}_H}/g \\ f_{\ddot{z}_H}/g \end{Bmatrix} = \boldsymbol{T}_{HP}\begin{Bmatrix}0\\0\\1\end{Bmatrix}$$

Furthermore, expanding the transformation matrix $\boldsymbol{T}_{HP}$ using Equation 4.5 leads to,

$$= \begin{Bmatrix} 2\left(e_{x_{HP}}e_{z_{HP}} - e_{0_{HP}}e_{y_{HP}}\right) \\ 2\left(e_{y_{HP}}e_{z_{HP}} + e_{0_{HP}}e_{x_{HP}}\right) \\ e_0^2 - e_{x_{HP}}^2 - e_{y_{HP}}^2 + e_{z_{HP}}^2 \end{Bmatrix} \tag{D.4}$$

The transfer between the upper platform $\mathbb{E}_P$ and HMD $\mathbb{E}_H$ reference frames can be achieved in infinite rotations. Restricting the yaw, i.e. $e_{z_{HP}} = 0$, results in,

$$= \begin{Bmatrix} -2e_{0_{HP}}e_{y_{HP}} \\ 2e_{0_{HP}}e_{x_{HP}} \\ e_0^2 - e_{x_{HP}}^2 - e_{y_{HP}}^2 \end{Bmatrix} \tag{D.5}$$

The system presented in Equation D.5 is fully determined, taking into account that Equation D.2 can be rewritten as $\left(\frac{f_{\ddot{x}_H}}{g}\right)^2 + \left(\frac{f_{\ddot{y}_H}}{g}\right)^2 + \left(\frac{f_{\ddot{z}_H}}{g}\right)^2 = 1$, and choosing the solution with a positive quaternion scalar e_0, the tilt quaternion $\underline{\breve{\boldsymbol{e}}}_{HP}$ can be equated to,

$$\underline{\breve{\boldsymbol{e}}}_{HP} = \begin{pmatrix} \sqrt{\frac{f_{\ddot{z}_H}/g+1}{2}} & \frac{f_{\ddot{y}_H}/g}{\sqrt{2\left(f_{\ddot{z}_H}/g+1\right)}} & \frac{-f_{\ddot{x}_H}/g}{\sqrt{2\left(f_{\ddot{z}_H}/g+1\right)}} & 0 \end{pmatrix} \tag{D.6}$$

Equation D.6 is valid for all values, except $f_{\ddot{z}_H}/g = -1$. In order to solve for the singularity, $e_{y_{HP}} = 0$ is set in Equation D.4 to obtain an alternative reduced system [70]:

$$\begin{Bmatrix} f_{\ddot{x}_H}/g \\ f_{\ddot{y}_H}/g \\ f_{\ddot{z}_H}/g \end{Bmatrix} = \begin{Bmatrix} 2\left(e_{x_{HP}}e_{z_{HP}}\right) \\ 2\left(e_{0_{HP}}e_{x_{HP}}\right) \\ e_0^2 - e_{x_{HP}}^2 + e_{z_{HP}}^2 \end{Bmatrix} \tag{D.7}$$

The following equation is the solution to Equation D.7:

$$\underline{\breve{\boldsymbol{e}}}_{HP} = \begin{pmatrix} \frac{-f_{\ddot{y}_H}/g}{\sqrt{2\left(1-f_{\ddot{z}_H}/g\right)}} & \sqrt{\frac{1-f_{\ddot{z}_H}/g}{2}} & 0\frac{f_{\ddot{x}_H}/g}{\sqrt{2\left(1-f_{\ddot{z}_H}/g\right)}} \end{pmatrix} \tag{D.8}$$

Equation D.8 was chosen over other solutions, as it has its singularity at $f_{\ddot{z}_H}/g = 1$. Combining both results, Equations D.6 and D.8, allows both equation's singularity to be mitigate with a margin. This margin helps to avoid numerical instability around both singularities.

$$\underline{\breve{\boldsymbol{e}}}_{HP} = \begin{cases} \begin{pmatrix} \sqrt{\frac{f_{\ddot{z}_H}/g+1}{2}} & \frac{f_{\ddot{y}_H}/g}{\sqrt{2\left(f_{\ddot{z}_H}/g+1\right)}} & \frac{-f_{\ddot{x}_H}/g}{\sqrt{2\left(f_{\ddot{z}_H}/g+1\right)}} & 0 \end{pmatrix} & , f_{\ddot{z}_H} \geq 0 \\ \begin{pmatrix} \frac{-f_{\ddot{y}_H}/g}{\sqrt{2\left(1-f_{\ddot{z}_H}/g\right)}} & \sqrt{\frac{1-f_{\ddot{z}_H}/g}{2}} & 0\frac{f_{\ddot{x}_H}/g}{\sqrt{2\left(1-f_{\ddot{z}_H}/g\right)}} \end{pmatrix} & , f_{\ddot{z}_H} < 0 \end{cases} \tag{D.9}$$

Rather then using a single observation to ascertain the tilt of the system, multiple measurements are used of which the average is taken $\underline{\bar{\boldsymbol{f}}}_H$ and has the following advantages. First, any translational movement of the system is lost in the average given that the system stays in the same place (which can be checked by the visual sensor). Second, the possible influence of the sensor's noise on the tilt correction is reduced by averaging multiple measurements.

Any tilt changes during the calibration process are mitigated by also recording the observed **tilt** attitude of the TrackIR sensor in the same time frame and 'deducting' the resulting average $\underline{\bar{\boldsymbol{e}}}_{HP_m}$ from the gravity based tilt attitude $\underline{\breve{\boldsymbol{e}}}_{HP}$, i.e.,

$$\underline{\boldsymbol{e}}_{HP_\delta} = \underline{\breve{\boldsymbol{e}}}_{HP} \underset{L}{\otimes} \left(\sim \underline{\bar{\boldsymbol{e}}}_{HP_m} \right) \tag{D.10}$$

The average quaternion attitude of the HMD since the initialization of the TrackIR sensor $\underline{\bar{\boldsymbol{e}}}_{HP_m}$ relative to the upper platform reference frame $\mathbb{E}_I$ can be computed via Equation 7.47.

Moreover, the tilt attitude of the TrackIR sensor measurements is acquired by setting the measured Euler attitude yaw to zero before computing the corresponding quaternion attitude and averaging them.

D.3. Unscented Kalman Filter modification

The only step that needs modification to implement the calibration of the TrackIR is the observation function h_{HP} in Equation 5.18. As mentioned, the observation of the TrackIR $\underline{\boldsymbol{y}}_{HP}$ is relative to the zero initialized pose. Thus, the actual pose is the measurements pose plus the offset and can be written as:

$$\begin{gathered} \text{actual} = \text{measurement} + \text{offset} \\ \downarrow \\ \text{measurement} = \text{actual} - \text{offset} \end{gathered}$$

Rewriting Equation 5.10 with the offset in mind results in,

$$\underline{\boldsymbol{y}}_{HP} = \mathrm{h}_{HP} \left(\underline{\boldsymbol{x}}_H(t), \underline{\boldsymbol{x}}_P(t), \underline{\boldsymbol{o}}_H(t) \right) \tag{D.11}$$

$$\begin{bmatrix} \underline{\boldsymbol{c}}_{HP_m} \\ \underline{\boldsymbol{e}}_{HP_m} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{c}}_{HP} - \underline{\boldsymbol{c}}_{HP_\delta} + \underline{\boldsymbol{n}}_{\boldsymbol{c}_H} \\ \underline{\boldsymbol{e}}_{HP} \underset{L}{\otimes} \left(\sim \underline{\boldsymbol{e}}_{HP_\delta} \right) + \underline{\boldsymbol{n}}_{\boldsymbol{e}_H} \end{bmatrix} \tag{D.12}$$

Expanding the HMD's quaternion attitude $\underline{\boldsymbol{e}}_{HP}$ relative to the upper platform of the SIMONA Research Simulator (SRS) using Equation 5.16, i.e.,

$$= \begin{bmatrix} \underline{\boldsymbol{c}}_{HP} - \underline{\boldsymbol{c}}_{HP_\delta} + \underline{\boldsymbol{n}}_{\boldsymbol{c}_H} \\ \underline{\boldsymbol{e}}_{H} \underset{L}{\otimes} \left(\sim \underline{\boldsymbol{e}}_{P} \right) \underset{L}{\otimes} \left(\sim \underline{\boldsymbol{e}}_{HP_\delta} \right) + \underline{\boldsymbol{n}}_{\boldsymbol{e}_H} \end{bmatrix} \tag{D.13}$$

D.4. Conclusion

The duration necessary for the calibration was initially envisioned around one minute to be on the (very) save side. After running this calibration computation a couple times, it can be stated that the calibration duration should depend on the amount of movement during the calibration phase. Given that the SRS is in rest and neither the TrackIR or the HMD's IMU measure motion, calibration durations around $5\,\mathrm{s}$ will give reliable results. (Of note, some of these runs where performed with the HMD strapped to the headrest.)

Care should be taken with the yaw angle of the HMD when initializing the TrackIR 5 sensor, due to the fact that the yaw angle cannot be calibrated with the present method. The HMD should face forward when initializing the TrackIR sensor, as there is no yaw offset.

Moreover, a constant attitude error/offset on the TrackIR 5 sensor would impact its Kalman Filter (KF) corrections, resulting in a HMD attitude error. This in turn would create a pose estimation problem, similar to an IMU alignment issues for the HMD.

E

Noise Parameter Validation through Static Tests

Using the ‘No Movement’ motion profile in Section C.2 various sensors’ noise parameters used in the simulation in Chapter 7 could be analyzed. The values found in this appendix are the ones used in the Paper.

In this appendix, the data recorded in enhanced Communication Abstraction Layer (eCAL) is plotted against the simulated data. For more details on the simulation, see Chapter 7.

E.1. SRS

E.1.1. Linear Encoders

Inspecting the real measurements plotted in Figures E.1 and E.2 leads to three interesting observations.

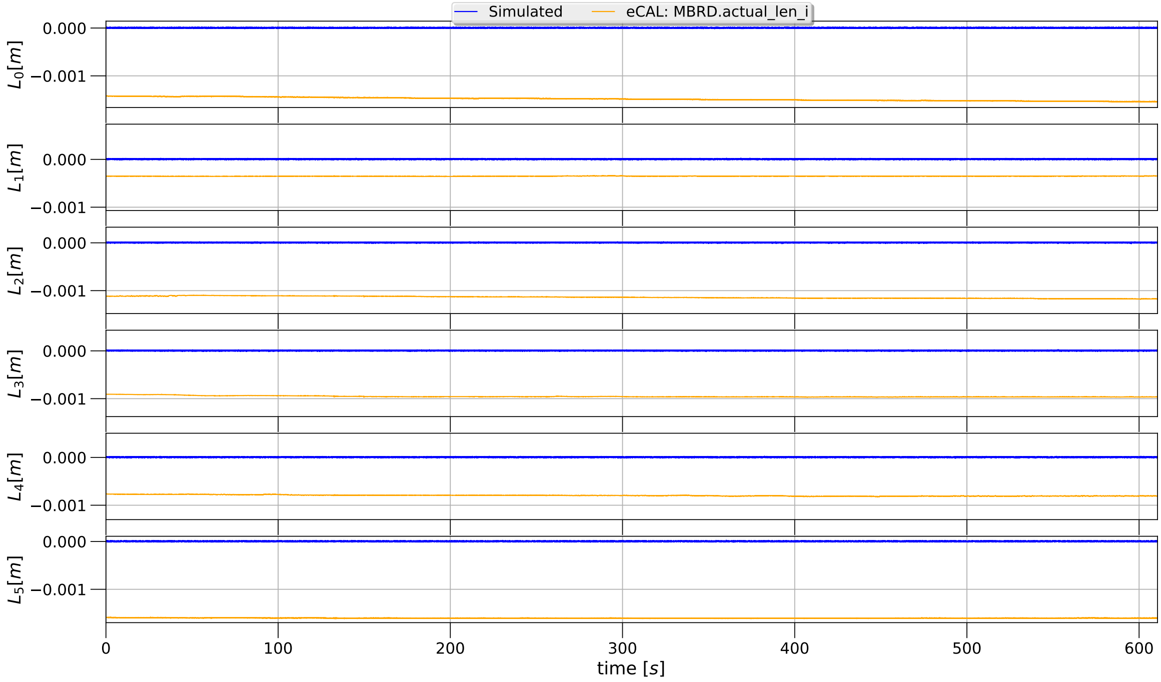


Figure E.1: SRS linear encoders’ measurements during a static test.

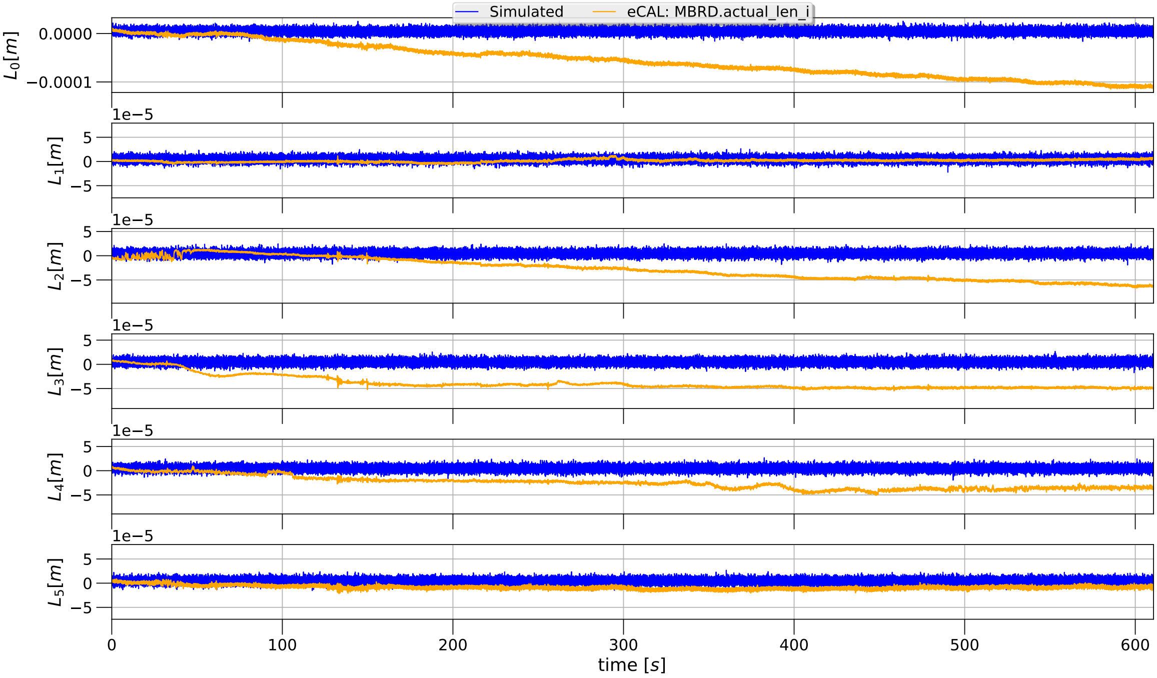


Figure E.2: SRS linear encoders' measurements during a static test (adjusted to align with simulation).

The **first observation** is that the platform is not resting at exactly the zero-position. This has no further consequence for the simulation or algorithm.

The **second observation** is that the linear encoders reported a downward trend over time, in all but l_{m1} and l_{m5}. This effect is most likely caused by the reduction of the hydraulic oil operating temperature due to the tested motion, i.e., 'No Movement'. This in turn resulted in a minor reduction of max 0.1 mm of hydraulic actuator length, correctly reported by the linear encoders. This has no further consequence for the simulation or algorithm.

The **third observation** is that the Standard Deviation (SD) of the measurements were slightly lower then the simulated values. This was further highlighted in Table E.1, where the expected and the measured SD are collected. The actual SD of each linear encoder is probably between 2×10^{-6} m to 3×10^{-6} m. The computation of the SD were hindered by the wandering values and some irregularities at the beginning of the measurement, see l_{m2} in Figure E.2.

Table E.1: SRS linear encoders' SD of the first 50 s of a static test, as shown in Figure E.1.

	L_0	L_1	L_2	L_3	L_4	L_5	
Expected SD:	5.00	5.00	5.00	5.00	5.00	5.00	($\times 10^{-6}$ m)
Measured SD:	3.52	1.94	6.78	5.61	2.69	3.31	($\times 10^{-6}$ m)

As discussed in the sensitivity analysis, this would have minor implications. The ν would be less then expected if the noise parameters of the linear encoders are adapted. Meaning that the algorithm should thrust the linear encoder observations more.

E.1.2. Accelerometer

The Accelerometer of the SIMONA Research Simulator (SRS) is plotted during standstill for approximately 10 min in Figure E.3. Here, two observation stand out.

First, norm of the signal is larger then the expected 9.81 m/s^2. Assuming this is the result of biases, the algorithm should have no problem estimating these.

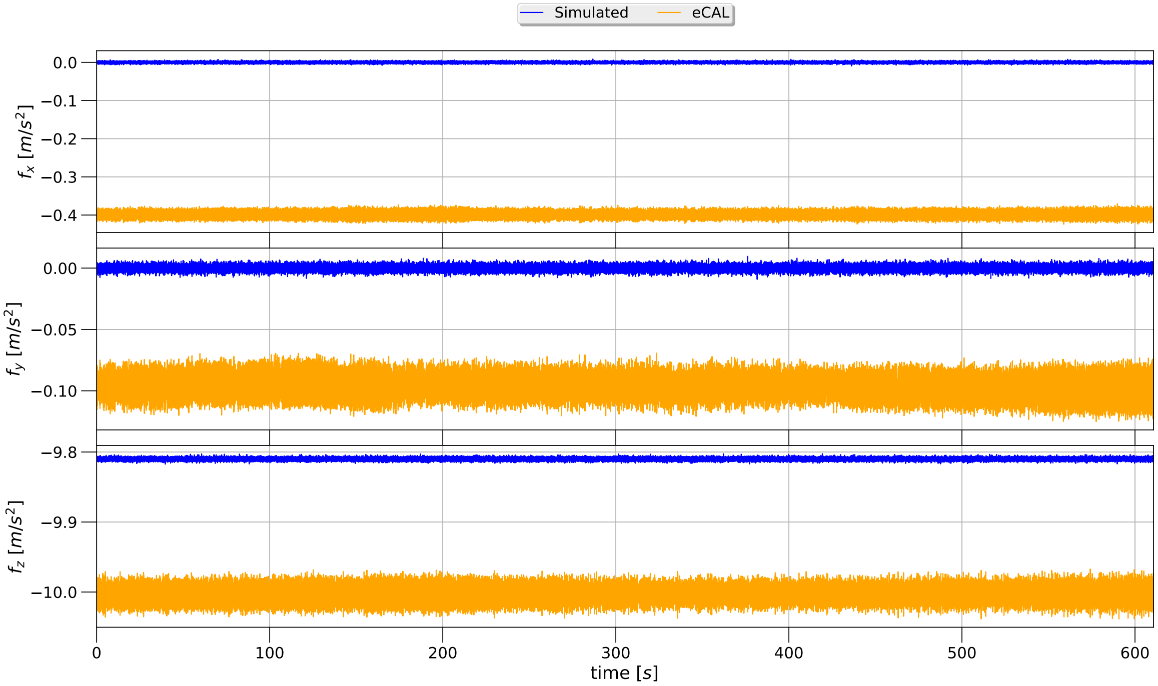


Figure E.3: SRS accelerometer's measurement during a static test.

Second, the SRS accelerometer's SD, presented in Table E.2, highlight a discrepancy between the expected and measured SD. From the sensitivity analysis, it was clear that the major performance hurdle with the SRS Inertial Measurement Unit (IMU) implementation was the 100 Hz publishing rate. However, the increase in noise of the accelerometer causes the algorithm to trust the output of the IMU to much and this should be accounted for.

Table E.2: SRS accelerometer SD measured from the data presented in Figure E.3.

	f_x	f_y	f_z	
Expected SD:	1.90	2.10	1.80	($\times 10^{-3}$ m/s^2)
Measured SD:	8.10	9.31	12.09	($\times 10^{-3}$ m/s^2)

E.1.3. Gyroscope

The SRS's gyroscope is plotted during the ≈ 10 min standstill in Figure E.4.

In this figure, the bias of the gyro is clearly shown, but can be estimated in the algorithm and should thus be of no further consequence. From this plot nothing can be said about any potential misalignment.

However, in the SRS's gyroscope case, the expected noise parameters were overestimated by quite the margin. The SD's are collected in Table E.3. Deducing from the sensitivity analysis, the decrease in noise present of the gyroscope would have a minor but positive influence.

Table E.3: SRS gyroscope SD measured from the data presented in Figure E.4.

	p	q	r	
Expected SD:	24.00	41.00	33.00	($\times 10^{-4}$ rad/s)
Measured SD:	1.41	1.69	1.73	($\times 10^{-4}$ rad/s)

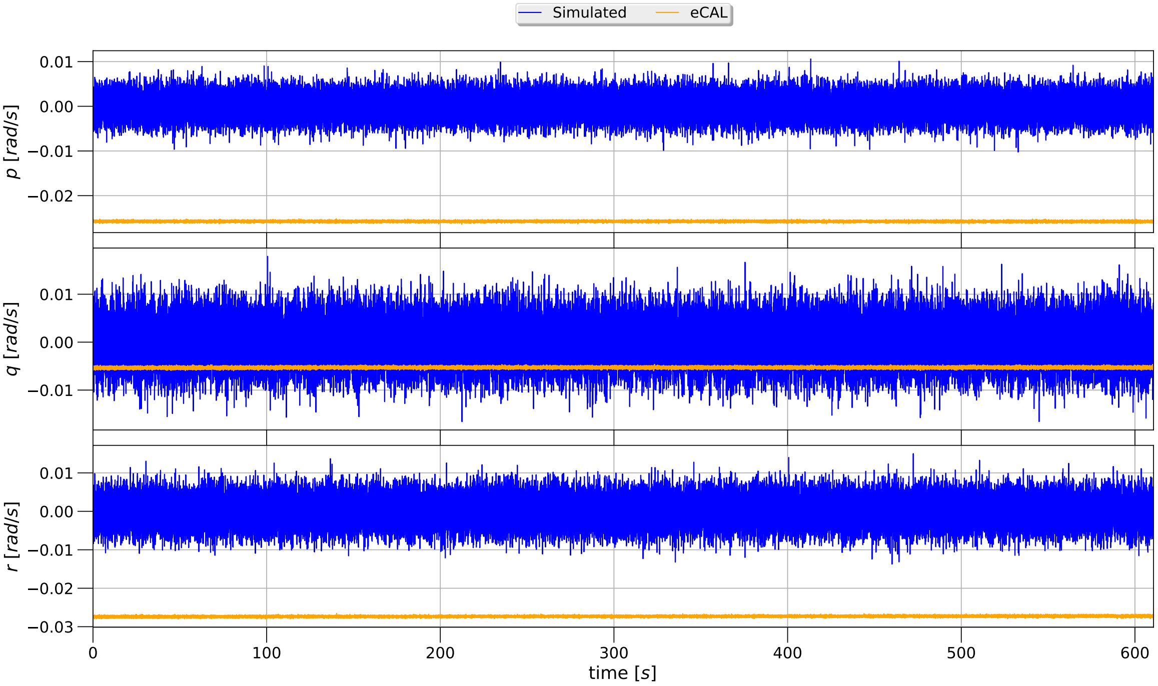


Figure E.4: SRS gyroscope's measurement during a static test.

E.2. HMD

E.2.1. TrackIR 5

Figures E.5 and E.6 show the measurements of the 10 min standstill. The drifting of the output is evident from these figures. A possible explanation is that the Head-Mounted Display (HMD) settles into place onto the headrest and/or in software, as this was the first experiment after all. This finding is reinforced by subsequent TrackIR data further into the test. Moreover, in these Figures E.5 and E.6 and Tables E.4 and E.5 show both an overestimation regarding the TrackIR noise parameters.

The sensitivity analysis showed pose estimation improvements due to the use of the algorithm versus only the visual tracking regardless of the proposed sensor quality of the TrackIR. However, with the TrackIR, setting the correct expected noise parameter did have a relative large impact on the pose estimation and 'condition' of the algorithm, i.e., a ν value within expected bounds.

Table E.4: TrackIR 5 positional SD measured over the first 50 s in Figure E.5.

	x	y	z	
Expected SD:	9.19	30.40	49.40	($\times 10^{-5}$ m)
Measured SD:	2.75	3.58	3.69	($\times 10^{-5}$ m)

Table E.5: TrackIR 5 attitudinal SD measured over the first 50 s in Figure E.6.

	ϕ	θ	ψ	
Expected SD:	19.40	13.00	24.70	($\times 10^{-4}$ rad)
Measured SD:	2.26	1.02	2.15	($\times 10^{-4}$ rad)

E.2.2. Accelerometer

The mean of the norm of the data presented in Figure E.7 resulted in $9.84\,\mathrm{m/s^2}$, close to the expected $9.81\,\mathrm{m/s^2}$. Moreover, the apparent bias present on the x-axis is actually the result of the pitch down resting position of the HMD.

The resulting SD from this dataset are collected in Table E.6. These are close to the expected

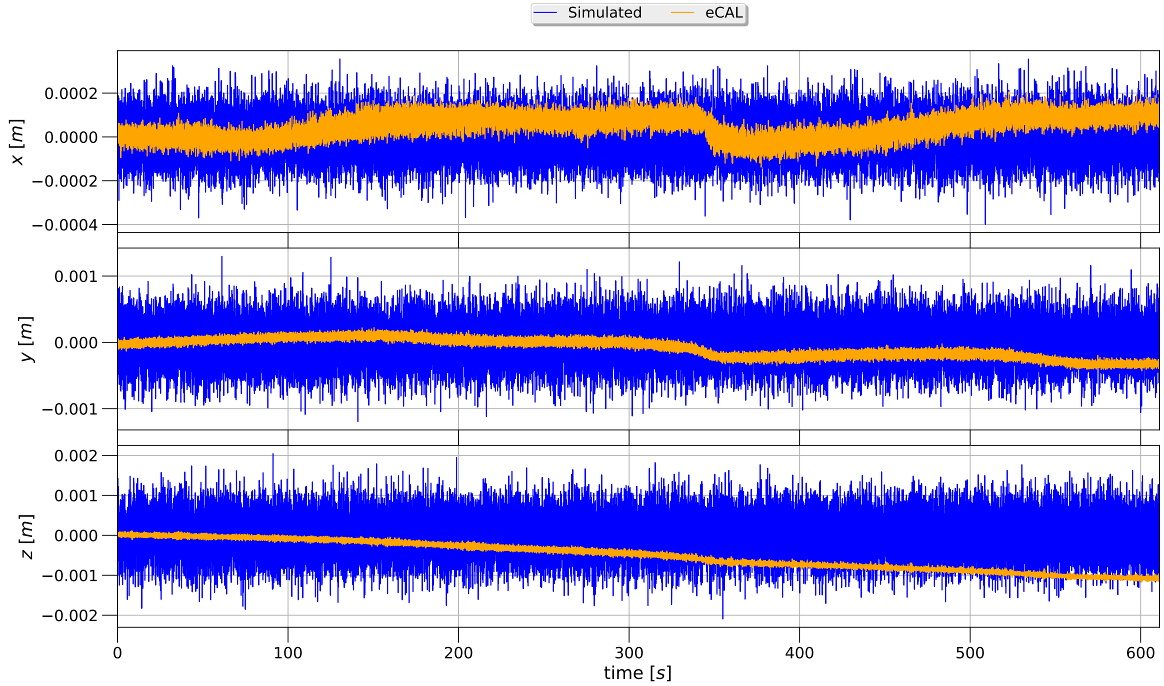


Figure E.5: TrackIR 5 position during a static test.

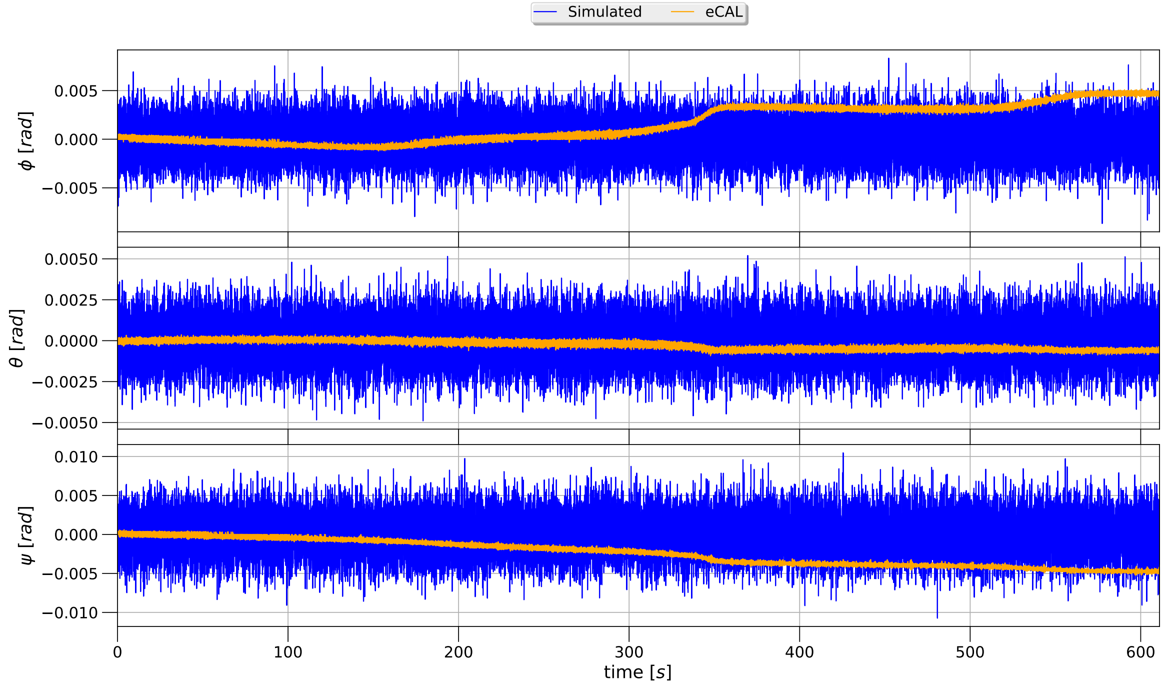


Figure E.6: TrackIR 5 attitude during a static test.

values, but consistently underestimated. The original method to estimate the noise parameters, the Overlapping Allan Deviation, also took Brownian noise into account and could account for some of the discrepancy found.

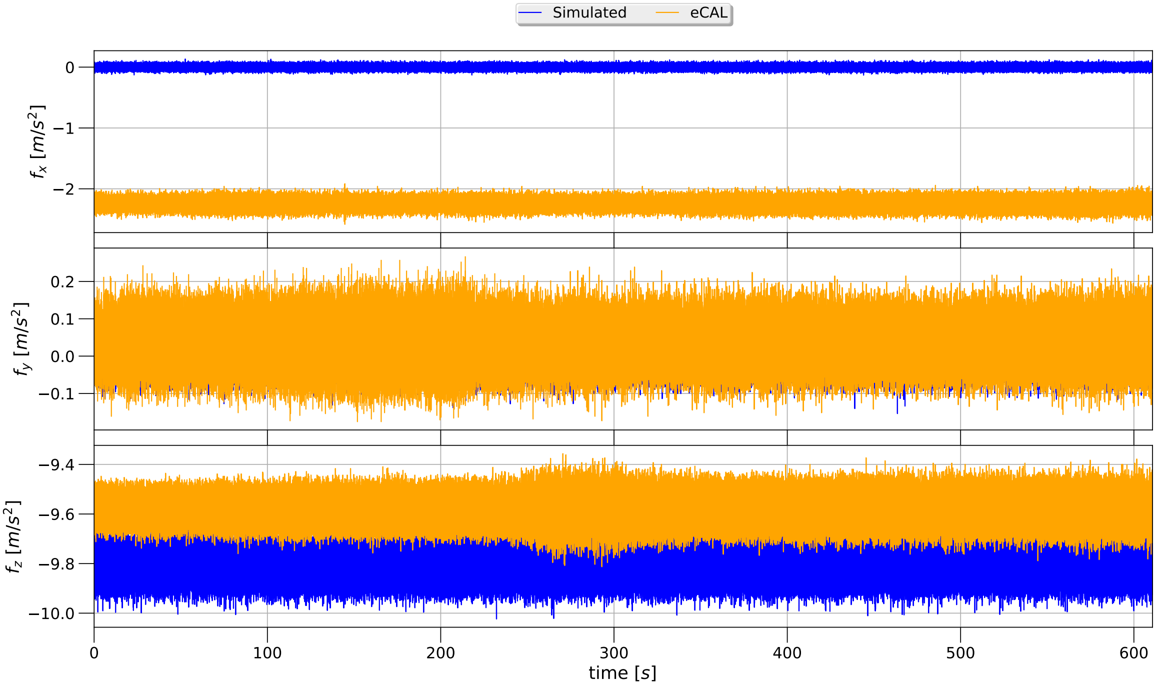


Figure E.7: HMD accelerometer data during the static test.

Table E.6: HMD accelerometer's SD measured from the data presented in Figure E.7.

	f_x	f_y	f_z	
Expected SD:	3.00	2.90	4.70	$(\times 10^{-2}\,\text{m/s}^2)$
Measured SD:	7.51	5.32	5.17	$(\times 10^{-2}\,\text{m/s}^2)$

E.2.3. Gyroscope

The data of the HMD's gyroscope is plotted in Figure E.8. The bias present over ≈ 10 min in Figure E.8 is small and can be estimated by the algorithm.

The SD's presented in Table E.7 indicate that the expected values were good.

Table E.7: HMD gyroscope SD measured from the data presented in Figure E.8.

	p	q	r	
Expected SD:	3.30	3.60	3.80	$(\times 10^{-3}\,\text{rad/s})$
Measured SD:	4.22	3.60	3.00	$(\times 10^{-3}\,\text{rad/s})$

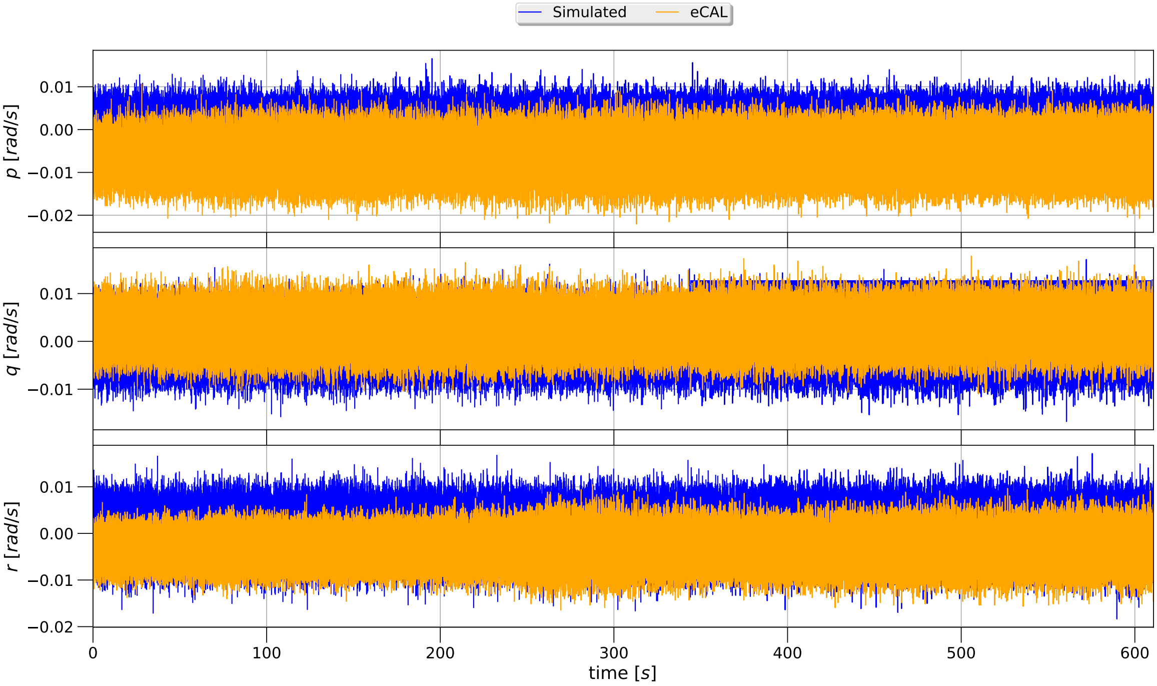


Figure E.8: HMD gyroscope measurement during the static test.

F

TrackIR 5 Delay

The proprietary software of the TrackIR 5 sensor obfuscates the raw sensor measurements. The data used in this appendix was based on the measurements done while the SIMONA Research Simulator (SRS) was at standstill, and the user wearing the Head-Mounted Display (HMD) was rotating their head around x, y, and z in sequence. The resulting data of the HMD's gyroscope was integrated and subsequently compared in Figure F.1 to the TrackIR 5 reported data.

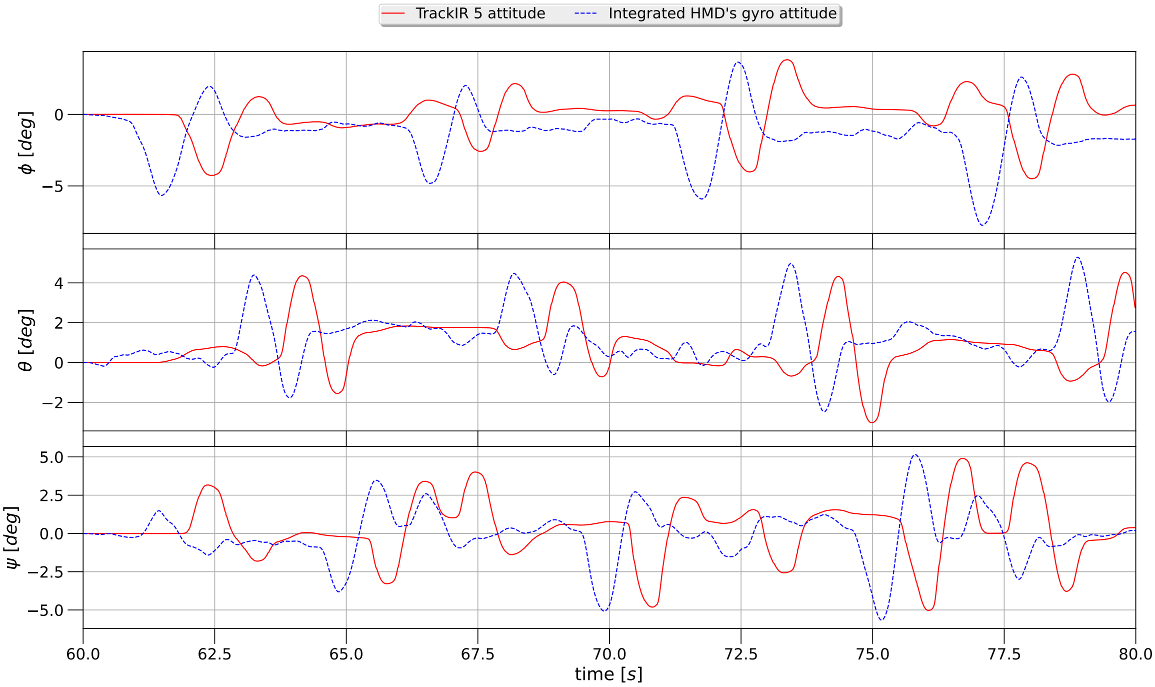


Figure F.1: Integrated

In this figure it can be observed that the TrackIR 5 has a approximately one second delay to the integrated gyro attitude. This one second delay of the TrackIR 5 renders its measurements mostly obsolete.

This delay is too large for the delay compensation method described in Section 3.3 due to the large state and its covariance matrix and its resulting computational requirement. In order to reduce the impact of this delay in sensor data, it was opted to increase the TrackIR noise parameters to reduce the trust the algorithm placed in the TrackIR 5 sensor's measurements in favor of the Inertial Measurement Unit (IMU) based prediction.

Moreover, the resulting attitude of the TrackIR 5 looks smoothed when compared to the HMD's gyroscope based attitude. This behavior makes sense when considering its original purpose, to measure and send the head rotation to a Personal Computer (PC) based game. These head rotation would be scaled and used in-game to change the viewport into the game. In this use case a stable, i.e., smooth, viewport was probably more important than a precise one.

G

Recording Timestamp Issue

While recording the data with the layout described in Appendix C.1, a timing issue on the recording side happened. The issue concerns the data gathered by ethernet connection, i.e., the SIMONA Research Simulator (SRS) related data.

The difference between timestamps of the incoming data, as recorded by enhanced Communication Abstraction Layer (eCAL), are plotted in Figure G.1. Here, the messages' average and most delta times Δt were equal to the expected value of 10 ms, i.e., 100 Hz. However, the timing in certain instances would vary between 0 ms to 24 ms as shown in Figure G.1. Other recordings were checked for this behavior with similar results as presented here.

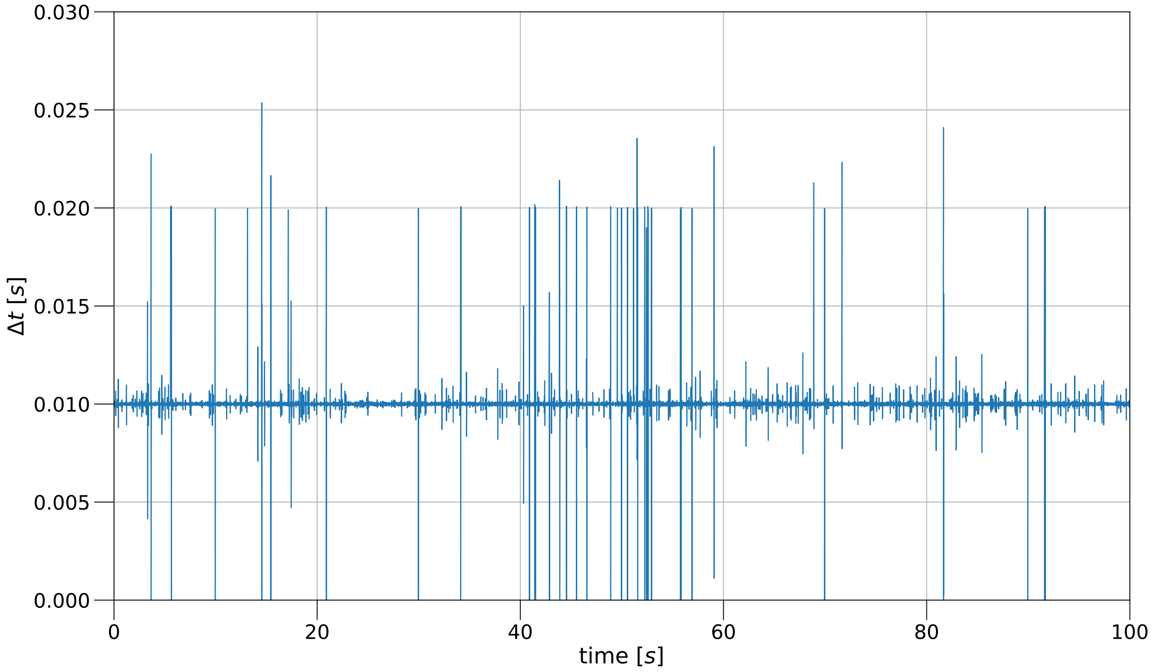


Figure G.1: Time difference Δt in seconds between recorded data messages of the SRS of the 'Complex SRS motion'.

This issue is likely due to the packet coalescing setting of the recording Personal Computer (PC)'s network card, hindering the timely arrival of the various data messages. While using packet coalescing, a reduction in system interrupts is achieved through coalescing, i.e., combining multiple data packages received through the ethernet port [42]. This resulted in eCAL receiving these messages at a non-constant time interval.

H

Final Results

In this appendix the most interesting plots are shown based on the data gathered with the method described in Appendix C and the state estimated by the relevant algorithm described by the section headings. The noise and starting values used in these results are presented in Section H.1.

H.1. Parameters and Initial State

The reasoning for the parameters and initial states listed can be found in the paper in part I.

Table H.1: HMD's IMU noise parameters

Description	Var.	Value	Unit	
$\underline{\boldsymbol{n}}_{f_H} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{f_H}\right)$ Accelerometer's Velocity Random Walk	$\sigma_{f_{x_H}}$ $\sigma_{f_{y_H}}$ $\sigma_{f_{z_H}}$	8.26 5.85 5.60	$\Big\}\times10^{-2}$ m/s^2	
$\underline{\boldsymbol{\mu}}_{f_H} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\mu_{f_H}}\right)$ Accelerometer's Acceleration Random Walk	$\sigma_{\mu_{f_{\square_H}}}$	1.00	$\times10^{-4}$ m/s^3	$\forall\square = x, y, z$
$\underline{\boldsymbol{n}}_{\omega_H} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\omega_H}\right)$ Gyroscope's Angle Random Walk	σ_{p_H} σ_{q_H} σ_{r_H}	4.64 3.96 3.30	$\Big\}\times10^{-3}$ rad/s	
$\underline{\boldsymbol{\mu}}_{\omega_H} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\mu_{\omega_H}}\right)$ Gyroscope's Rate Random Walk	$\sigma_{\mu_{\square_H}}$	1.00	$\times10^{-5}$ rad/s^2	$\forall\square = p, q, r$

The estimated initial stochastic state $\hat{\underline{\boldsymbol{x}}}_0$ of Configuration 1, where only the Head-Mounted Display (HMD) has an Inertial Measurement Unit (IMU), is set in Equation H.1. The initial position of the SIMONA Research Simulator (SRS) was set to its neutral position and the HMD's position was set to that of the captain's headrest. Both quaternion attitudes were set to an identity rotation, and all other values were set to zero.

$$\hat{\underline{\boldsymbol{x}}}_0 = E\left|\underline{\boldsymbol{x}}_0\right| = E\left|\begin{pmatrix}\underline{\boldsymbol{x}}_{P_0}^T & \underline{\boldsymbol{x}}_{H_0}^T\end{pmatrix}^T\right| \tag{H.1}$$

$$\begin{Bmatrix}\hat{\underline{\boldsymbol{x}}}_{P_0}\\ \hat{\underline{\boldsymbol{x}}}_{H_0}\end{Bmatrix} = \begin{Bmatrix}\begin{pmatrix}\mathbf{0}_{2\times} & -2.39 & \mathbf{0}_{3\times} & 1 & \mathbf{0}_{6\times}\end{pmatrix}^T\\ \begin{pmatrix}0 & -0.55 & -3.5975 & \mathbf{0}_{3\times} & 1 & \mathbf{0}_{9\times}\end{pmatrix}^T\end{Bmatrix} \tag{H.2}$$

The covariance of the estimated initial stochastic state $\boldsymbol{P}_{\hat{x}_0}$ of Configuration 1 is shown in Equation H.3.

Table H.2: SRS Process noise parameters

Description	Var.	Value	Unit	
Configuration 1:				
$\underline{\boldsymbol{\nu}}_{\ddot{c}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\ddot{c}_P}\right)$ Acceleration Deviation	$\sigma_{\square_P}$	1.5	m/s^2	$\forall\square = \ddot{x}, \ddot{y}, \ddot{z}$
$\underline{\boldsymbol{\nu}}_{\dot{\omega}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}\right)$ Angular Acceleration Deviation	$\sigma_{\square_P}$	1.5	rad/s^2	$\forall\square = \dot{p}, \dot{q}, \dot{r}$
Configuration 2:				
$\underline{\boldsymbol{n}}_{f_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{f_P}\right)$ Accelerometer's Velocity Random Walk	$\sigma_{f_{\square_P}}$	4.0×10^{-2}	m/s^2	$\forall\square = x, y, z$
$\underline{\boldsymbol{\mu}}_{f_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\mu_{f_P}}\right)$ Accelerometer's Acceleration Random Walk	$\sigma_{\mu_{f_{\square_P}}}$	1.0×10^{-4}	m/s^3	
$\underline{\boldsymbol{n}}_{\omega_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\omega_P}\right)$ Gyroscope's Angle Random Walk	$\sigma_{\square_P}$	8.0×10^{-3}	rad/s	$\forall\square = p, q, r$
$\underline{\boldsymbol{\mu}}_{\omega_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\mu_{\omega_P}}\right)$ Gyroscope's Rate Random Walk	$\sigma_{\mu_{\square_P}}$	1.0×10^{-5}	rad/s^2	
Configuration 3:				
$\underline{\boldsymbol{\eta}}_{\tau_{\ddot{c}}} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\tau_{\ddot{c}_P}}\right)$ Translational Lag Deviation	$\sigma_{\tau_\square}$	1.0×10^{-3}	s	$\forall\square = \ddot{x}, \ddot{y}, \ddot{z}$
$\underline{\boldsymbol{\nu}}_{\ddot{c}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\ddot{c}_P}\right)$ Acceleration Deviation	$\sigma_{\square_P}$	5.0×10^{-1}	m/s^2	
$\underline{\boldsymbol{\eta}}_{\tau_{\dot{\omega}}} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\tau_{\dot{\omega}_P}}\right)$ Rotational Lag Deviation	$\sigma_{\tau_\square}$	1.0×10^{-3}	s	$\forall\square = \dot{p}, \dot{q}, \dot{r}$
$\underline{\boldsymbol{\nu}}_{\dot{\omega}_P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P}\right)$ Angular Acceleration Deviation	$\sigma_{\square_P}$	5.0×10^{-1}	rad/s^2	

Table H.3: Observation sensors noise parameters

Description	Var.	Value	Unit	
$n_{l_i} \approx \aleph\left(0, \sigma_{l_i}\right),\ \sigma_{l_i} \in \underline{\boldsymbol{\sigma}}_l$ Absolute Linear Encoders Deviation	σ_{l_i}	5.0×10^{-6}	m	$\forall i = 1, \dots, 6$
TrackIR 5 Deviation: Positional: $\underline{\boldsymbol{n}}_{c_{HP}} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{c_{HP}}\right)$	$\sigma_{\square_{HP}}$	1.5×10^{-3}	m	$\forall\square = x, y, z$
Attitudinal: $\underline{\boldsymbol{n}}_{e_H^P} \approx \aleph\left(0, \underline{\boldsymbol{\sigma}}_{e_H^P}\right)$	$\sigma_{\square_{HP}}$	5.0×10^{-3}	rad	$\forall\square = \phi, \theta, \psi$

The deviation in the expected initial stochastic state are all set independent to one another.

$$\boldsymbol{P}_{\hat{x}_0} = E \left| \left\{ \underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0 \right\} \left\{ \underline{\boldsymbol{x}}_0 - \hat{\underline{\boldsymbol{x}}}_0 \right\}^T \right| \tag{H.3}$$

$$\begin{bmatrix} \boldsymbol{P}_{\hat{x}_{P_0}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{P}_{\hat{x}_{H_0}} \end{bmatrix} = \begin{bmatrix} \text{diag}(\mathbf{1}_{6\times}\ \mathbf{0.44}_{3\times}\ \mathbf{1}_{3\times})^2 & \mathbf{0}_{12\times 15} \\ \mathbf{0}_{15\times 12} & \text{diag}(\mathbf{1}_{6\times}\ \mathbf{0.52}_{3\times}\ \mathbf{0.1}_{3\times}\ \mathbf{0.05}_{3\times})^2 \end{bmatrix} \tag{H.4}$$

Taking the deviation values presented in Table H.1, H.2, and H.3 into account, the process $\boldsymbol{Q}$ and observation noise matrices $\boldsymbol{R}$ can be constructed as follows:

$$\boldsymbol{Q} = \text{diag}(\underline{\boldsymbol{\sigma}}_P \quad \underline{\boldsymbol{\sigma}}_H)^2 \tag{H.5}$$

$$= \text{diag}\left(\underline{\boldsymbol{\sigma}}_{\ddot{c}_P} \quad \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P} \quad \underline{\boldsymbol{\sigma}}_{f_H} \quad \underline{\boldsymbol{\sigma}}_{\mu_{f_H}} \quad \underline{\boldsymbol{\sigma}}_{\omega_H} \quad \underline{\boldsymbol{\sigma}}_{\mu_{\omega_H}}\right)^2$$

$$\boldsymbol{R} = \text{diag}\left(\underline{\boldsymbol{\sigma}}_l \quad \underline{\boldsymbol{\sigma}}_{c_{HP}} \quad \underline{\boldsymbol{\sigma}}_{e_H^P}\right)^2 \tag{H.6}$$

For Configuration 2, where both the SRS and HMD have an IMU, only the SRS's sub-system changes, thus resulting in:

$$\hat{\underline{\boldsymbol{x}}}_{P_0} = (\mathbf{0}_{2\times} \quad -2.39 \quad \mathbf{0}_{3\times} \quad 1 \quad \mathbf{0}_{9\times})^T \tag{H.7}$$

$$\boldsymbol{P}_{\hat{x}_{P_0}} = \text{diag}(\mathbf{1}_{6\times} \quad \mathbf{0.44}_{3\times} \quad \mathbf{0.1}_{3\times} \quad \mathbf{0.05}_{3\times})^2 \tag{H.8}$$

$$\boldsymbol{Q} = \text{diag}\left(\underline{\boldsymbol{\sigma}}_{f_P} \quad \underline{\boldsymbol{\sigma}}_{\mu_{f_P}} \quad \underline{\boldsymbol{\sigma}}_{\omega_P} \quad \underline{\boldsymbol{\sigma}}_{\mu_{\omega_P}} \quad \underline{\boldsymbol{\sigma}}_H\right)^2 \tag{H.9}$$

For Configuration 3, where setpoints are used rather than an IMU for the SRS, the setpoint related values are printed. Here, attention should be paid when selecting a lag starting value and its corresponding covariance. Choosing a initial lag value close to zero with a large covariance could cause a sigma-point to be sampled below zero causing unstable behavior. Here, it was opted to choose the initial lag variance too small to increase robustness at the cost of convergence duration of said lag.

$$\hat{\underline{\boldsymbol{x}}}_{P_0} = (\mathbf{0}_{2\times} \quad -2.39 \quad \mathbf{0}_{6\times} \quad 1 \quad \mathbf{0}_{9\times} \quad \mathbf{0.03}_{6\times})^T \tag{H.10}$$

$$\boldsymbol{P}_{\hat{x}_{P_0}} = \text{diag}(\mathbf{1}_{9\times} \quad \mathbf{0.44}_{3\times} \quad \mathbf{1}_{6\times} \quad \mathbf{10^{-5}}_{6\times})^2 \tag{H.11}$$

$$\boldsymbol{Q} = \text{diag}\left(\underline{\boldsymbol{\sigma}}_{\tau_{\ddot{c}_P}} \quad \underline{\boldsymbol{\sigma}}_{\ddot{c}_P} \quad \underline{\boldsymbol{\sigma}}_{\tau_{\dot{\omega}_P}} \quad \underline{\boldsymbol{\sigma}}_{\dot{\omega}_P} \quad \underline{\boldsymbol{\sigma}}_H\right)^2 \tag{H.12}$$

Finally, the sigma parameters (α, β, κ) were chosen as follows:

$$\alpha = 0.01, \quad \beta = 2, \quad \kappa = 100 \tag{H.13}$$

κ is chosen larger than the recommended value [22] to increase mathematical stability in case of Configuration 3 with the small initial lag variance. These values were kept constant for all configurations.

H.2. SRS Complex Motion

In this section, the states and innovation per configuration of the motion described in 'Complex Motion — SRS Motion' in Appendix C.2 are plotted.

This motion profile is the most complex and most firm of all motion profiles tested. Therefore, this motion profile will stress the nonlinearities of the Stewart platform and maximize the integration error in the process models.

Note the 43 s mark in most plots in this section showing the biggest impact of the timing issue discussed in Appendix G.

H.2.1. Configuration 1 --- only HMD with IMU

SRS

From Figures H.1 and H.3 the motion profile can be ascertained via the position and attitude of the SRS.

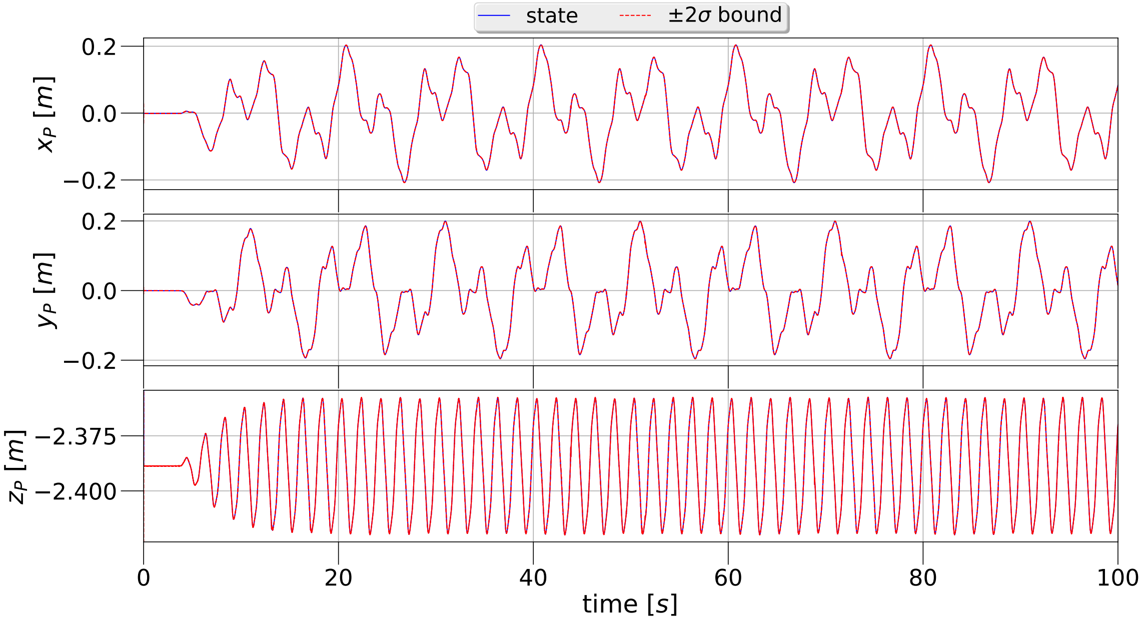


Figure H.1: SRS's position in the inertial reference frame $\mathbb{E}_I$.

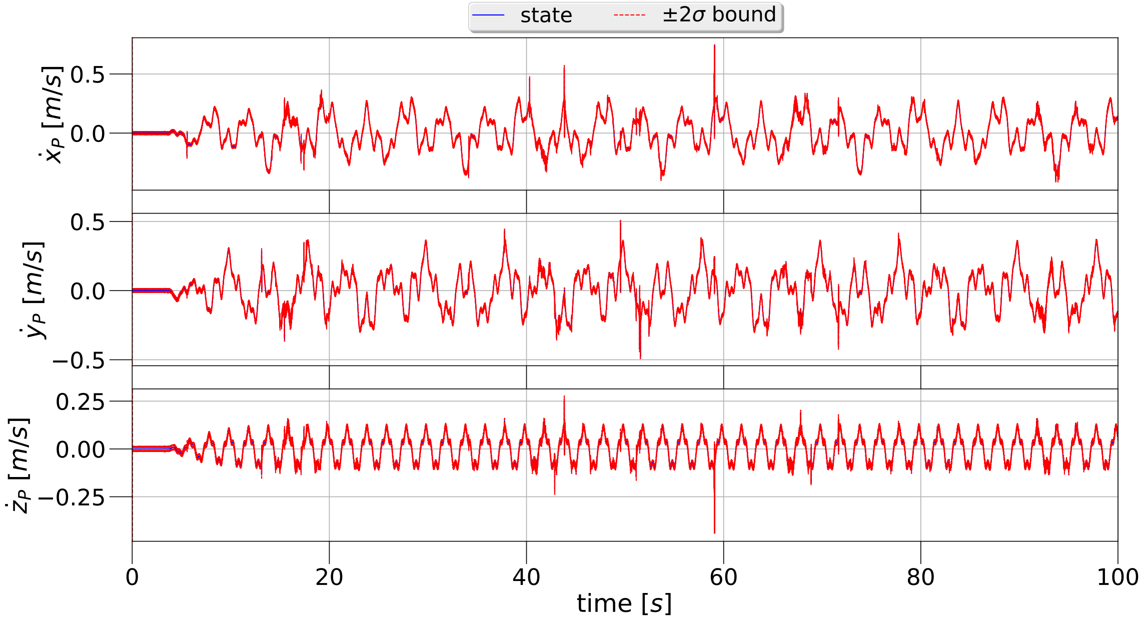


Figure H.2: SRS's velocity in the inertial reference frame $\mathbb{E}_I$.

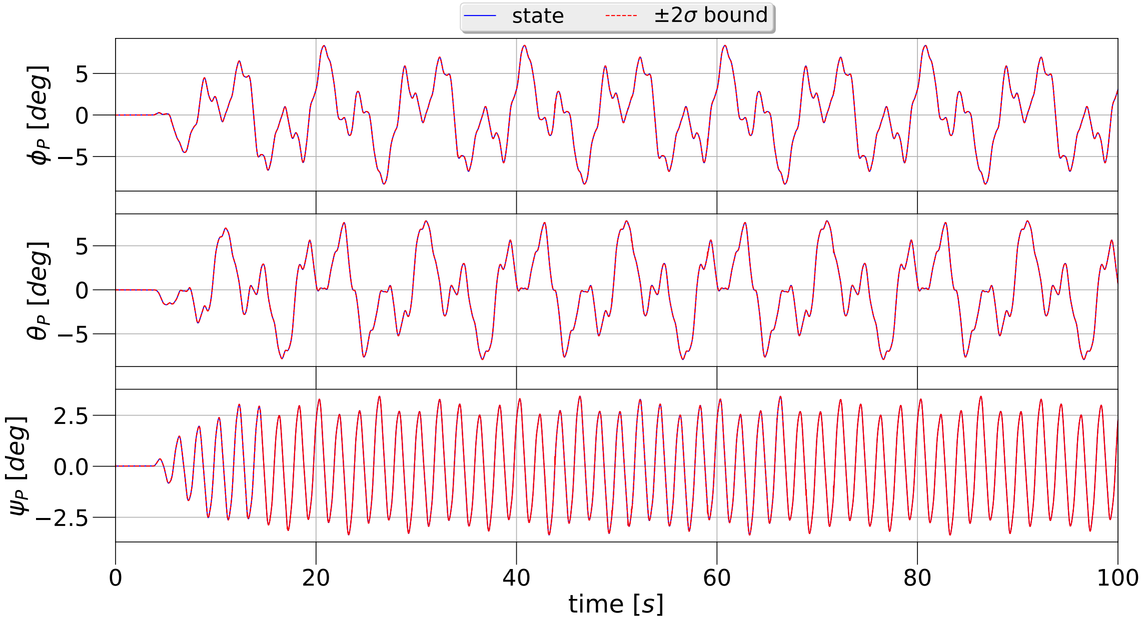


Figure H.3: SRS's Euler attitude w.r.t. the inertial reference frame $\mathbb{E}_I$.

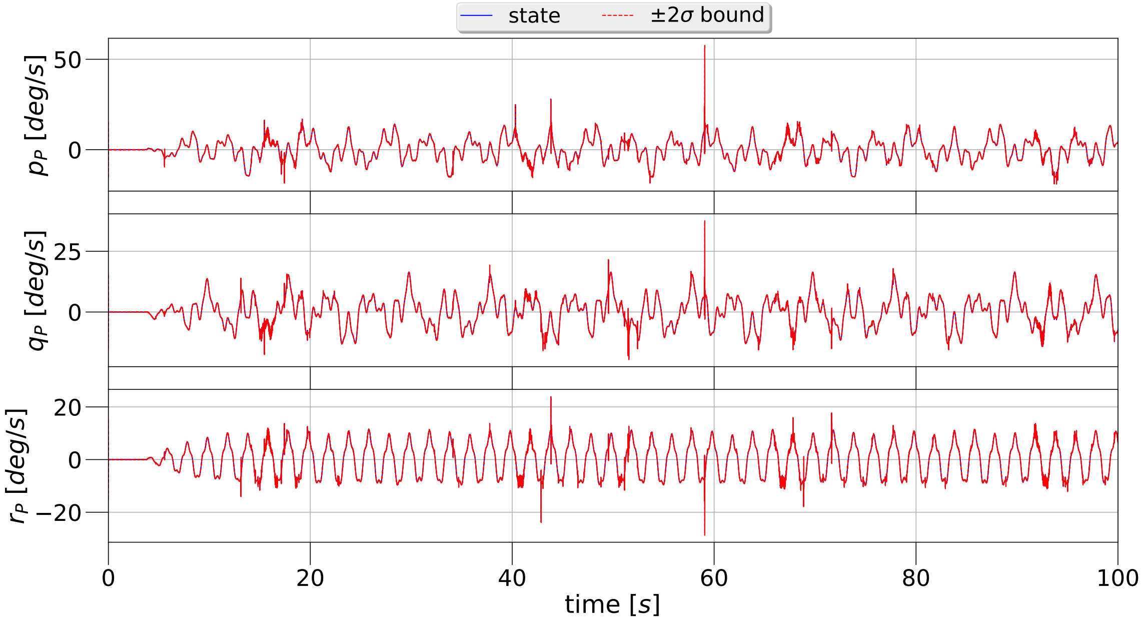


Figure H.4: SRS's angular rate in the upper moving reference frame $\mathbb{E}_P$.

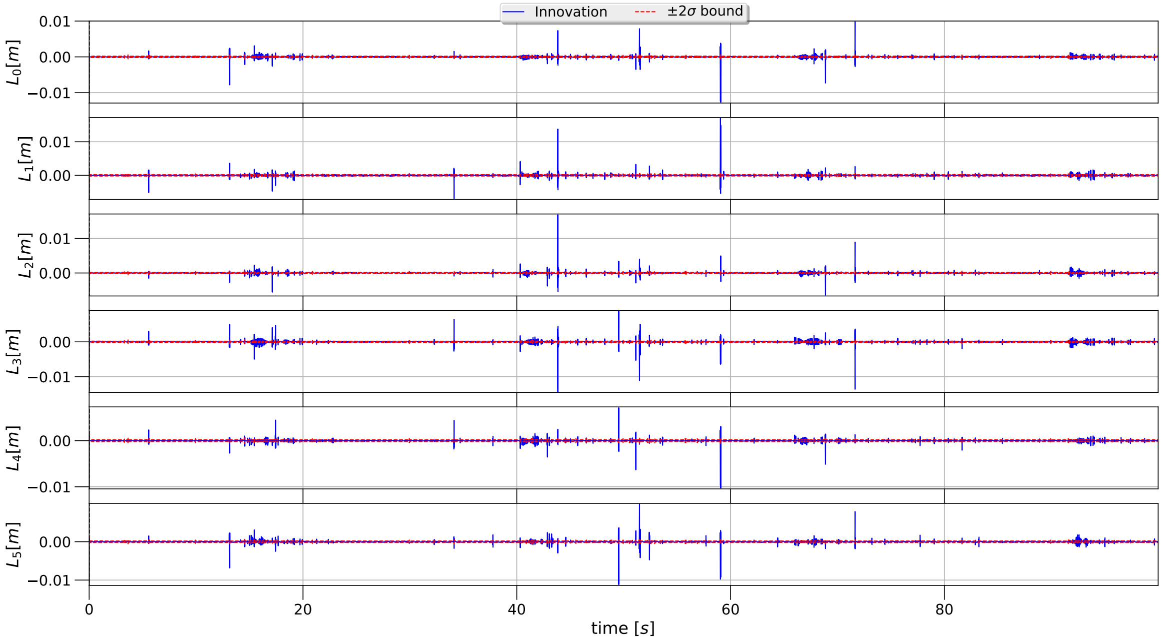


Figure H.5: SRS's linear encoders innovation.

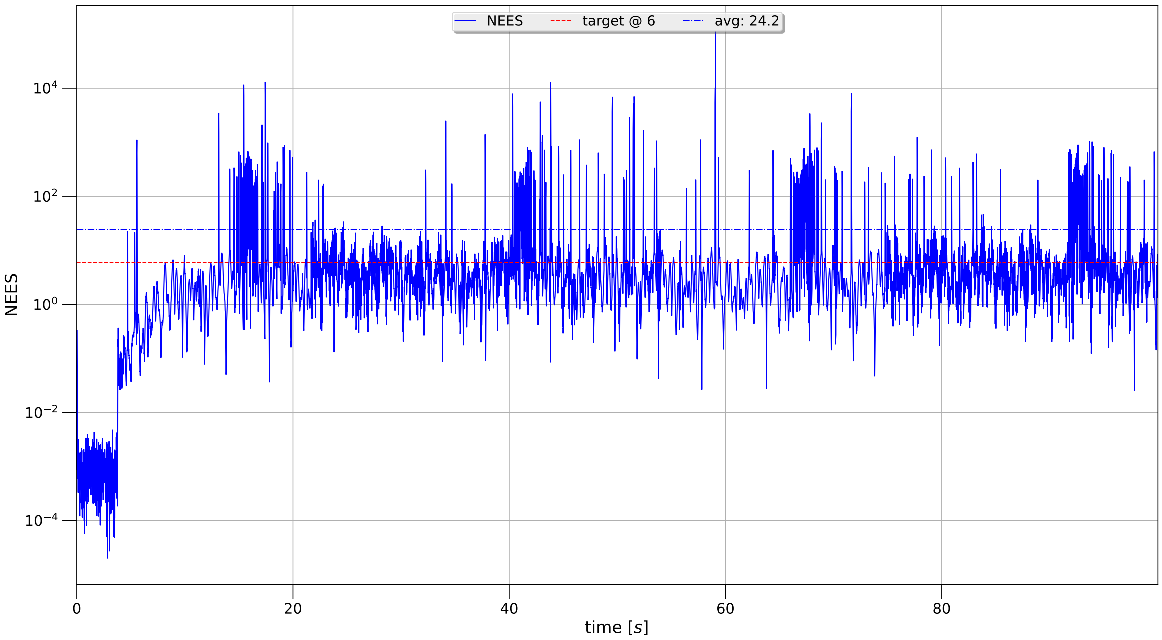


Figure H.6: SRS's linear encoders innovation .

HMD

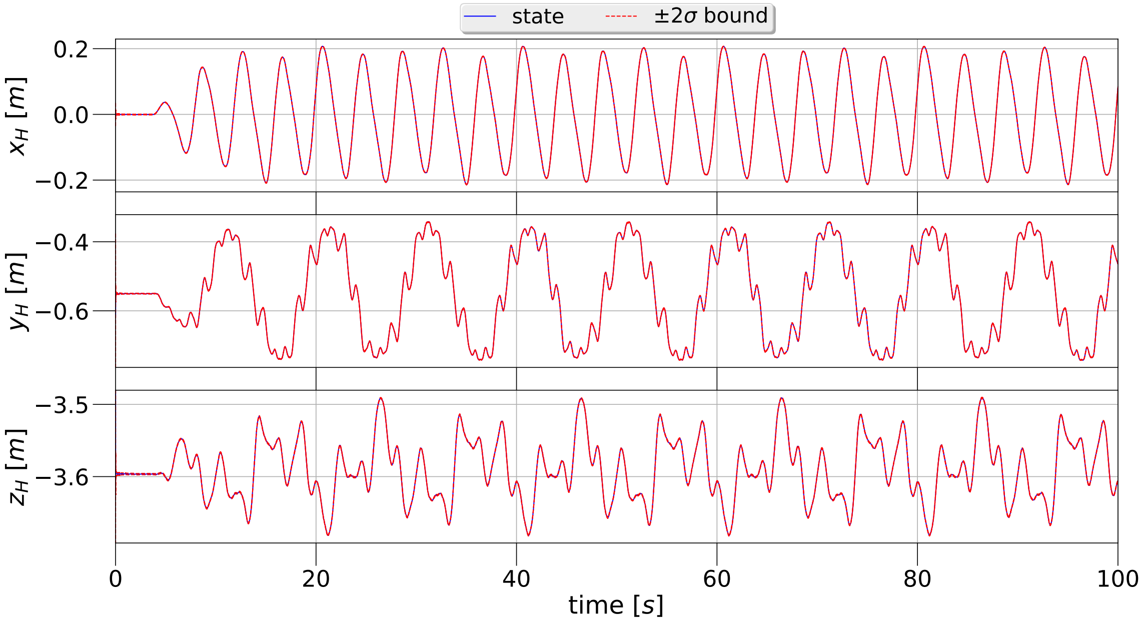


Figure H.7: HMD's position in the inertial reference frame $\mathbb{E}_I$.

With the firm motion profile the chair, headrest, and HMD still move. These small vehicle-fixed movements can be seen in Figures H.10 and H.11

The second of delay on the TrackIR 5 measurements combined with the minor movements the HMD made while strapped to the chair resulted in the innovations plotted in Figures H.14 and H.15. In these figures, the underlying motion profile is visible and thus colored, i.e., non-Gaussian.

The TrackIR 5 noise parameters were tuned such that the ν were close to the targeted values, i.e., six. This caused the Unscented Kalman Filter (UKF) algorithm to trust the measurements of the TrackIR 5 less and the predictions more.

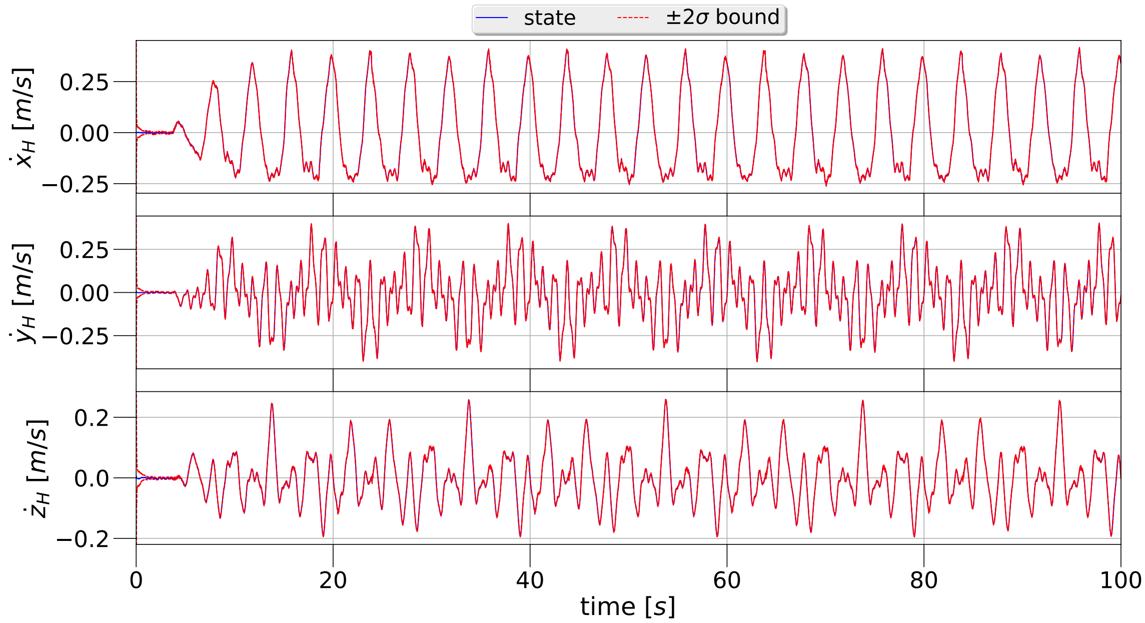


Figure H.8: HMD's velocity in the inertial reference frame $\mathbb{E}_I$.

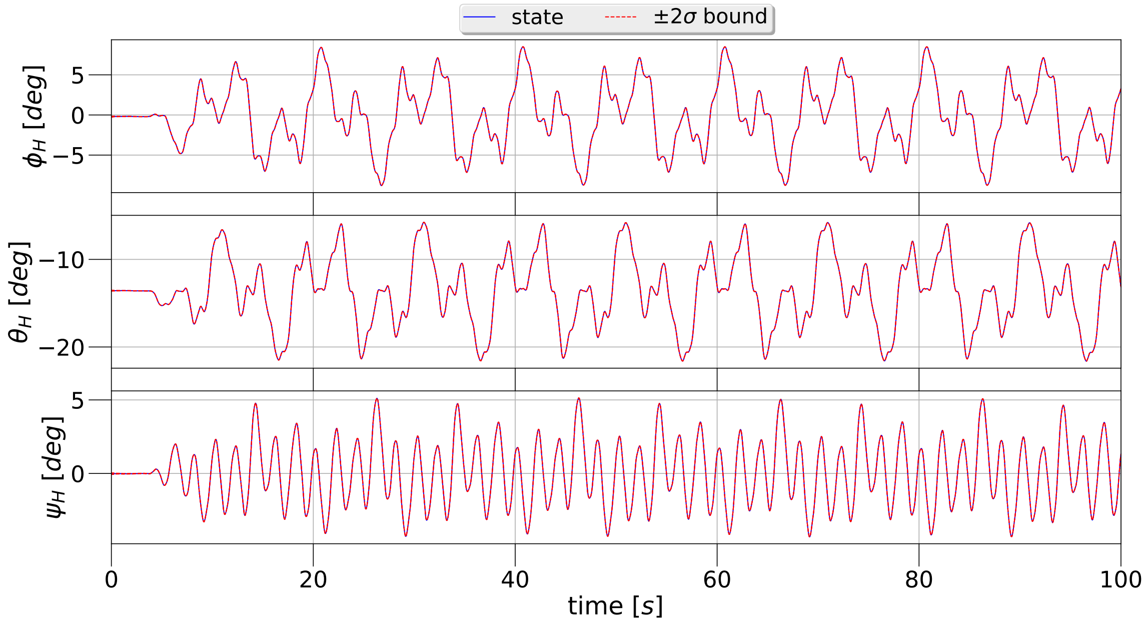


Figure H.9: HMD's Euler attitude w.r.t. the inertial reference frame $\mathbb{E}_I$.

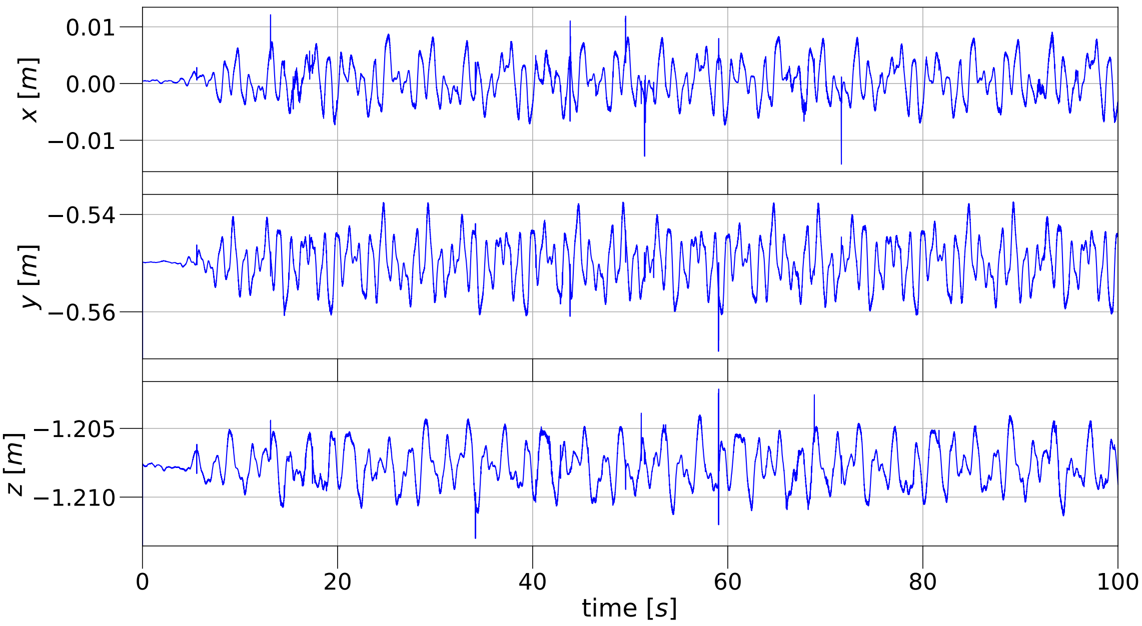


Figure H.10: HMD's relative position to the UGP in the upper moving reference frame $\mathbb{E}_P$.

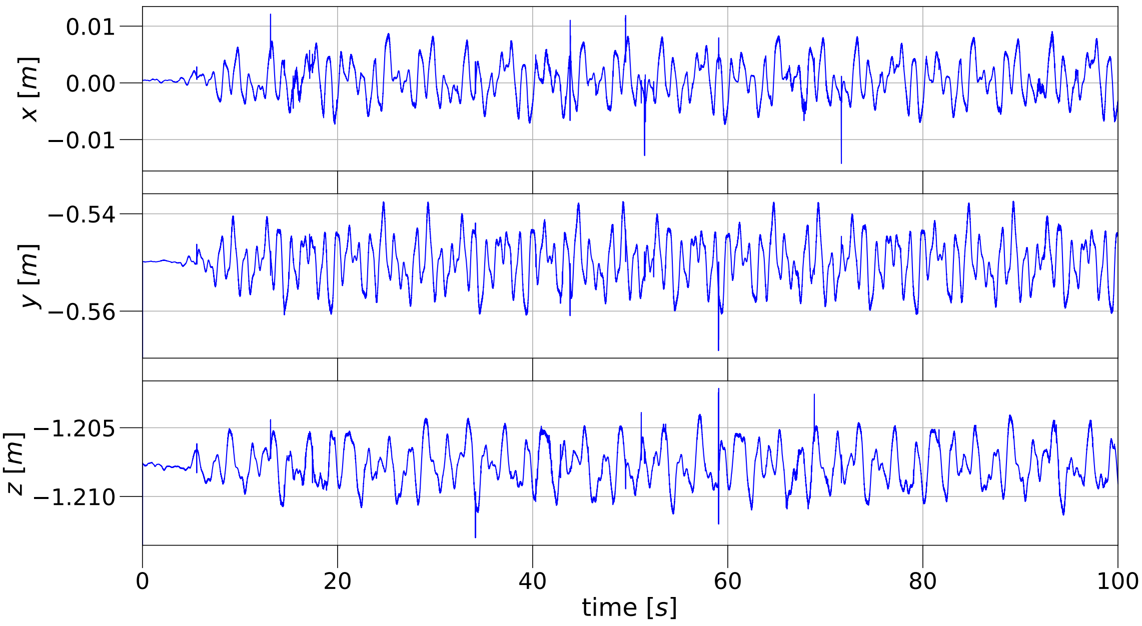


Figure H.11: HMD's relative Euler attitude w.r.t. the upper moving reference frame $\mathbb{E}_P$.

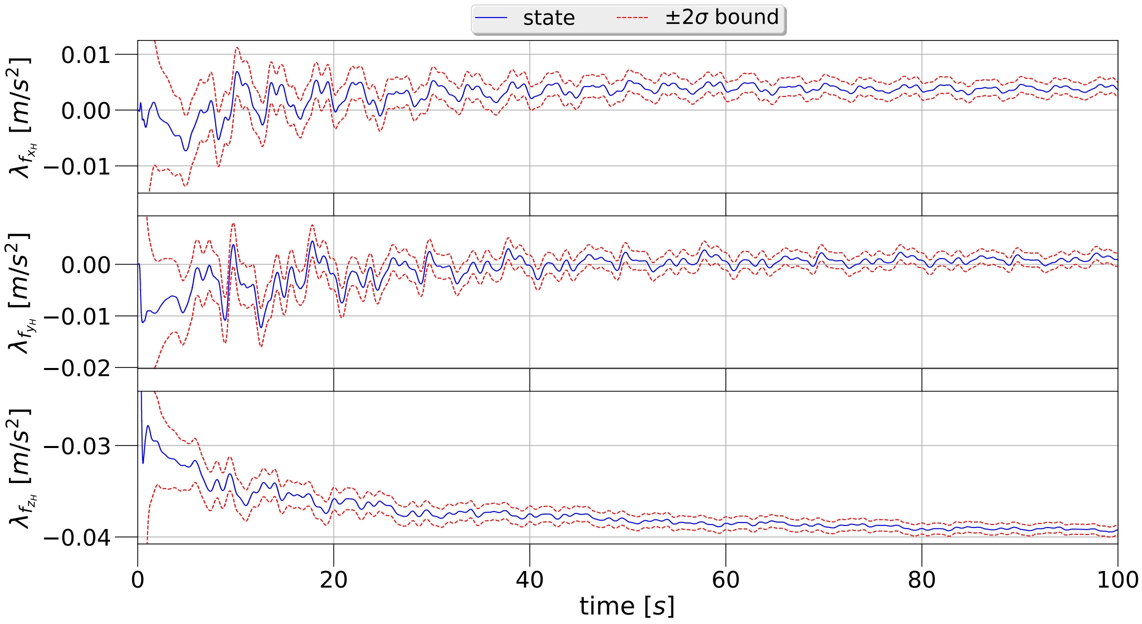


Figure H.12: HMD's accelerometer bias in the HMD reference frame $\mathbb{E}_H$.

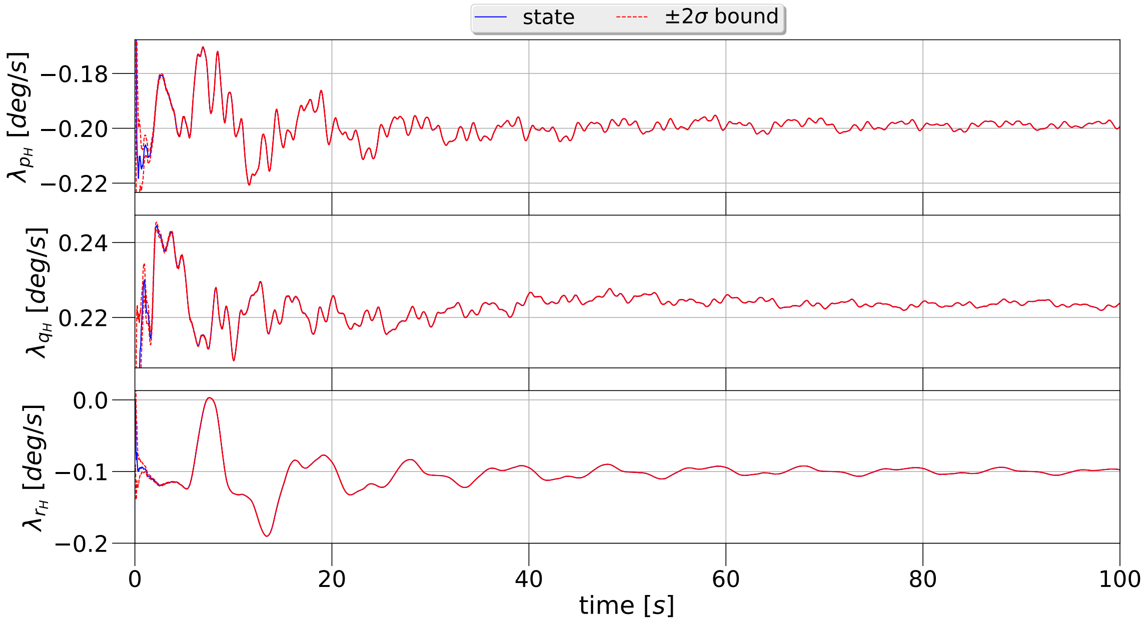


Figure H.13: HMD's gyroscope bias in the HMD reference frame $\mathbb{E}_H$.

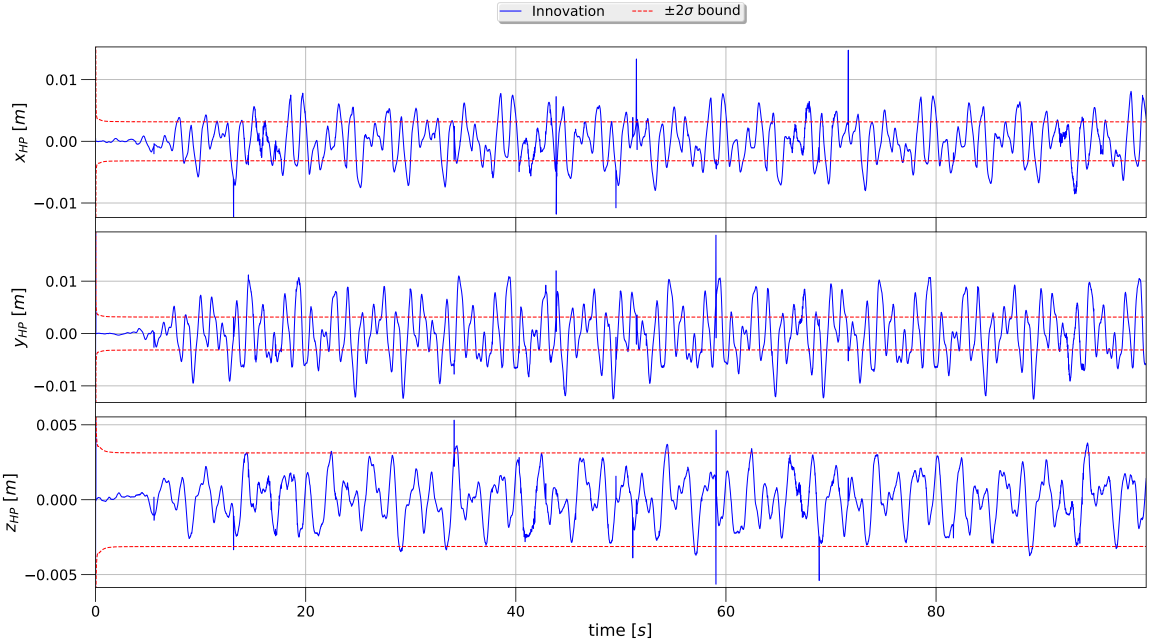


Figure H.14: TrackIR 5 position based innovation.

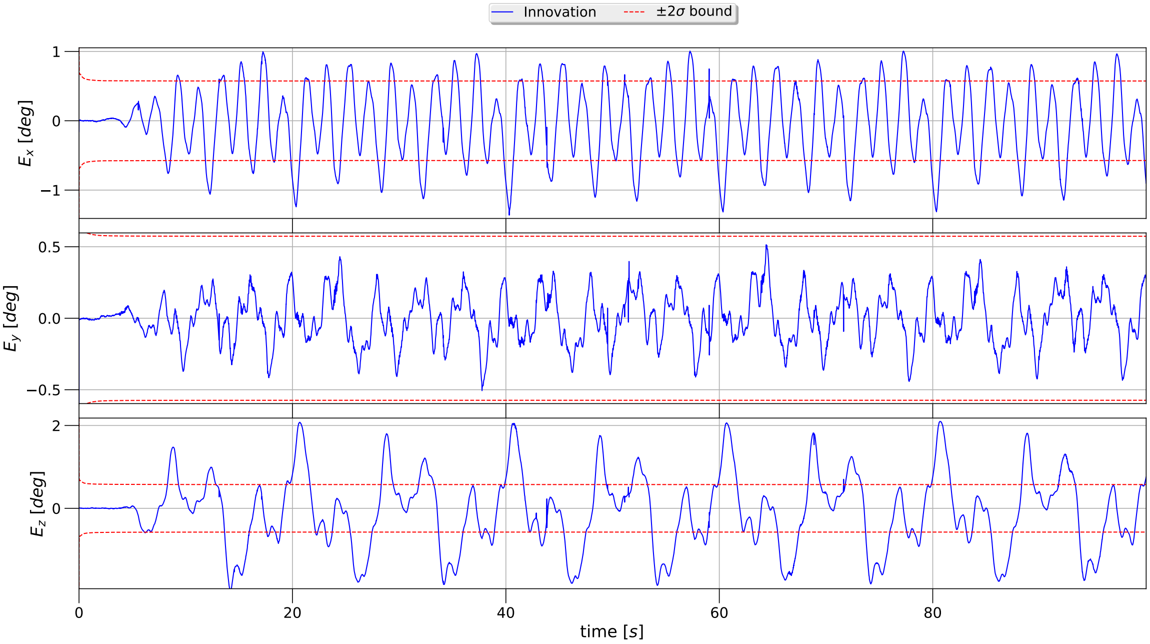


Figure H.15: TrackIR 5 attitude based innovation.

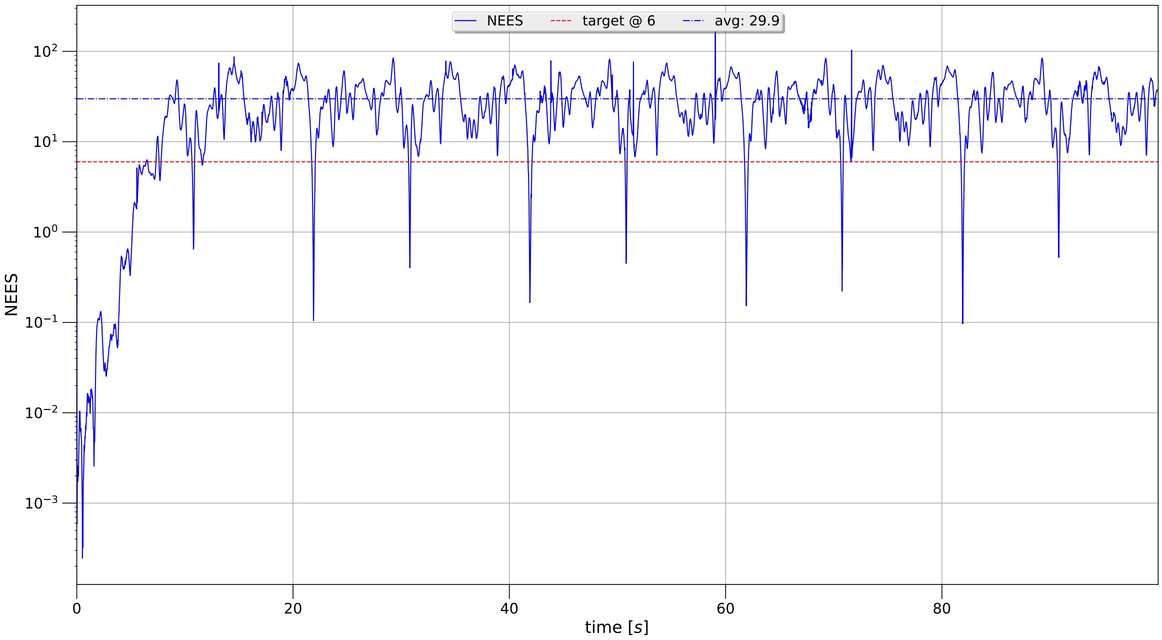


Figure H.16: TrackIR 5 innovation .

H.2.2. Configuration 2 --- both HMD and HMD with IMU

Due to the fact that the motion is the same, the plots already presented in the previous section (H.2.1) are not shown here due to their similarity. However, the linear encoders innovation is shown again as these plots are different enough. The plots of note are the SRS IMU biases.

SRS

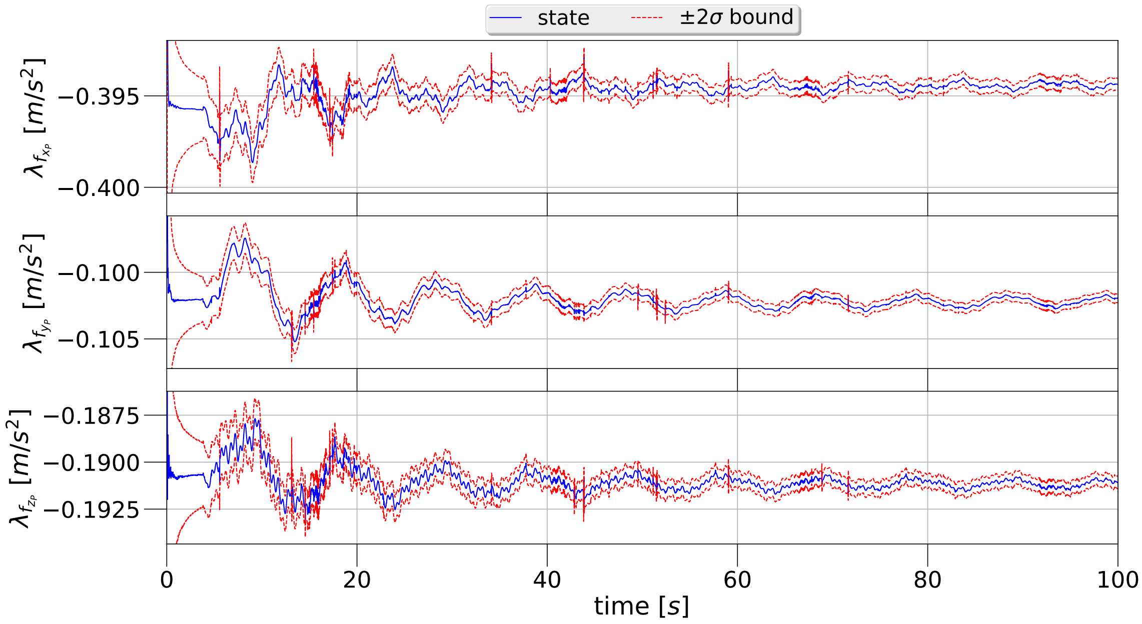


Figure H.17: SRS's accelerometer bias in the upper moving reference frame $\mathbb{E}_P$.

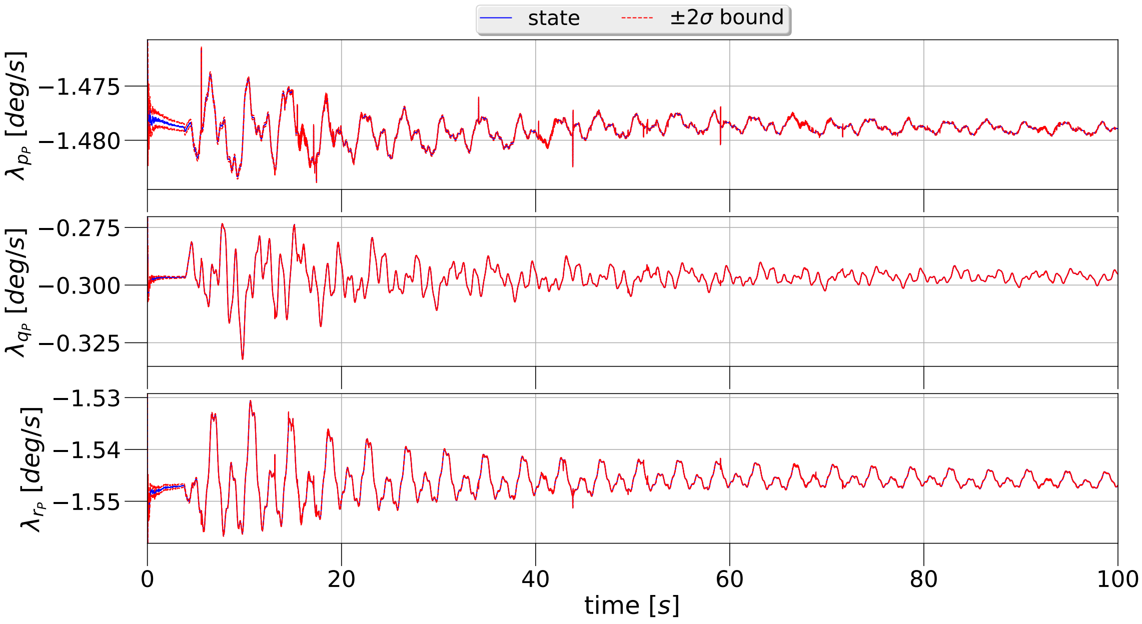


Figure H.18: SRS's gyroscope bias in the upper moving reference frame $\mathbb{E}_P$.

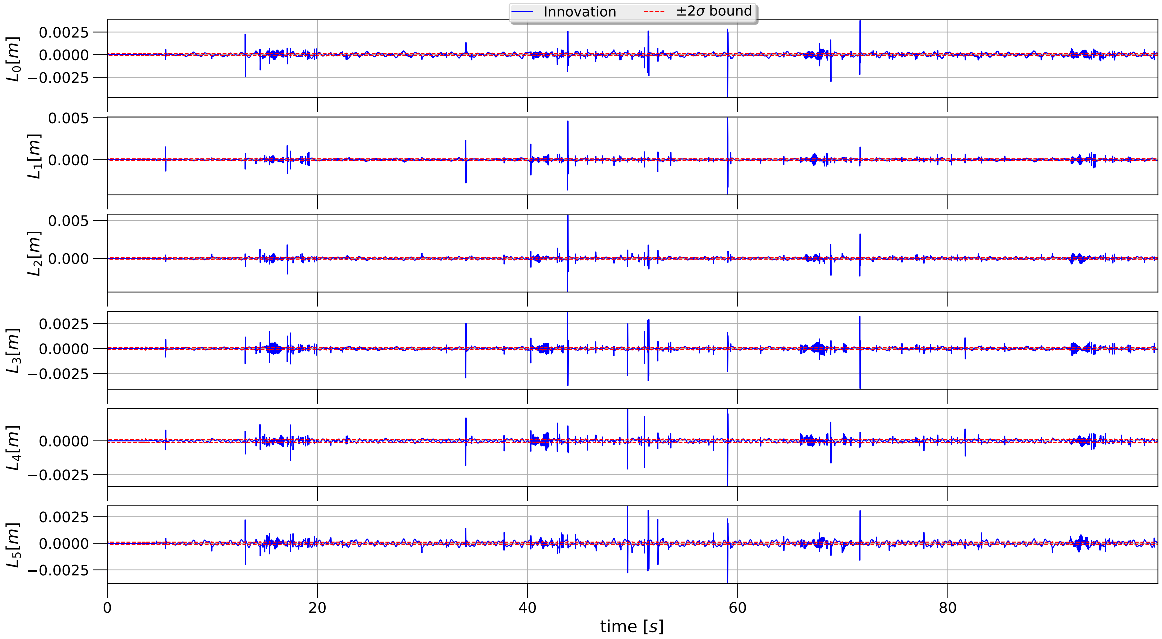


Figure H.19: SRS's linear encoders innovation.

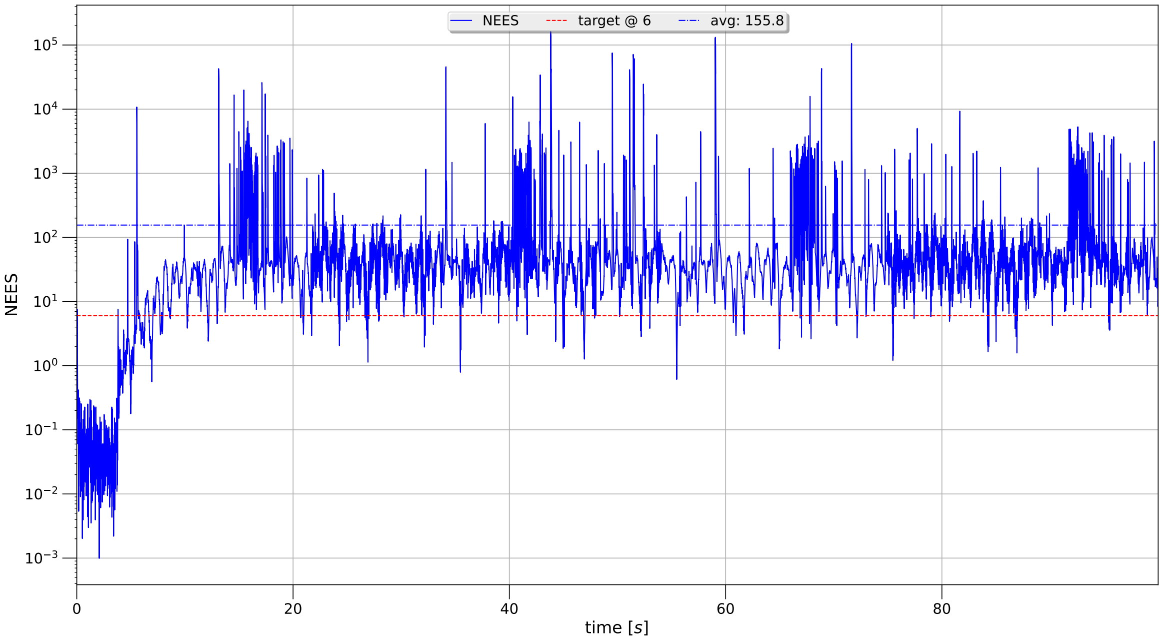


Figure H.20: SRS's linear encoders innovation .

H.2.3. Configuration 3 --- HMD with IMU and SRS with Setpoints

Due to the fact that the motion is the same, the plots already presented in the previous sections (H.2.1, H.2.2) are not shown here due to their similarity. However, the linear encoders innovation is shown again as these plots are different enough. The plots of note are the SRS lags on the setpoints.

SRS

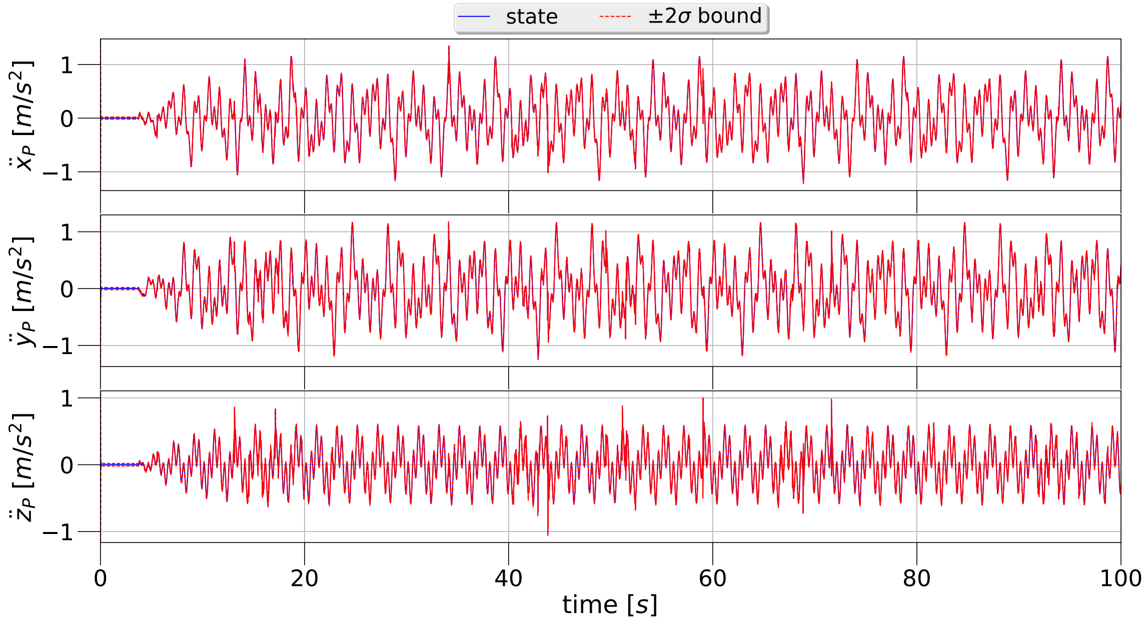


Figure H.21: SRS's acceleration in the inertial reference frame $\mathbb{E}_I$.

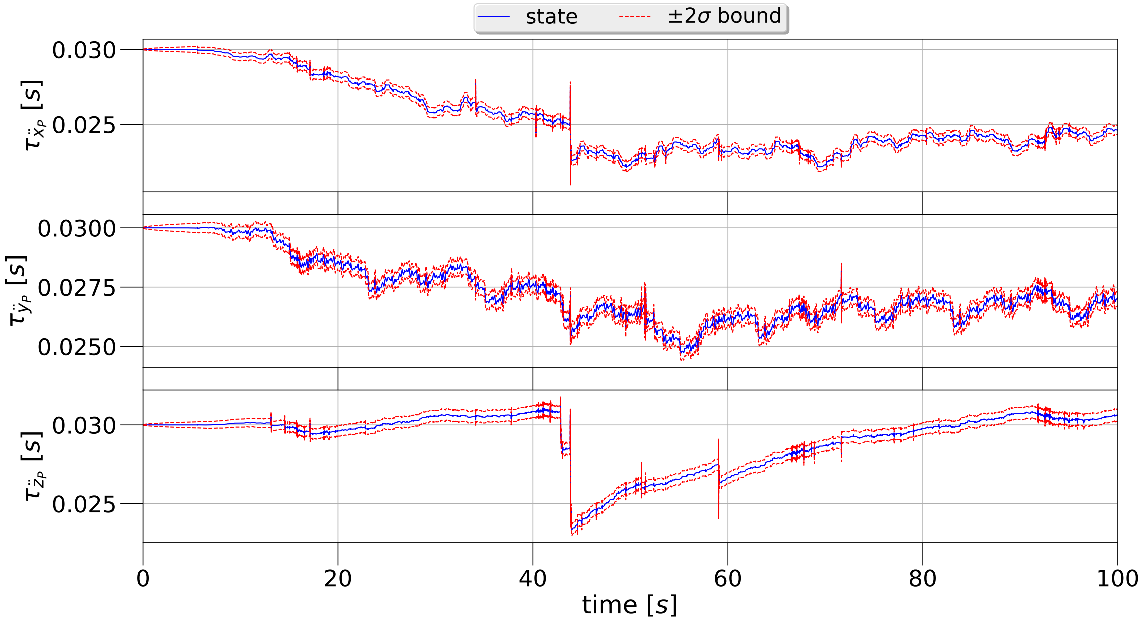


Figure H.22: SRS's translation lag in the inertial reference frame $\mathbb{E}_I$.

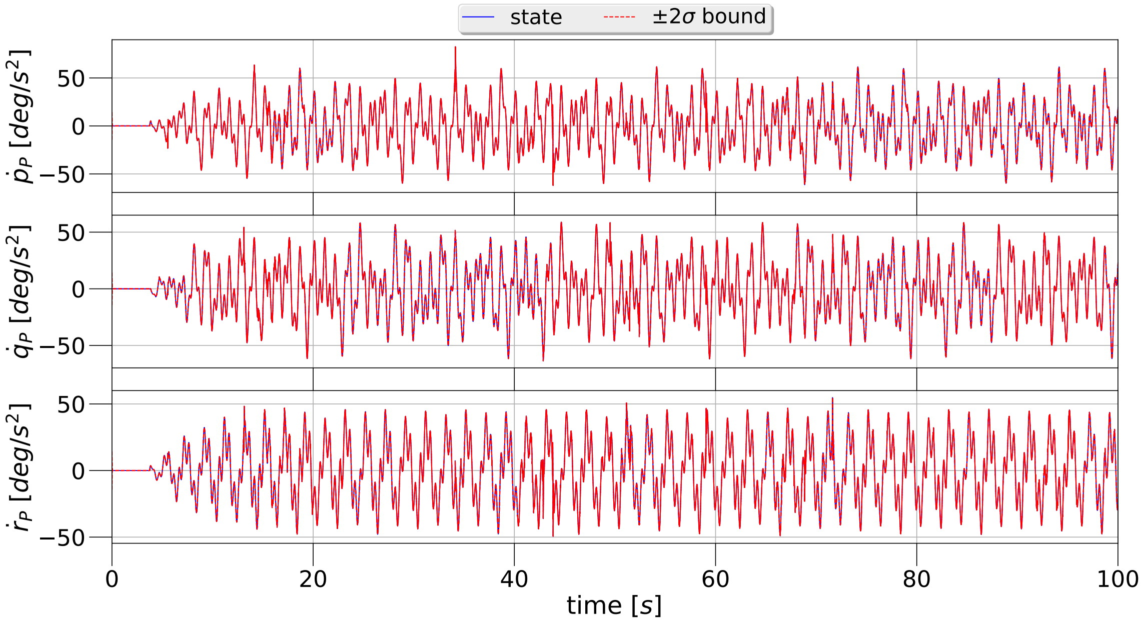


Figure H.23: SRS's angular acceleration in the upper moving reference frame $\mathbb{E}_P$.

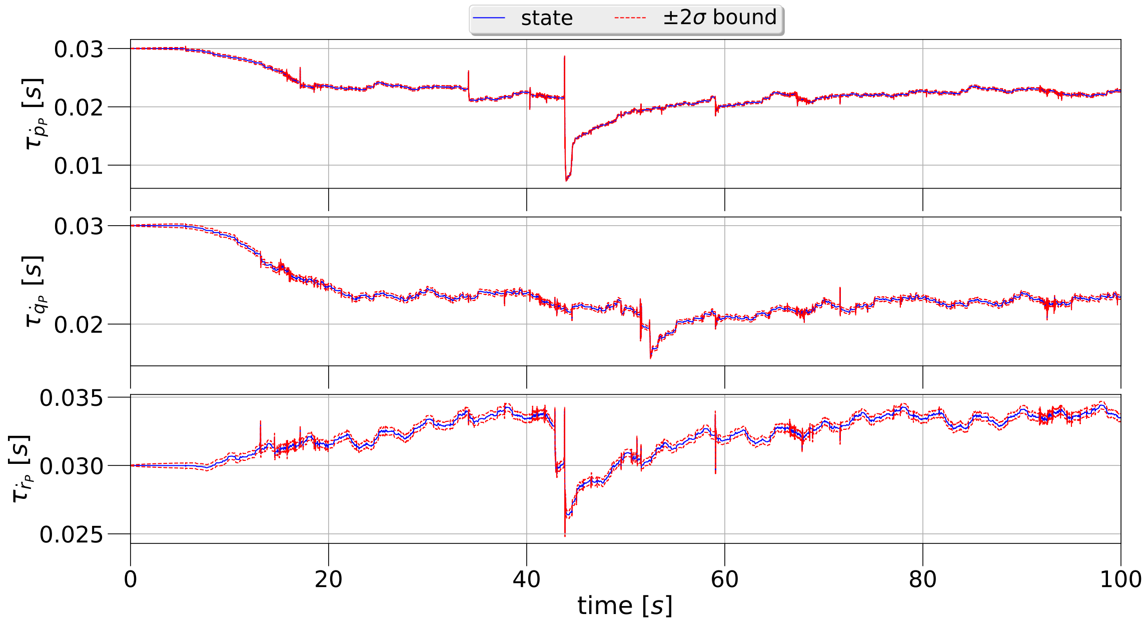


Figure H.24: SRS's rotational lag in the upper moving reference frame $\mathbb{E}_P$.

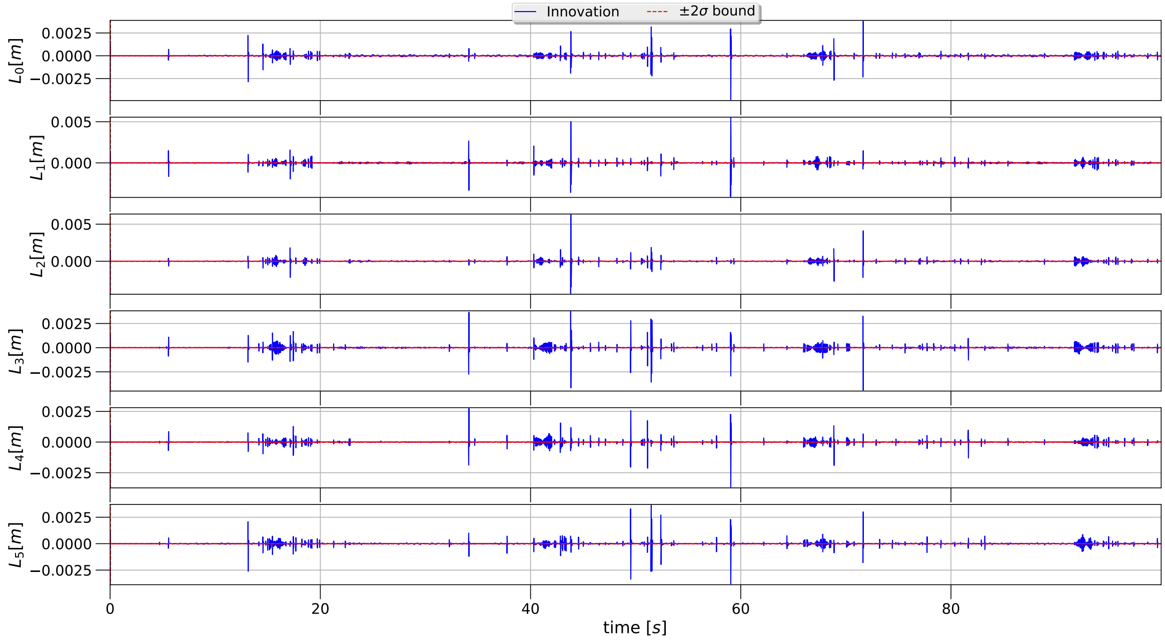


Figure H.25: SRS's linear encoders innovation.

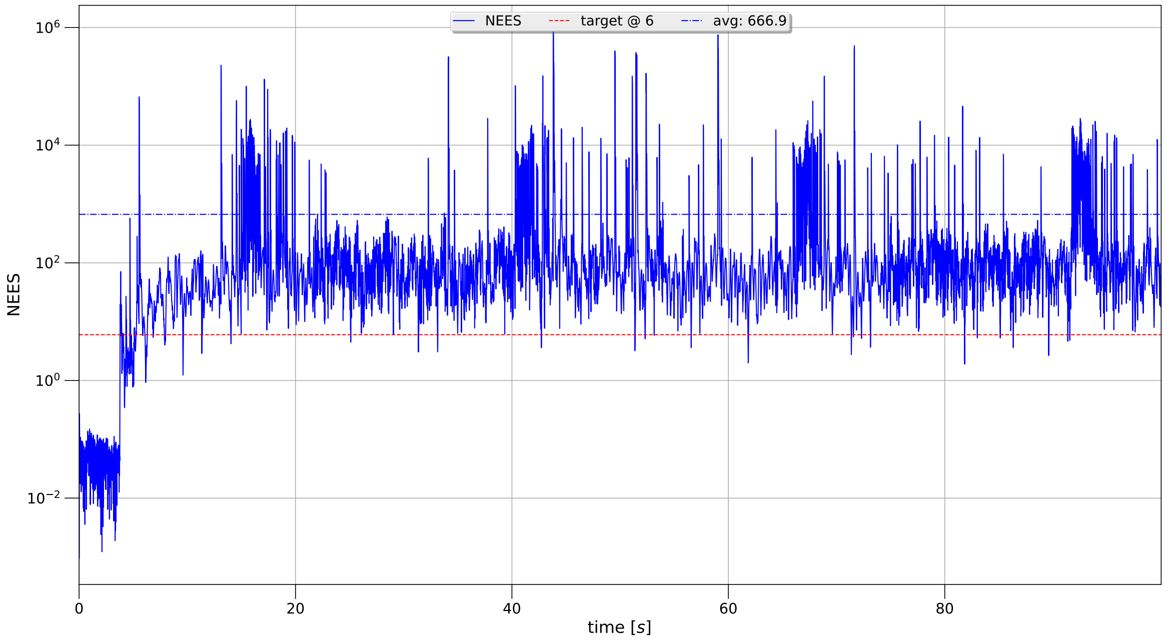


Figure H.26: SRS's linear encoders innovation .

H.3. SRS's Circular Translation around UGP

In comparison to the 'Complex' motion profile, the circular translation only has motion on certain axis. For Configuration 1 and 2 this is not an issue and makes the state estimation actually easier due to the decrease in overall motion. However, for Configuration 3, the lack of data causes observability issue on certain lags.

H.3.1. Configuration 3 --- HMD with IMU and SRS with Setpoints

SRS

Note in Figures H.28 and H.30 the sigma-bound of the lag on the z-axis or yaw increasing with time. This is due to lack of information on said lag with the current motion profile.

The lag in this motion profile is also lower then then in the 'complex' motion profile. This is probably due to the 'relaxed' motion profile chosen, making it easier for the SRS to follow its commands, resulting in a lower overall time-lag.

The timing issue, discussed in Appendix G, was worse for this dataset. Due to this reason, the ν values shown in Figure H.32 are larger than in Figure H.26.

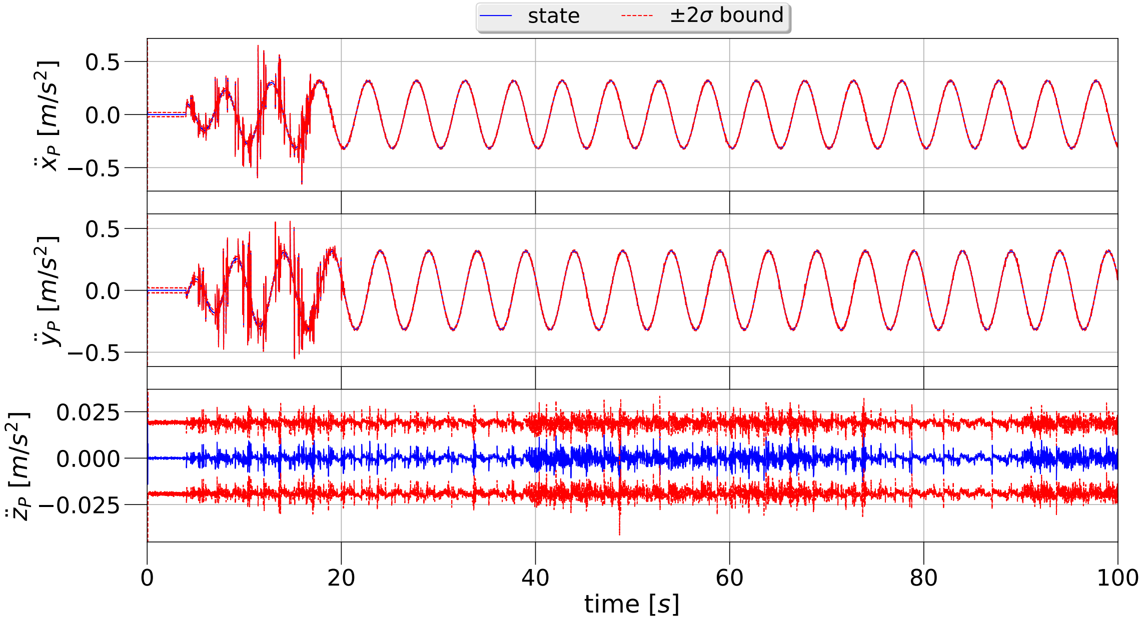


Figure H.27: SRS's acceleration in the inertial reference frame $\mathbb{E}_I$.

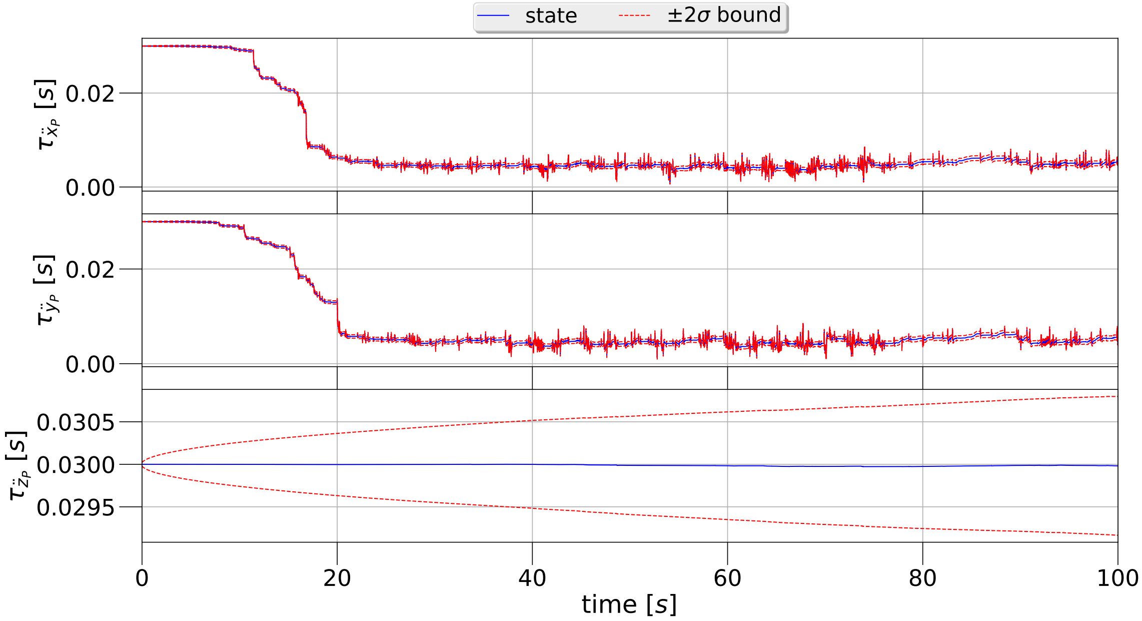


Figure H.28: SRS's translation lag in the inertial reference frame $\mathbb{E}_I$.

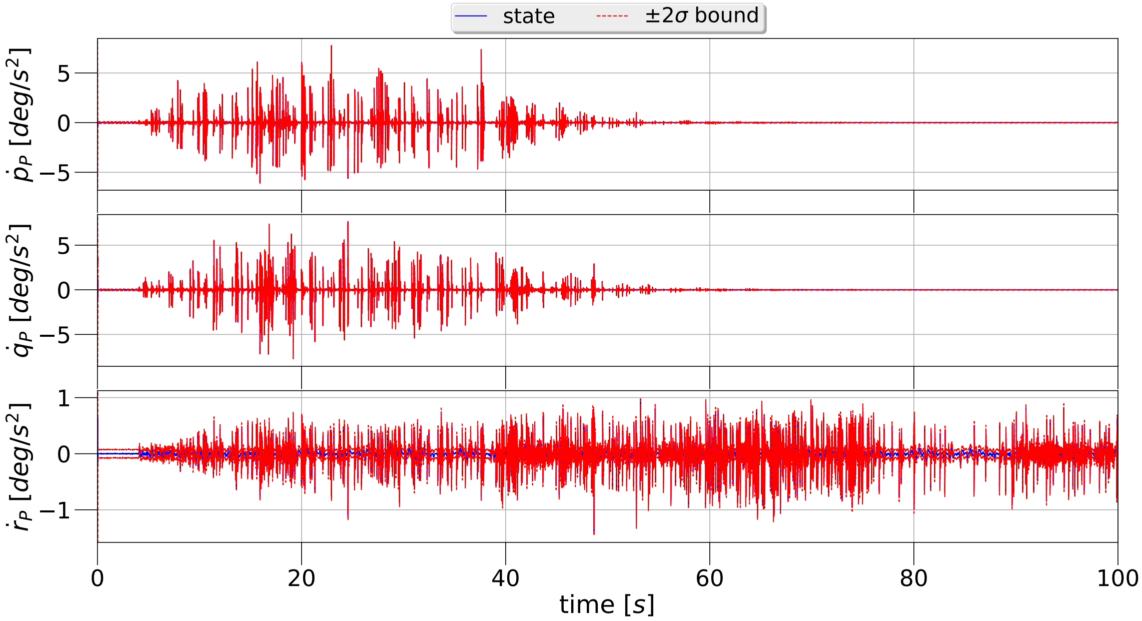


Figure H.29: SRS's angular acceleration in the upper moving reference frame $\mathbb{E}_P$.

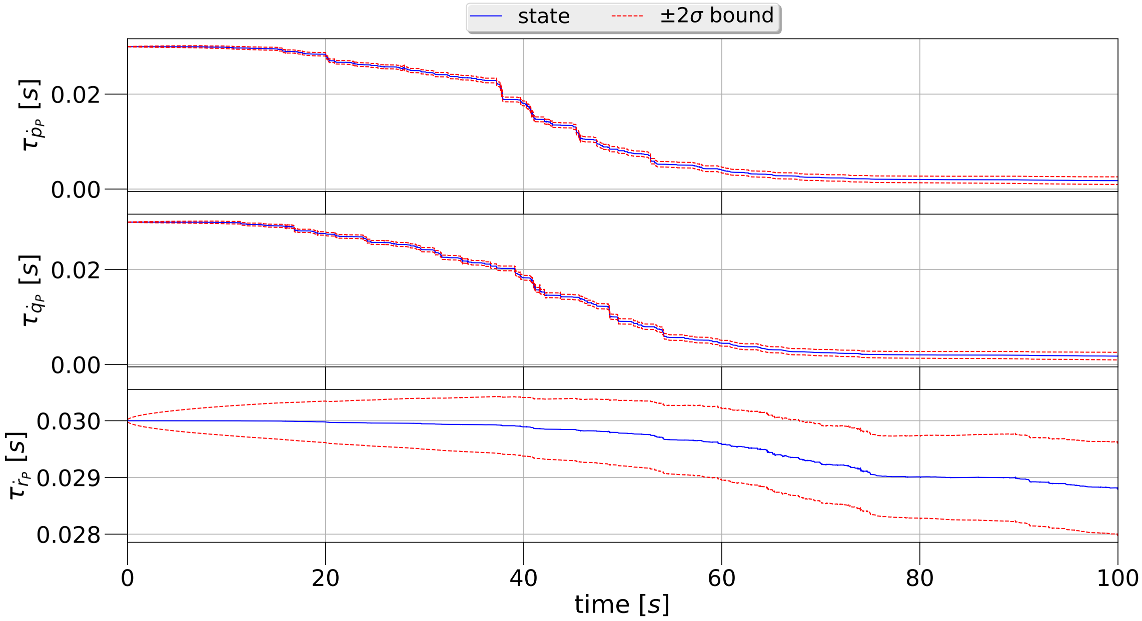


Figure H.30: SRS's rotational lag in the upper moving reference frame $\mathbb{E}_P$.

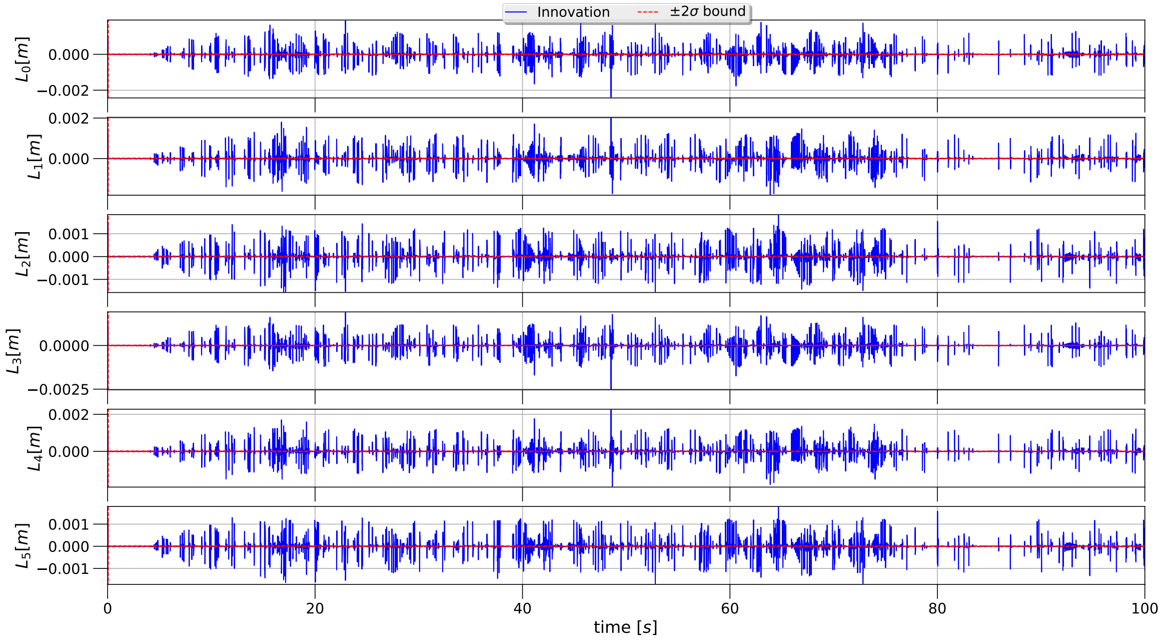


Figure H.31: SRS's linear encoders innovation.

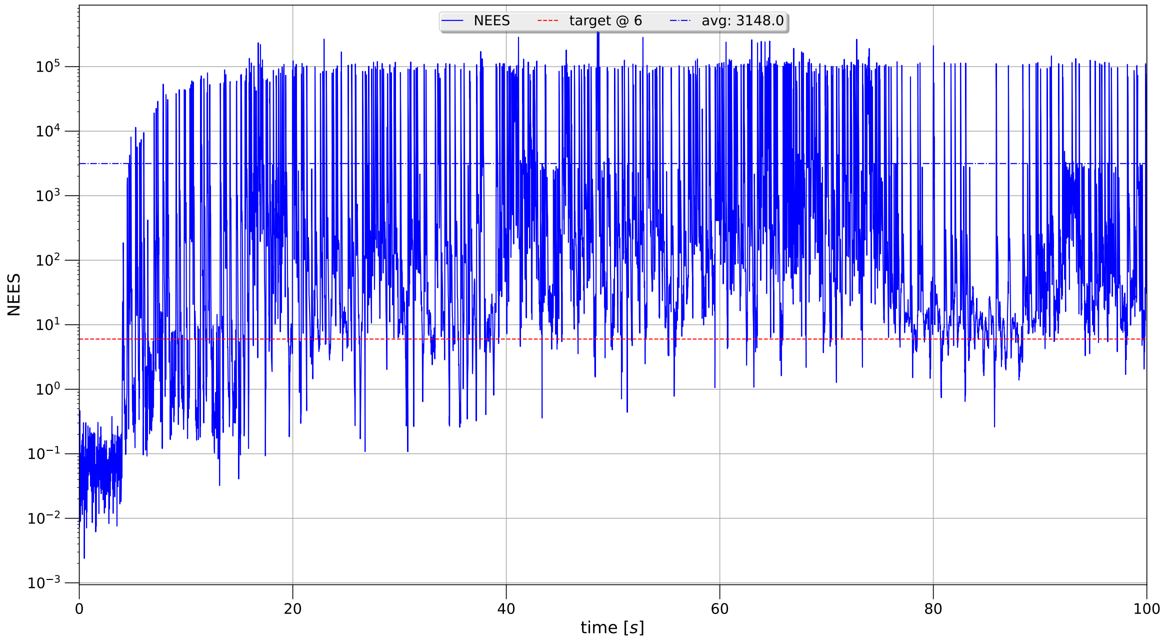


Figure H.32: SRS's linear encoders innovation .

H.4. No Movement

Due to the fact that there is absolutely no movement, Configuration 3 gains no knowledge on its lags. This is evident from Figures H.34 and H.36.

H.4.1. Configuration 3 --- HMD with IMU and SRS with Setpoints

SRS

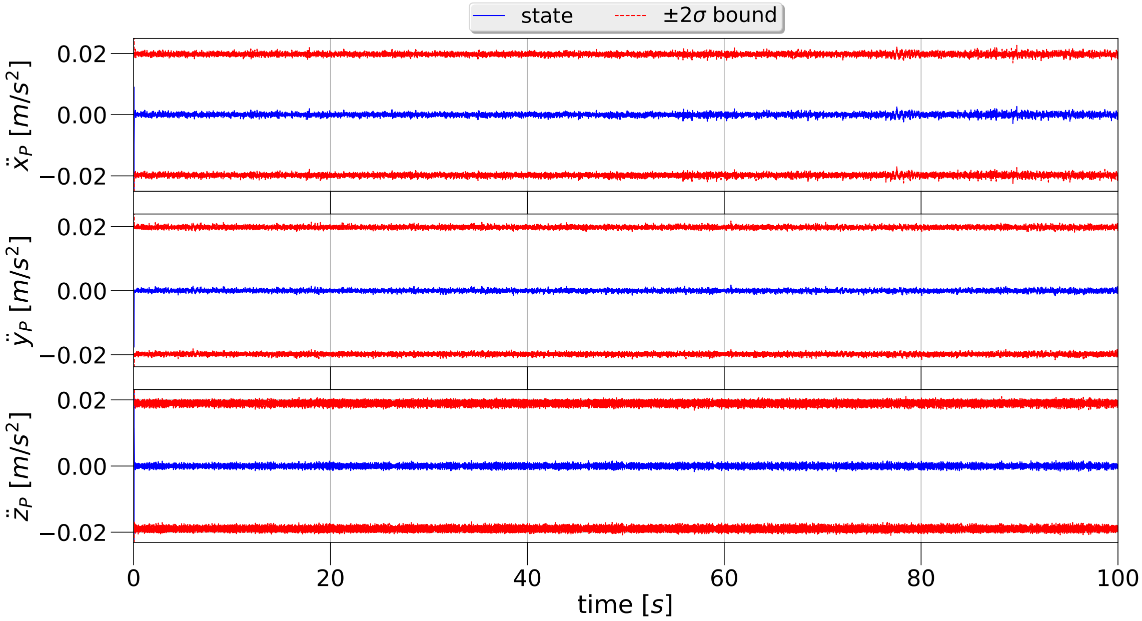


Figure H.33: SRS's acceleration in the inertial reference frame $\mathbb{E}_I$.

This motion profile does not strain the process model in any way as is evident from the innovations plotted in Figures H.37 and H.38.

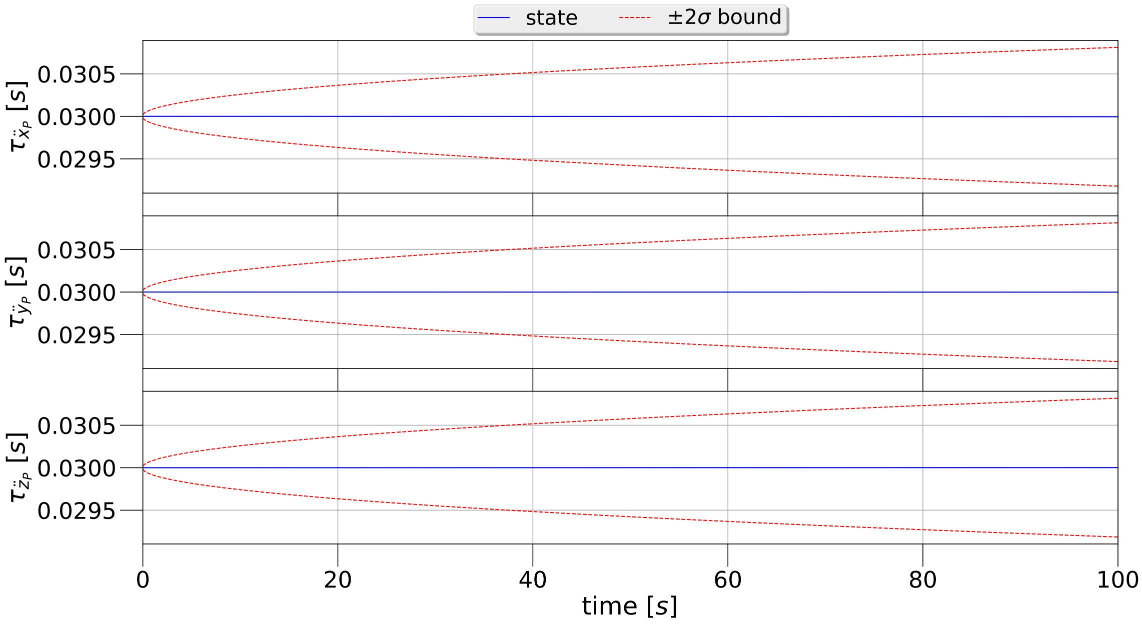


Figure H.34: SRS's translation lag in the inertial reference frame $\mathbb{E}_I$.

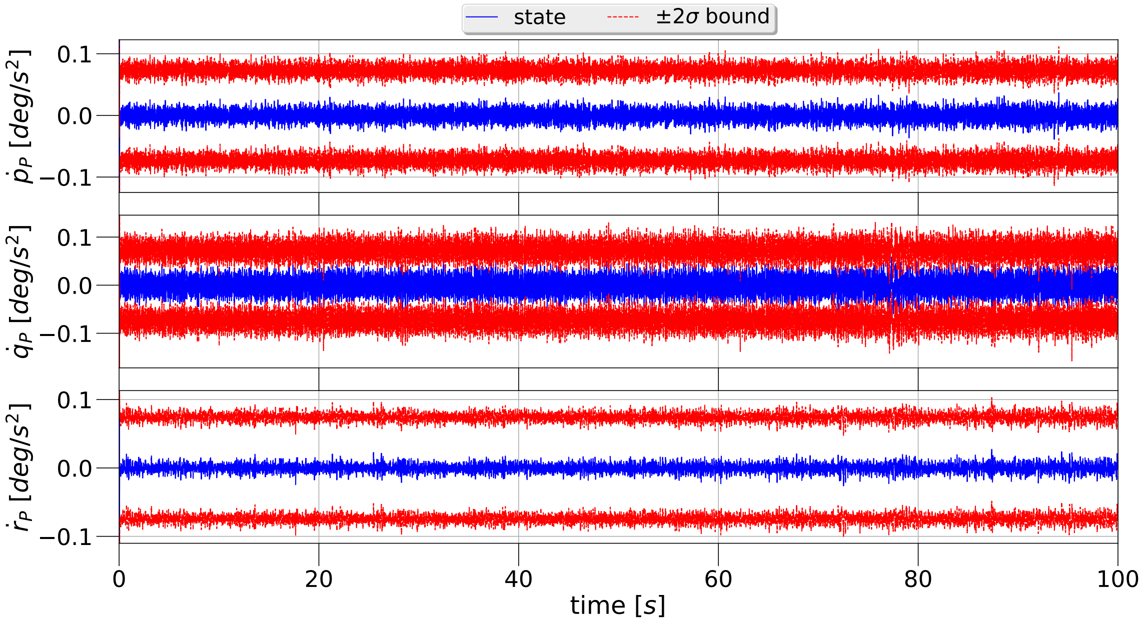


Figure H.35: SRS's angular acceleration in the upper moving reference frame $\mathbb{E}_P$.

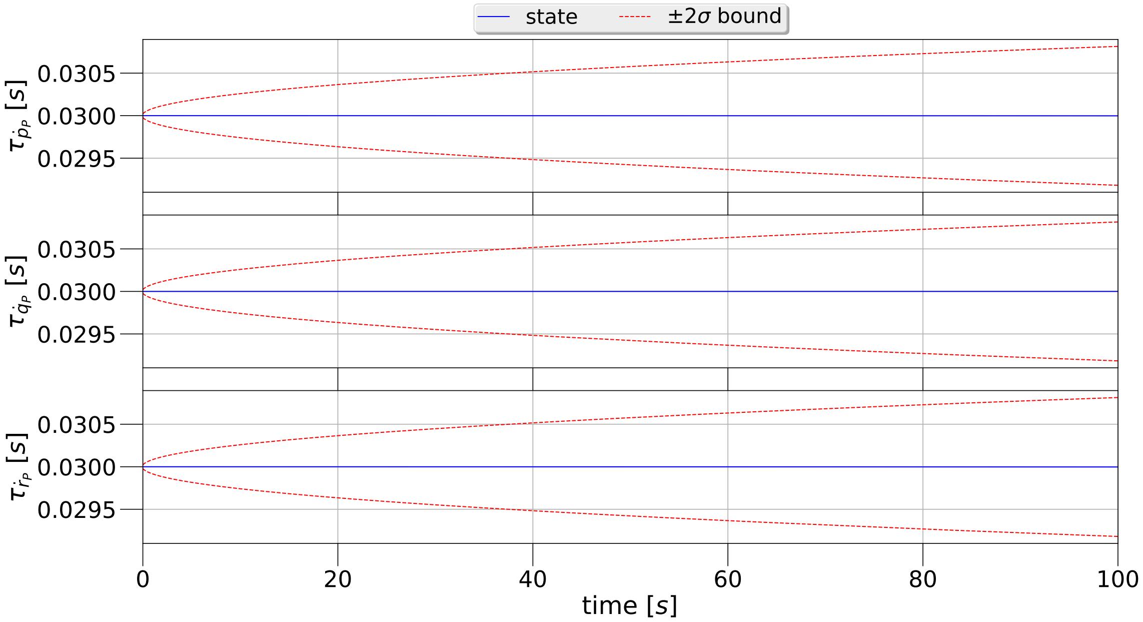


Figure H.36: SRS's rotational lag in the upper moving reference frame $\mathbb{E}_P$.

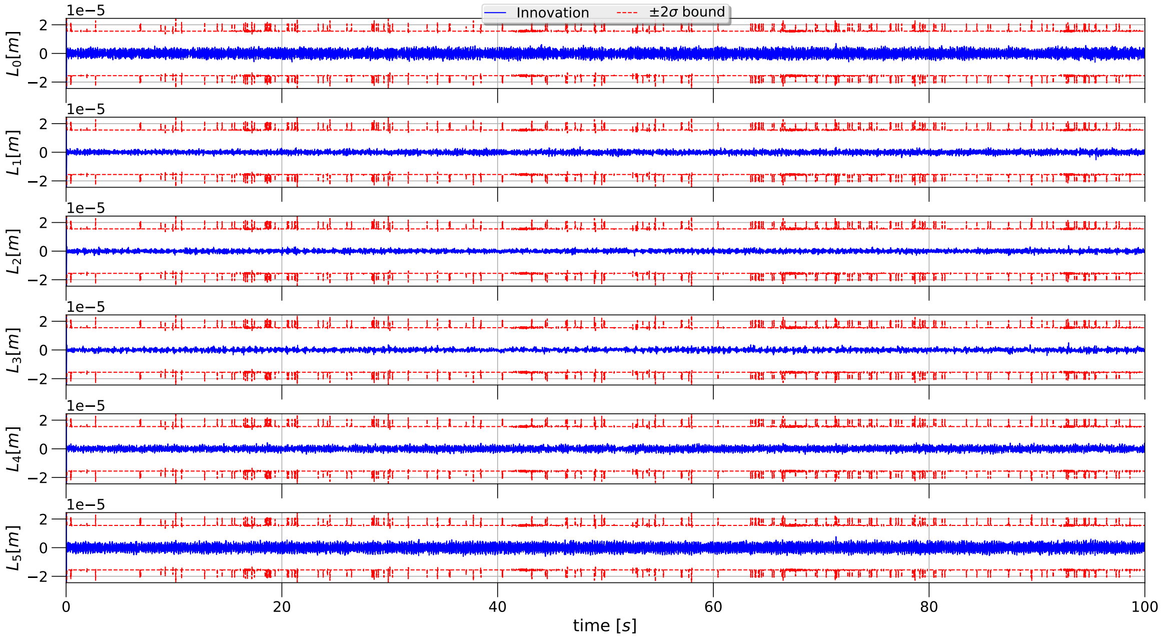


Figure H.37: SRS's linear encoders innovation.

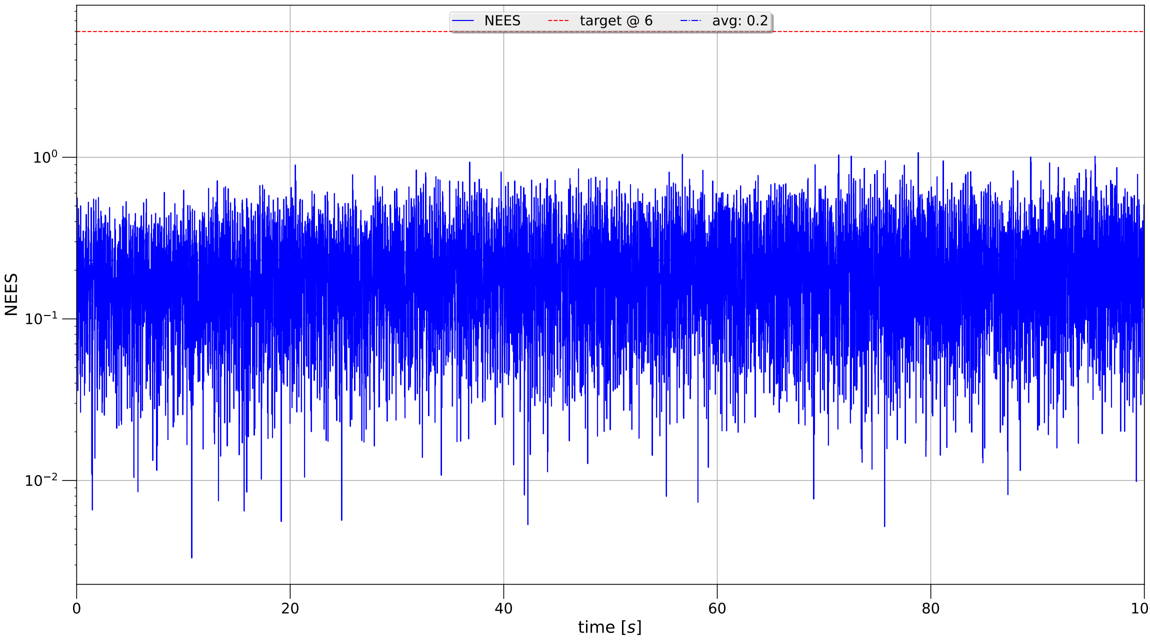


Figure H.38: SRS's linear encoders innovation .