Improved Stewart platform state estimation using inertial and actuator position measurements

MiletoviC, I.; Pool, D. M.; Stroosma, O.; van Paassen, M. M.; Chu, Q. P.

DOI
10.1016/j.conengprac.2017.03.006

Publication date
2017

Document Version
Accepted author manuscript

Published in
Control Engineering Practice

Improved Stewart Platform State Estimation using Inertial and Actuator Position Measurements

I. Miletović[a,*], D.M. Pool[a], O. Stroosma[a], M.M. van Paassen[a], Q.P. Chu[a]

[a] *Delft University of Technology, Kluyverweg 1, 2629HS Delft, The Netherlands*

Abstract

Accurate and reliable estimation of the kinematic state of a six degrees-of-freedom Stewart platform is a problem of interest in various engineering disciplines. Particularly so in the area of flight simulation, where the Stewart platform is in widespread use for the generation of motion similar to that experienced in actual flight. Accurate measurements of Stewart platform kinematic states are crucial for the application of advanced motion control algorithms and are highly valued in quantitative assessments of simulator motion fidelity. In the current work, a novel method for the reconstruction of the kinematic state of a Stewart platform is proposed. This method relies on an Unscented Kalman Filter (UKF) for a tight coupling of on-platform inertial sensors with measurements of the six actuator positions. The proposed algorithm is shown to be superior to conventional iterative methods in two main areas. First, more accurate estimates of motion platform velocity are obtained and, second, the algorithm is robust to inherent measurement uncertainties like noise and bias. The results were validated on the SIMONA Research Simulator (SRS) at TU Delft. To this end, an efficient implementation of the algorithm was driven, in real time, by actual sensor measurements from two representative motion profiles.

Keywords: Sensor integration, State estimation, Unscented Kalman Filter, Parallel robotics, Stewart platform

1. Introduction

The Stewart platform [1] (see Figure 1) is a six degrees-of-freedom (DOF) parallel manipulator that is in widespread use throughout various robotic discliplines. The main reasons for this are its relatively low mass, high accuracy, rigidity and support for heavier loads as compared to conventional serial manipulators. Some common applications include manufacturing [2, 3], surgery [4, 5] and medical rehabilitation [6, 7]. Perhaps the most popular application is that of vehicular motion simulation, e.g., aeronautical flight simulation [8, 9].

*Corresponding author

Email addresses: I.Miletovic@tudelft.nl (I. Miletović), D.M.Pool@tudelft.nl (D.M. Pool), O.Stroosma@tudelft.nl (O. Stroosma), M.M.vanPaassen@tudelft.nl (M.M. van Paassen), Q.P.Chu@tudelft.nl (Q.P. Chu)

Given this wide range of applications, obtaining an accurate estimate of a Stewart platform's kinematic state is often of interest. A typical example is the application of advanced control techniques to Stewart platforms, e.g., [10, 11, 12, 13]. An increasingly relevant application is also that of flight simulator motion fidelity assessment. It is well known that humans perceive inertial self-motion predominantly by means of the vestibular system, which is sensitive to both specific force and angular acceleration [14]. Current efforts to quantify motion fidelity therefore focus on the development and standardization of frequency-domain system identification methods to capture and specify the motion transfer characteristics of contemporary motion cueing systems [15, 16, 17, 18]. Estimation of the kinematic state of a Stewart platform, however, is inhibited by two particular difficulties. The first is the inference of a Stewart platform's position and attitude from measurements of its six actuator positions (the so-called *forward kinematics* problem). The second is the subsequent inference of platform (angular) velocity and acceleration, as required by many of the typical applications listed here.

The forward kinematics of the Stewart platform have been a subject of study for many decades [19]. Over the years, several authors have shown, for numerous possible platform geometries, that the forward kinematics of a Stewart platform has up to 40 possible solutions [20, 21, 22, 23, 24]. As already acknowledged by Merlet in [25, 26], however, a closed-form algebraic method to obtain a single *unique* solution for the platform position and attitude based on only six actuator length measurements does not exist. As a result, a variety of numerical methods to solve the forward kinematics have been developed [27, 28, 29, 26]. A common limitation of these efforts to solve the forward kinematics of a Stewart platform, however, is that they are limited to inference of platform position and attitude *only*. Neither addresses the second problem, namely that of inference of platform (angular) velocity and acceleration. While numerical differentiation of position and attitude is applied to obtain the latter, the presence of measurement noise in physical sensors clearly renders such an approach suboptimal. Nonlinear state observers have therefore also been relied upon to directly evaluate the platform pose and velocity [30, 31, 32]. These typically require an explicit model of both platform dynamics and, especially for hydraulically driven platforms, actuator dynamics. While these methods offer some robustness to *model* uncertainties, they lack an inherent mechanism to account for *sensor* uncertainties, e.g., noise and bias. In light of the growing availability of both affordable and accurate inertial sensors [33], this paper presents a novel approach to estimate the kinematic state of a Stewart platform.

The proposed approach relies on the fusion of on-platform inertial sensors, encapsulated in an Inertial Measurement Unit (IMU), with the six available actuator position sensors. This is accomplished by using an extension of the Kalman Filter (KF) [34] to nonlinear systems. Through this approach, the need for explicit knowledge of motion platform and actuator dynamics is eliminated, while at the same time incorporating robustness to measurement inaccuracies. The idea of using a Kalman filter-based method for such an application is not new and has been widely demonstrated in, e.g., robotics [35, 36, 37], aerospace [38, 39, 40,

41], biomedical engineering [42] and power plant control [43]. Applications to Stewart platforms, however, remain limited to only a small number of DOFs [44, 45]. A recent extension of a sensor fusion scheme to all six DOFs, in which the application of the Iterated Extended Kalman Filter (IEKF) [46] was investigated, did not yield satisfactory results [47]. The main difficulty in the application of contemporary filters like the Extended Kalman Filter (EKF) or IEKF lies in the highly nonlinear kinematics of the Stewart platform, which causes the inherent assumption of local linearity to break down. The current work is therefore focused on the simulation and real-time experimental validation of a method particularly suited for highly nonlinear systems, namely the Unscented Kalman Filter (UKF) [48, 49].

The paper is structured as follows. First, a brief introduction to the Stewart platform and its kinematics is provided. Then, the proposed state reconstruction algorithm is introduced, followed by a verification of the algorithm on the basis of computer simulations. Subsequently, the validation of the proposed algorithm on the SIMONA Research Simulator (SRS), using a real-time implementation driven by actual sensor measurements, is presented. This validation also includes a comparison to an iterative scheme commonly applied to estimate the kinematic state of Stewart platforms. Finally, a brief discussion is provided and the paper is concluded.

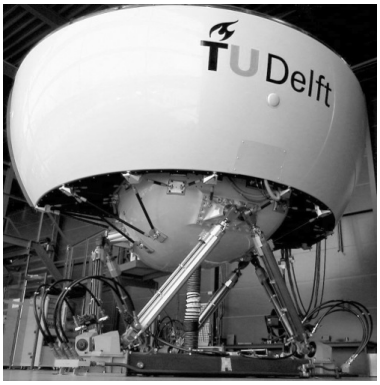
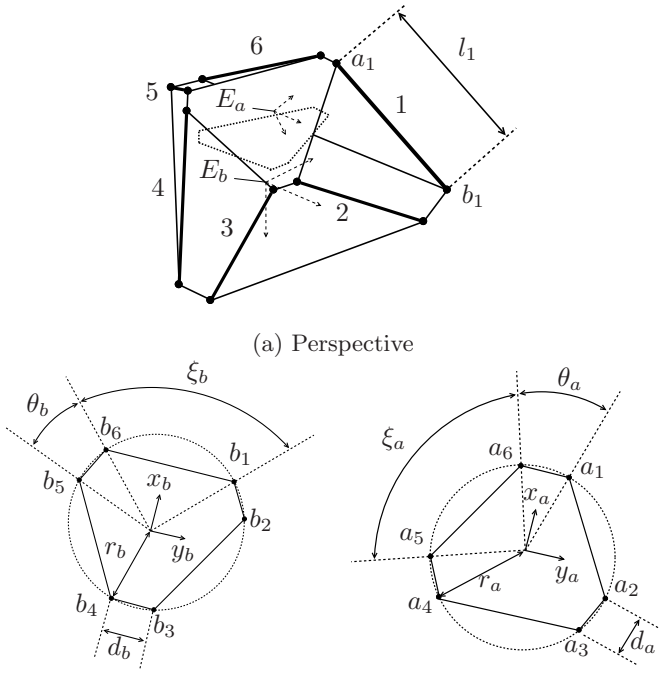


Figure 1: The SIMONA Research Simulator, using a Stewart platform as the motion providing mechanism.

2. Stewart platform kinematics

The kinematics of a Stewart platform are defined by its geometry, shown in Figure 2. The coordinates of the joints on the lower (Figure 2b) and upper (Figure 2c) platform, in turn, determine the geometry of the platform. The locations of these joints are conveniently specified with respect to the centroids of the joints on the lower and upper platform. These centroids are typically referred to as the Lower Gimbal Point (LGP) and Upper Gimbal Point (UGP), respectively. The origins of two right-handed reference frames E_a and E_b are attached to the UGP and LGP, respectively. All joints on the lower and upper platforms lie equidistantly spaced with a certain distance (i.e., d_a or d_b) in pairs on a circle with a given radius (i.e., r_a or r_b), such

that the coordinates of each joint (i.e., $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$) can be derived from basic trigonometry. Note that even though this specific configuration of the Stewart is used in the current work, the authors foresee no issues in applying the proposed methodology to any non-singular platform geometry.



(a) Perspective

(b) Lower platform

(c) Upper platform

Figure 2: The geometry of a Stewart platform.

The length of each of the connecting elements between the joints then follows from the geometry as:

$$l_i(\boldsymbol{c}, \boldsymbol{\Phi}) = \|\boldsymbol{c} + T_{ba}(\boldsymbol{\Phi})\, \boldsymbol{a}_i^a - \boldsymbol{b}_i^b\| \quad \forall \quad i \in [1, \ldots, 6] \tag{1}$$

The vector $\boldsymbol{c}$ defines the position of the origin of reference frame E_a with respect to E_b and is expressed in Cartesian coordinates as:

$$\boldsymbol{c} = \begin{bmatrix} x & y & z \end{bmatrix}^\intercal \tag{2}$$

T_{ba} is the transformation matrix that describes the transformation $a_i^a \rightarrow a_i^b$ and therefore Equation (1) depends on the attitude of frame E_a with respect to E_b. Here, attitude is represented using the Euler-Rodrigues quaternion formulation [50, 51] for numerical efficiency. As such, the attitude vector is:

$$\boldsymbol{\Phi} = \begin{bmatrix} e_0 & e_x & e_y & e_z \end{bmatrix}^\intercal \tag{3}$$

and the transformation matrices T_{ab} and T_{ba} can subsequently be defined as [50]:

$$T_{ab} = T_{ba}^\intercal = \begin{bmatrix} e_x^2 + e_0^2 - e_y^2 - e_z^2 & 2\,(e_x e_y + e_z e_0) & 2\,(e_x e_z - e_y e_0) \\ 2\,(e_x e_y - e_z e_0) & e_y^2 + e_0^2 - e_x^2 - e_z^2 & 2\,(e_y e_z + e_x e_0) \\ 2\,(e_x e_z + e_y e_0) & 2\,(e_y e_z - e_x e_0) & e_z^2 + e_0^2 - e_x^2 - e_y^2 \end{bmatrix} \tag{4}$$

Equation (1) specifies the so-called *inverse* kinematics of the Stewart platform. Inference of the platform pose from knowledge of the length of each actuator is known as the *forward* kinematics of the motion platform. While the inverse kinematics are trivial to compute, the forward kinematics cannot be expressed in a closed algebraic form [25, 26]. To compute the forward kinematics of the platform iterative schemes such as the Newton-Raphson (NR) or Gauss-Newton (GN) method [27, 10] are therefore commonly applied. Such methods iteratively compute the position and attitude of the motion platform, denoted collectively as the platform pose vector $\boldsymbol{p}$, from the measured actuator position vector $\boldsymbol{l}_m$:

$$\boldsymbol{p}_{j+1} = \boldsymbol{p}_j + J_{lp}^{-1}(\boldsymbol{p}_j)\,(\boldsymbol{l}_m - \boldsymbol{l}_j) \tag{5}$$

until satisfactory convergence is reached (i.e., $\|\boldsymbol{p}_{j+1} - \boldsymbol{p}_j\| \leq e_{gn}$) or until a fixed number of iterations (i.e., $j \leq N_{gn}$) have been performed. J_{lp} is the Jacobian of the actuator position vector $\boldsymbol{l}$ with respect to the platform pose vector $\boldsymbol{p}$.

These iterative numerical methods are limited in that they lack inherent robustness to measurement inaccuracies in the actuator position sensors and, moreover, provide information on platform pose only. As knowledge of velocity and acceleration is also of interest, a novel approach based on a tightly-coupled fusion of on-platform inertial sensors with actuator position sensors is discussed and evaluated in the remainder of this paper.

3. Stewart platform state reconstruction using sensor fusion

In [45], a novel sensor fusion scheme was proposed and evaluated for the reconstruction of the partial (three DOF) kinematic state of a Stewart platform. This method relies on a tightly-coupled fusion of on-platform inertial sensors and actuator position sensors. Extensions of the well-known KF [34] to nonlinear systems, namely the EKF and IEKF [46], were evaluated. The EKF uses direct linearization of the nonlinear system in order to propagate the stochastic properties of the estimated system state. The IEKF, in turn, iteratively corrects the estimated system state based on a consecutively re-linearized observation model until satisfactory convergence of the estimated system state is attained. Systems which are characterized by a highly nonlinear observation model typically benefit from the application of the IEKF. In case of the Stewart platform, it was demonstrated that, even for the reconstruction of the partial kinematic state, the IEKF was required to attain convergence of the filter to the true system state.

In [47], the IEKF-based sensor fusion scheme from [45] was extended to all six DOF of the Stewart platform. Due to the significant increase in nonlinearity of the stochastic system model, however, the IEKF generally proved unable to attain satisfactory convergence. Consequently, the UKF [48, 49], which promises improved filter performance for highly nonlinear systems, is applied in the current work. This section will first introduce the stochastic system model of the Stewart platform, after which the working principle of the KF and, specifically, the UKF is briefly recapitulated.

3.1. Stochastic system model of the Stewart platform

The basis of a typical sensor fusion scheme is formed by a stochastic system model that relates the various available physical sensor measurements to the parameters that determine the state of the dynamic system under consideration. Such a model has the following general structure in the nonlinear case [46]:

$$\begin{aligned} \dot{\underline{\boldsymbol{x}}}(t) &= f(\underline{\boldsymbol{x}}(t), \boldsymbol{u}(t), t) + G(\underline{\boldsymbol{x}}(t), t)\underline{\boldsymbol{w}}(t) \\ \underline{\boldsymbol{y}}(t) &= h(\underline{\boldsymbol{x}}(t), \boldsymbol{u}(t), t) + \underline{\boldsymbol{v}}(t) \end{aligned} \tag{6}$$

Here, $\underline{\boldsymbol{x}}$ is the stochastic system state vector, $\boldsymbol{u}$ is the input vector and $\underline{\boldsymbol{y}}$ is the output or observation vector. The quantities $\underline{\boldsymbol{w}}$ and $\underline{\boldsymbol{v}}$ are the process and observation noise vectors, respectively. These are commonly assumed to be white and Gaussian noise processes:

$$\begin{aligned} \underline{\boldsymbol{v}}(t) \sim \mathfrak{N}(0, R(t)) \quad \text{and} \quad E\{\underline{\boldsymbol{v}}(t)\underline{\boldsymbol{v}}^{\mathsf{T}}(t+\tau)\} = R(t)\delta(t) \\ \underline{\boldsymbol{w}}(t) \sim \mathfrak{N}(0, Q(t)) \quad \text{and} \quad E\{\underline{\boldsymbol{w}}(t)\underline{\boldsymbol{w}}^{\mathsf{T}}(t+\tau)\} = Q(t)\delta(t) \end{aligned} \tag{7}$$

The vector function $f(\cdot)$ and matrix $G(\cdot)$ describe the evolution of the stochastic process state in time as a function of the current value of the state, input and process noise vector, respectively. The equation in which these terms appear is therefore referred to as the prediction model. Similarly, the vector function $h(\cdot)$ relates the value of the stochastic process state to the available observation vector and the equation it appears in is therefore known as the observation model.

In case of the sensor fusion scheme proposed in [45, 47] for the reconstruction of the kinematic state of a Stewart platform, the input vector that drives the prediction of the system state consists of measurements from an IMU. IMUs typically contain three accelerometers and three rate gyros:

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{f}^{\mathsf{T}} & \boldsymbol{\omega}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \tag{8}$$

where $\boldsymbol{f}$ and $\boldsymbol{\omega}$ are the specific force and angular rate vector, respectively, defined in the right-handed reference frame E_a shown in Figure 2:

$$\boldsymbol{f} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^{\mathsf{T}} \quad \text{and} \quad \boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^{\mathsf{T}} \tag{9}$$

Based on this definition of the input vector, the stochastic system state is defined as:

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \underline{\boldsymbol{c}}^{\mathsf{T}} & \underline{\boldsymbol{V}}^{\mathsf{T}} & \underline{\boldsymbol{\Phi}}^{\mathsf{T}} & \underline{\boldsymbol{\lambda}}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \tag{10}$$

where $\boldsymbol{V}$ is the inertial velocity of the reference point attached to the upper platform expressed in reference frame E_a:

$$\boldsymbol{V} = \begin{bmatrix} u & v & w \end{bmatrix}^{\mathsf{T}} \tag{11}$$

and $\boldsymbol{\lambda}$ is a vector that contains the six constant biases typically assumed inherent in measurements of the inertial sensors (see, e.g., [38, 39]):

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_f^{\mathsf{T}} & \boldsymbol{\lambda}_\omega^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \lambda_x & \lambda_y & \lambda_z & \lambda_p & \lambda_q & \lambda_r \end{bmatrix}^{\mathsf{T}} \tag{12}$$

Note that the inclusion of the inertial sensor biases as part of the state vector allows for their estimation in conjunction with the kinematic state of the platform. The observation vector is defined to contain the measured lengths of each of the six actuators:

$$\boldsymbol{y} = \begin{bmatrix} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{bmatrix}^\intercal \tag{13}$$

Given these definitions in combination with knowledge on the geometry of the Stewart platform as shown in Figure 2, it is possible to formulate expressions for the vector functions $f(\cdot)$ and $h(\cdot)$ as well as the matrix $G(\cdot)$ appearing in Equation (6). From basic kinematics, it follows:

$$f(\underline{\boldsymbol{x}}, \boldsymbol{u}) = \begin{bmatrix} \dot{\underline{\boldsymbol{c}}} \\ \dot{\underline{\boldsymbol{V}}} \\ \dot{\underline{\boldsymbol{\Phi}}} \\ \dot{\underline{\boldsymbol{\lambda}}} \end{bmatrix} = \begin{bmatrix} T_{ba}(\underline{\boldsymbol{\Phi}})\,\underline{\boldsymbol{V}} \\ (\boldsymbol{f} - \underline{\boldsymbol{\lambda}}_f) + T_{ab}(\underline{\boldsymbol{\Phi}})\,\boldsymbol{g} - (\underline{\Omega}_\times - \underline{\Lambda})\underline{\boldsymbol{V}} \\ \underline{\Omega}_q(\boldsymbol{\omega} - \underline{\boldsymbol{\lambda}}_\omega) \\ \boldsymbol{0} \end{bmatrix} \tag{14}$$

where:

$$\Omega_\times = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}, \; \Lambda = \begin{bmatrix} 0 & -\lambda_r & \lambda_q \\ \lambda_r & 0 & -\lambda_p \\ -\lambda_q & \lambda_p & 0 \end{bmatrix} \tag{15}$$

and where T_{ab} and T_{ba} are transformation matrices. Furthermore, $\boldsymbol{g}$ is the gravitational acceleration and Ω_q is defined as [50]:

$$\Omega_q = \frac{1}{2} \begin{bmatrix} -e_x & -e_y & -e_z \\ e_0 & -e_z & e_y \\ e_z & e_0 & -e_x \\ -e_y & e_x & e_0 \end{bmatrix} \tag{16}$$

Similarly, $G(\cdot)$ can be derived as:

$$G(\underline{\boldsymbol{x}}) = \begin{bmatrix} 0_{3\times 6} \\ -I_3 \quad \underline{\Omega}_v \\ 0_{4\times 3} \quad -\underline{\Omega}_q \\ 0_{6\times 6} \end{bmatrix}, \text{ with } \Omega_v = \begin{bmatrix} 0 & w & -v \\ -w & 0 & u \\ v & -u & 0 \end{bmatrix} \tag{17}$$

where I_3 is the three-dimensional identity matrix and Ω_q is given by Equation (16). Finally, the observation model contains the inverse kinematics of the motion platform:

$$h(\underline{\boldsymbol{x}}) = \begin{bmatrix} l_1(\underline{\boldsymbol{x}}) & l_2(\underline{\boldsymbol{x}}) & l_3(\underline{\boldsymbol{x}}) & l_4(\underline{\boldsymbol{x}}) & l_5(\underline{\boldsymbol{x}}) & l_6(\underline{\boldsymbol{x}}) \end{bmatrix}^\intercal \tag{18}$$

where $l_i(\boldsymbol{x})$ follows from the geometry depicted in Figure 2 as given by Equation (1). Notice that, in contrast to methods that rely on nonlinear state observers [30, 31, 32], the kinematic system model presented here requires no knowledge of motion platform or actuator dynamics.

3.2. The Unscented Kalman Filter

The stochastic system model defined by Equation (6) can be used in combination with a nonlinear recursive filtering algorithm to reconstruct the stochastic system state in real-time. Extensions of the well-known Kalman Filter (KF) [34, 46] to nonlinear systems are the most widely applied class of filtering algorithms used for this purpose. The KF is a minimum-variance estimator that minimizes the posterior variance of the stochastic process state at any discrete time t_k:

$$\min_{\underline{\hat{\boldsymbol{x}}}_k^{(+)}} \left[\text{trace} \left(P_k^{(+)} \right) \right] \tag{19}$$

where the superscript $(+)$ denotes posterior quantities and where P_k is the covariance of the stochastic process state at time t_k. This optimization constitutes two distinct phases, as illustrated in Figure 3. In the first phase, the *prediction*, the stochastic system state and its covariance are propagated in time using the system input $\boldsymbol{u}_k$ and the prediction model included within the stochastic system model given by Equation (6). The second phase, the *correction*, uses the most recent observation vector $\boldsymbol{y}_k$ in combination with the observation model to correct the predicted quantities. This is done by means of the so-called Kalman gain matrix K_k. For the computation of this matrix, the statistical properties of the process and observation noises are explicitly taken into account through the noise covariance matrices Q_k and R_k.

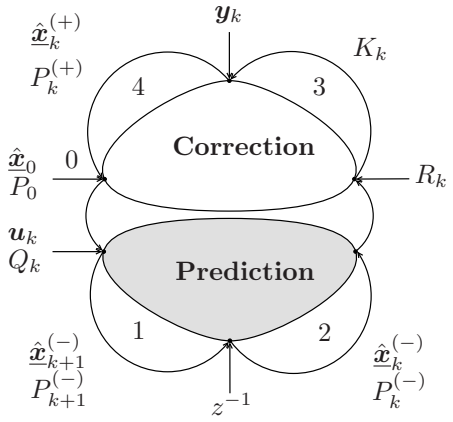


Figure 3: Schematic of the Kalman filter [34], illustrating the distinctive *prediction* and *correction* phases.

In implementations of the KF for nonlinear systems, the propagation of these statistical properties becomes problematic and typically results in a suboptimal filter [48]. The fundamental reason for this is that a Gaussian process which undergoes a nonlinear transformation is, in general, no longer Gaussian [52]. Consequently, knowledge of the mean and covariance as propagated by the KF is insufficient to fully capture the probability distribution of the stochastic process. Nonetheless, the KF can be applied to nonlinear systems by approximating the posterior mean and covariance of the stochastic system state. Such approximations are typically still classified as optimal filters from a *practical* point of view. A common

approach is to use local linearization on the stochastic system model given by Equation (6), which results in the well-known Extended Kalman Filter (EKF) and Iterated Extended Kalman Filter (IEKF) [46]. As mentioned previously, however, the application of these filters to the reconstruction of the complete kinematic state of the Stewart platform did not yield satisfactory results [47].

A better alternative to direct linearization of the stochastic system model is provided by the Unscented Kalman Filter (UKF) [48, 49]. The UKF approximates the nonlinear transformations of the stochastic system state and its covariance using a technique known as the Scaled Unscented Transformation (SUT) [53]. Given a general nonlinear function $g(\cdot)$, this technique entails the transformation of a deterministically calculated and symmetric set of samples $\boldsymbol{\mathcal{X}}$ around the mean of a general random variable $\underline{\boldsymbol{x}}$, as shown in Figure 4, to obtain the transformed set of samples $\boldsymbol{\mathcal{Y}}$. The posterior mean and covariance, $\underline{\bar{\boldsymbol{y}}}$ and $P_{\underline{yy}}$, can subsequently be approximated as illustrated in the figure. In contrast to the EKF or IEKF, the UKF therefore does not *directly* approximate the nonlinear transformation $g(\cdot)$, but approximates the probability distribution of $\underline{\boldsymbol{y}}$ by means of the computed values for $\underline{\bar{\boldsymbol{y}}}$ and $P_{\underline{yy}}$. This method is also referred to as *statistical linearization* and its merits for nonlinear state reconstruction problems have been demonstrated in various applications over the years, e.g., [43, 36, 42, 41].

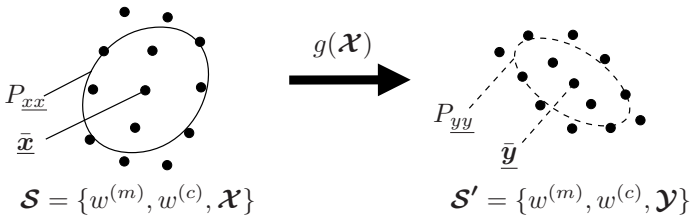


Figure 4: Illustration of the SUT inspired by [49, 54].

As outlined in [49, 53, 54], the accuracy and numerical stability of the UKF are mainly influenced by a total of three scalar parameters: κ, α and β. These parameters affect the scalar set of weights $w^{(m)}$ and $w^{(c)}$. Together with these set of weights, the collections $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ form the so-called sigma-points $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{S}}'$ (see Figure 4), respectively, used to approximate the probability distributions of the state and output. The parameters κ, α and β determine the distribution of sigma-points and can also be used to incorporate skewness or kurtosis of the probability distribution. In [55, 56], several guidelines can be found for selecting appropriate values for κ, α and β.

Using computer simulations, the next section will outline the verification of the proposed UKF-based sensor fusion scheme to the reconstruction of the kinematic state of a Stewart platform.

4. Verification using computer simulations

Before applying the proposed sensor fusion scheme to an actual Stewart platform, the method is evaluated using computer simulations in combination with a representative kinematic model of the Stewart platform

of the SRS at TU Delft (see Figure 1). The parameters of the SRS's platform geometry are summarized in Table 1. The gravitational acceleration is assumed to be 9.80665 m/s^2.

Table 1: Constants related to the geometry of the SRS (see Figure 2).

Constant	**Value**	**Unit**
r_a	1.60	m
r_b	1.65	m
d_a	0.20	m
d_b	0.60	m
l_{min}	2.08	m
l_{max}	3.33	m

These parameters, in combination with artificially generated and realistic motion profiles, are used to create simulated IMU and actuator position measurements that will be polluted using artificially generated measurement noises. The noise-polluted signals are subsequently passed to the UKF for evaluation. All computer simulations are performed using Python, with the open-source scientific libraries Numpy, Scipy and Matplotlib [57, 58]. The simulation data is sampled at a rate of 100 Hz and, where necessary, use is made of the forward Euler method for numerical integration.

4.1. Motion profiles

The simulated sensor measurements are generated from two distinct motion profiles, both illustrated in Figures 5 and 6. The first, denoted as Motion Profile 1 (MP1), is selected because it activates most of the six DOFs of the Stewart platform with relatively large amplitudes and therefore excites the highly nonlinear kinematics of the platform. After a lead-in period of 10 s, the UGP of the platform traces a circular path with a radius of 0.5 m in the horizontal plane with a period of five seconds. Superimposed on this circular motion is a periodic roll and pitch motion with an amplitude of 10 deg/s and a period of 2.5 s. This results in a maximum actuator travel of approximately 30 cm with respect to the neutral length of 2.705 m.

The second motion profile, denoted as Motion Profile 2 (MP2), is more representative for typical experiments performed on the SRS. This motion profile has been recorded as part of a human-in-the-loop (manual tracking) experiment [59] and features a relatively high-frequency combined roll and pitch motion of small amplitude (max. of approximately 7 deg/s). In this case, the maximum actuator travel from the neutral position is in the order of 10 cm. It is therefore difficult to distinguish the motion of the platform for MP2 on the basis of Figure 6. Several representative time traces of specific force, angular rate and actuator position corresponding to both motion profiles are therefore also shown in Figure 7.

(a) $t = 20.00$ s (b) $t = 21.25$ s (c) $t = 22.50$ s (d) $t = 23.75$ s

Figure 5: Impression of motion platform pose at various time instants during MP1.

(a) $t = 11.50$ s (b) $t = 16.80$ s (c) $t = 55.30$ s (d) $t = 69.20$ s

Figure 6: Impression of motion platform pose at various time instants during MP2.

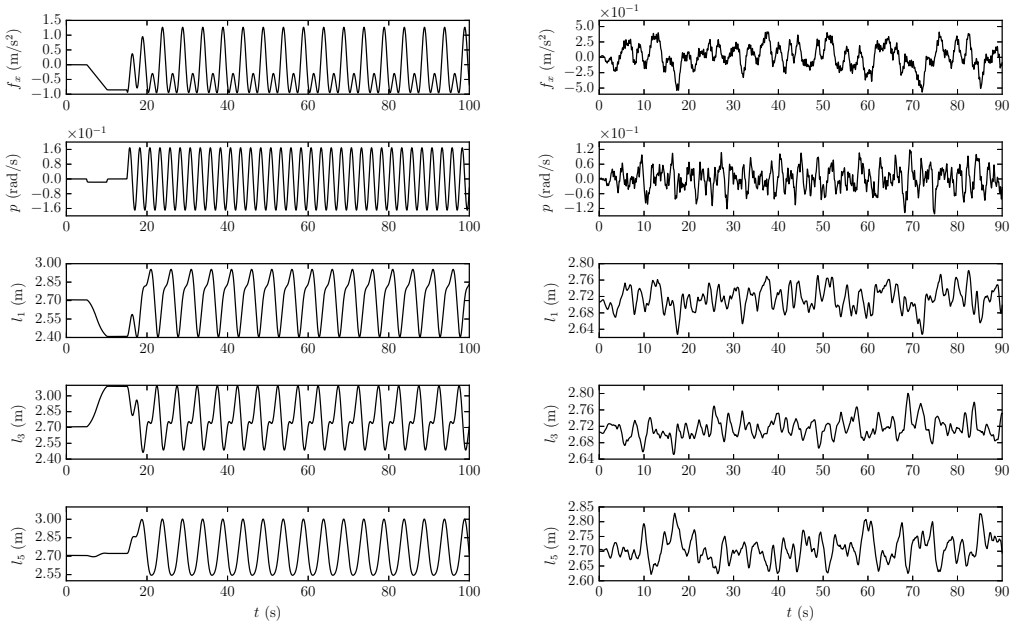


Figure 7: Representative time traces corresponding to MP1 (left) and MP2 (right).

4.2. Simulated sensor data

The selected motion profiles are superimposed with simulated zero-mean Gaussian noise signals to obtain the artificial sensor measurements used for verification of the proposed algorithm. These noise signals are fully defined by the standard deviation of the Gaussian distribution used to sample the simulated measurements. The inertial sensors are assumed to be polluted by a constant bias as well. Estimates of appropriate standard deviations and bias levels for the available inertial and actuator position sensors are obtained from static measurements of the available measurement equipment for the SRS. The measurement equipment on the SRS comprises an ISIS® IMU Rev. C manufactured by the Inertial Science Inc.® and six Temposonics® R-series linear position sensors manufactured by the MTS® Systems [60].

Figure 8 shows stationary measurements from a representative selection of sensors. Even though the actual sensor noise is evidently non-Gaussian, for the simulated sensor measurements Gaussian sequences are assumed. This allows for an evaluation of the theoretical accuracy attainable by the algorithm. The static IMU and actuator position measurements are used to obtain estimates of noise standard deviations and bias values, a summary of which can be found in Tables 2 and 3, respectively. The values shown in these tables, together with the available motion profile data, are finally used to generate the simulated IMU and actuator position measurements. A selection of these simulated signals is shown right in Figure 8.

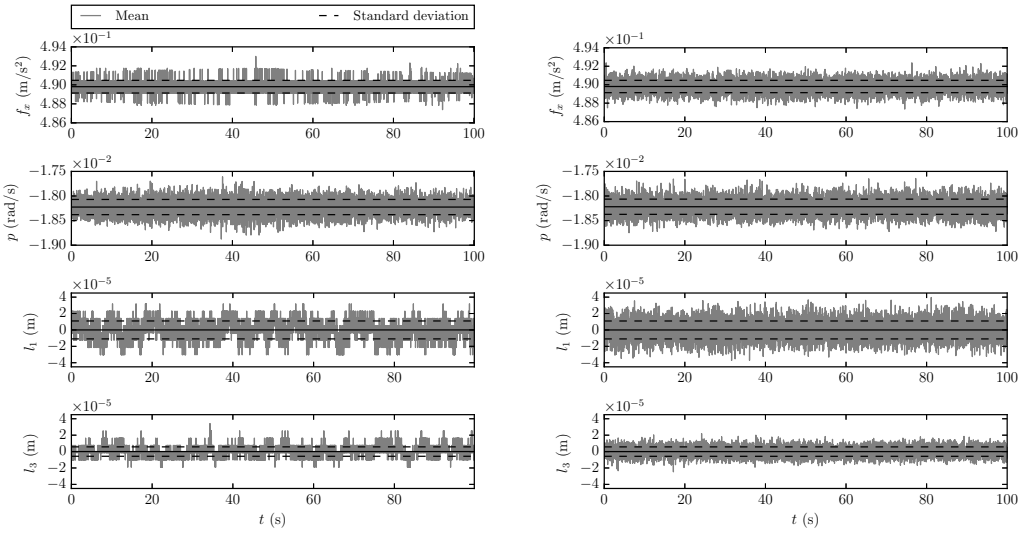


Figure 8: Selection of actual (left) and simulated (right) IMU and actuator position measurements for stationary motion base.

4.3. Method

The performance of the UKF-based algorithm is assessed on the basis of the accuracy and robustness of the algorithm. Accuracy reflects the error in the estimated quantities, while robustness pertains to the resilience of the algorithm to nonlinearities and model errors. Because the true values of estimated quantities

Table 2: Estimates of standard deviations of measurement noises in IMU and actuator position measurements.

Parameter	Value	Unit	Parameter	Value	Unit
σ_x	$6.680 \cdot 10^{-4}$	m/s^2	σ_{l_1}	$1.093 \cdot 10^{-5}$	m
σ_y	$8.578 \cdot 10^{-4}$	m/s^2	σ_{l_2}	$1.132 \cdot 10^{-5}$	m
σ_z	$6.915 \cdot 10^{-4}$	m/s^2	σ_{l_3}	$5.741 \cdot 10^{-6}$	m
σ_p	$1.545 \cdot 10^{-4}$	rad/s	σ_{l_4}	$1.482 \cdot 10^{-5}$	m
σ_p	$1.770 \cdot 10^{-4}$	rad/s	σ_{l_5}	$1.054 \cdot 10^{-5}$	m
σ_q	$1.791 \cdot 10^{-4}$	rad/s	σ_{l_6}	$9.770 \cdot 10^{-6}$	m

Table 3: Estimates of IMU measurement biases.

Parameter	Value	Unit
λ_x	$4.898 \cdot 10^{-1}$	m/s^2
λ_y	$-9.023 \cdot 10^{-3}$	m/s^2
λ_z	$-1.894 \cdot 10^{-1}$	m/s^2
λ_p	$-1.822 \cdot 10^{-2}$	rad/s
λ_p	$-5.067 \cdot 10^{-3}$	rad/s
λ_q	$-2.159 \cdot 10^{-2}$	rad/s

are often unknown in practice, accuracy and robustness are commonly evaluated on the basis of the so-called *innovation sequence*. The innovation appears in the correction step of the UKF as $\epsilon_k = \left(\boldsymbol{y}_k - \underline{\hat{\boldsymbol{y}}}_k^{(-)}\right)$. It can therefore be interpreted as the difference between the measured actuator positions, $\boldsymbol{y}_k$, and the reconstructed measurements obtained from the estimated kinematic state of the Stewart platform. Similarly, the *innovation covariance* appears in the computation of the Kalman gain as $P_{\epsilon,k} = P_{\underline{yy},k}$, i.e., the estimated covariance of the reconstructed measurements.

In case the process and observation noises $\underline{\boldsymbol{w}}$ and $\underline{\boldsymbol{v}}$ are both zero-mean Gaussian white noise, the innovation sequences should also be zero-mean, Gaussian and white for optimal filter performance [61]. In practice, this condition is most commonly verified by the observation that approximately 95 percent of the innovation sequence is within $\pm 2\sigma_\epsilon$, where σ_ϵ is the standard deviation of the innovation obtained from the diagonal elements of P_ϵ [61]. The rate of convergence and similarity of the innovation to such an "optimal" sequence is a measure of robustness. Since the current section deals with verification of the algorithm using *simulated* sensor measurements, however, additional information can be derived from the *estimation errors* in the estimated states.

Finally, for practical reasons, the quaternions in the estimated stochastic process state vector are kept normalized in both the prediction and correction steps of the UKF according to $\hat{\boldsymbol{\Phi}}_n = \hat{\boldsymbol{\Phi}}/\|\hat{\boldsymbol{\Phi}}\|$ [50].

4.4. Tuning of the UKF

Before the proposed algorithm can be applied to simulated sensor measurements, several parameters need to be configured. The UKF needs an initial estimate of the stochastic process state and its corresponding covariance as well as information about the process and observation noise covariances. In order to evaluate the algorithm in the most general sense, the initial process state estimate and auto-covariance are chosen arbitrarily far away from the true initial process state as:

$$\begin{aligned} \underline{\hat{\boldsymbol{x}}}_0 &= [0 \;\; 0 \;\; -1 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^\intercal \\ P_0 &= \operatorname{diag}\left(1^2, 1^2, 1^2, 1^2, 1^2, 1^2, 0.1^2, 0.1^2 \right. \\ &\qquad \left. 0.1^2, 0.1^2, 1^2, 1^2, 1^2, 0.1^2, 0.1^2, 0.1^2\right) \end{aligned} \tag{20}$$

The process and observation noise covariances selected are based on the values in Tables 2 and 3 as:

$$\begin{aligned} Q_k &= 1.1 \cdot \operatorname{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_p^2, \sigma_q^2, \sigma_r^2) \\ R_k &= 1.1 \cdot \operatorname{diag}(\sigma_{l_1}^2, \sigma_{l_2}^2, \sigma_{l_3}^2, \sigma_{l_4}^2, \sigma_{l_5}^2, \sigma_{l_6}^2) \end{aligned} \tag{21}$$

The ten percent increase in the magnitude of these covariances ensures some robustness against numerical round-off errors. In the current work, the system and observation noise covariance are assumed constant.

Other parameters that influence the UKF are κ, α and β, which are assigned values of 0, 0.1 and 2, respectively. These settings are in line with recommendations given in [55, 56]. Except for α, which if chosen too small degrades numerical stability, these parameters were not found to critically affect the obtained results.

4.5. Results

The proposed algorithm is subsequently applied to both MP1 and MP2. A representative selection of the obtained results is shown in Figures 9, 10 and 11. It is evident that the UKF shows quick convergence and optimal filter performance for both motion profiles, as witnessed by the seemingly white and Gaussian innovation sequences. In addition, both the platform states as well as the IMU biases are estimated with a high accuracy. For most of the estimated quantities, the obtained accuracy is equivalent regardless of the motion profile evaluated. However, a periodic trend in the estimation errors of λ_z, w and z for the case of MP1 can observed. Such fluctuations are typically associated with sub-optimal filter performance, although in this particular case the fluctuations do not visibly propagate through to the innovation sequences.

The results presented here confirm that the SUT is an excellent alternative to local linearization of the nonlinear system. Many of the shortcomings of the EKF/IEKF uncovered in earlier work [45, 47], such as slow convergence and a high sensitivity to chosen inititial conditions, are overcome by the application of the UKF. The pronounced fluctuations in λ_z, w and z for MP1 can nonetheless be explained by the physical limitations of the SUT inherent in the UKF. The SUT still uses *only* the mean and covariance to describe

the posterior probability distribution of the estimated quantities. Higher order moments of the posterior probability distributions are not taken into account, even though they may be important for fully capturing the effect of the highly nonlinear platform kinematics for large motions.

As such, it can be expected that the more violent the platform motion, i.e., the more pronounced the nonlinear platform kinematics, the more the performance of the UKF will degrade. Nonetheless, even for the case of MP1, which is considered extreme for all practical intents and purposes, position is estimated to an accuracy of $O(10^{-5})$ m, velocity to $O(10^{-4})$ m/s, attitude to $O(10^{-6})$ rad and the IMU biases to $O(10^{-5})$ m/s^2 and rad/s. In flight simulation, accuracy requirements are closely related to human motion perception thresholds. These are in the order of $O(10^{-2})$ m/s^2 for translations and $O(10^{-3})$ rad/s^2 for rotations [62]. Therefore, the accuracy attainable by the proposed algorithm is considered sufficient by a large margin for the purpose of flight simulator motion fidelity assessment. However, requirements for other applications may be far more stringent. An example is precision surgery, where a positional accuracy of the platform in the order of micrometers is required [4].

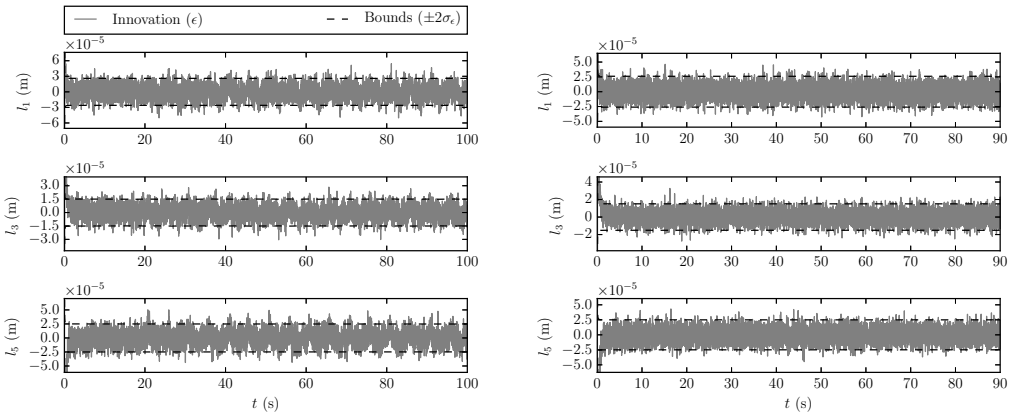


Figure 9: Selection of innovation sequences for MP1 (left) and MP2 (right).

5. Validation on the SRS

The proposed state reconstruction algorithm is subsequently implemented and validated on the SIMONA Research Simulator (SRS) at TU Delft. First, a real-time implementation of the algorithm proposed in Section 3 is discussed, after which the method of validation will be presented. Then, the tuning of the algorithm for application on the SRS will be discussed, followed by a presentation of the obtained results. Finally, the validation is conclused with a comparison to a conventional iterative method (e.g., [10]) used to reconstruct the kinematic state of a Stewart platform.

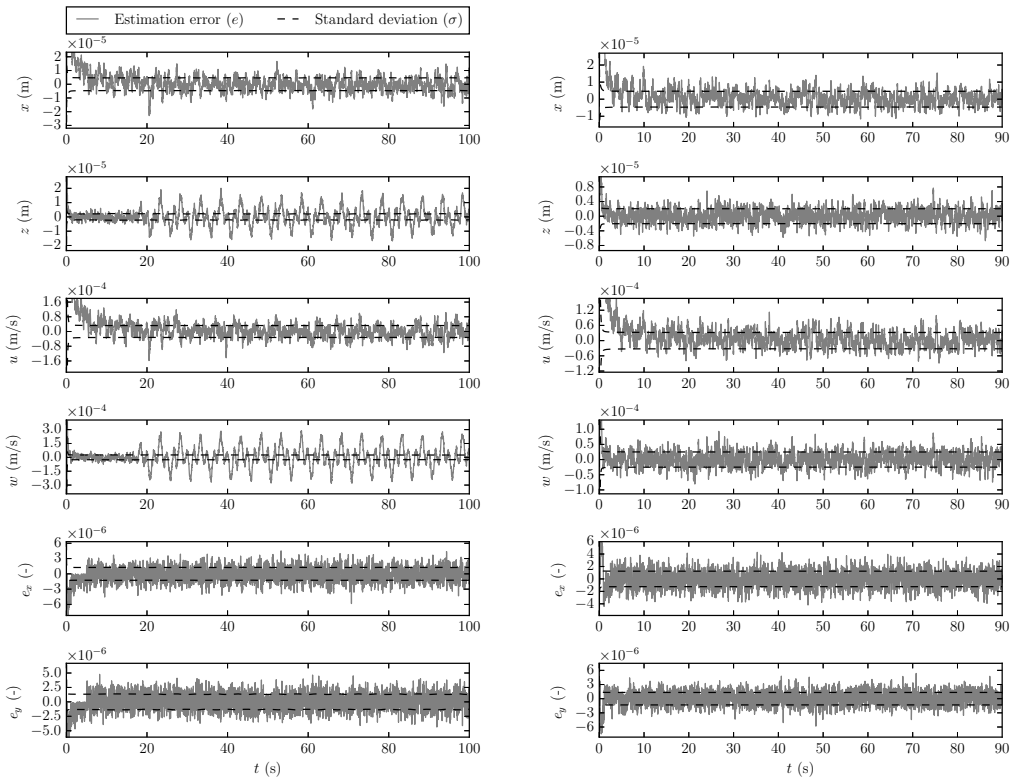


Figure 10: Selection of errors in estimated platform states for MP1 (left) and the MP2 (right).

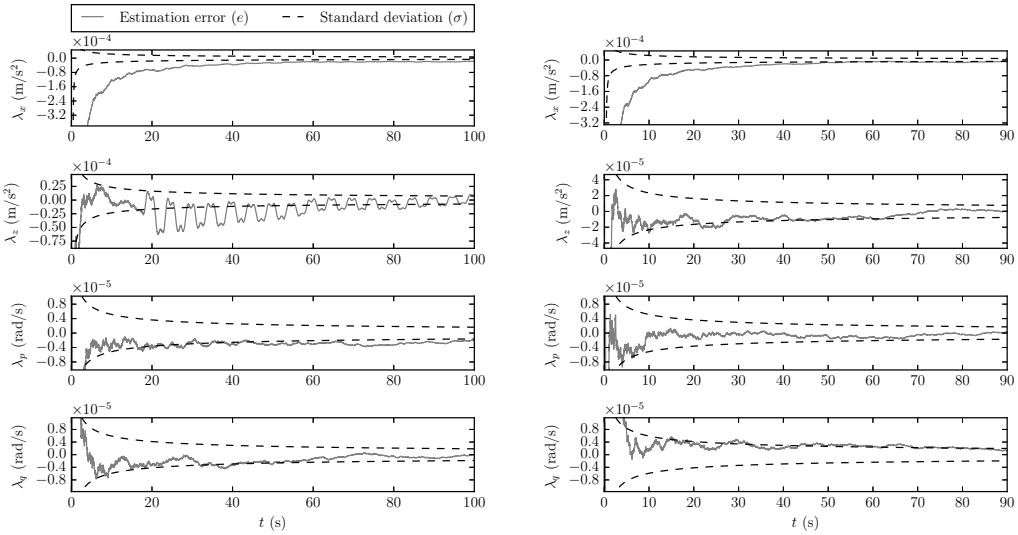


Figure 11: Selection of errors in estimated IMU biases for MP1 (left) and the MP2 (right).

5.1. Real-time implementation

In order to be able to execute the proposed state reconstruction algorithm on the SRS in real-time and using physical sensor measurements, the algorithm was implemented in the software architecture of the SRS, namely the Delft University Environment for Communication and Activation (DUECA) [63]. DUECA is a middleware software layer that takes care of many of the complex issues that arise from developing software that is suitable for execution on an operational full-flight simulator in real-time. Typical functionality includes the synchronization of processes with wall clock time, inter-process communication within a distributed computing environment and communication with simulator hardware (e.g., sensors, controls, displays, etc.).

The majority of the code base of DUECA is written in the C and C++ computer languages, to allow for high performance and favourable real-time characteristics. Software developed for the SRS is typically encapsulated in so-called DUECA modules, which use the services provided by DUECA to ensure adherence to the various computational requirements and constraints. The proposed algorithm was developed as a separate module in DUECA and thus in the C++ computer language. To support the complex matrix algebra inherent in the UKF-based algorithm, use was made of the open-source Eigen library [64]. This implementation was found to require an average computational time in the order of a few tenths of a millisecond per measurement sample on modern desktop and server hardware. It is therefore considered more than adequate for execution on the SRS in real time at a nominal rate of 100 Hz.

5.2. Method

The method employed to validate the proposed algorithm is identical to the one used for the verification outlined in Section 4. That is, the same motion profiles (MP1 and MP2) will be used and the same criteria to assess the performance of the algorithm will be applied. There are, however, a number of additional limitations and practical issues to consider in a real world application. One important limitation is the lack of knowledge of the true kinematic state of the platform. In case of simulated sensor measurements, the true state of the platform is known, such that the estimation error in each of the estimated states can be evaluated (e.g., see Figures 10 and 11). In practice, the true states are of course not available and the innovation sequences must be largely relied upon to assess the performance of the algorithm. However, in Section 5.5, a comparison to a traditional method which relies only on the actuator positions is presented. Also, as shown in Figure 8, the actual sensor noise may be non-Gaussian, such that a degraded performance can be expected.

An important issue to also take into consideration is the mounting location of the IMU. Until this point, it was assumed that the the IMU was located in the UGP of the platform, i.e., the origin of reference frame E_a in Figure 2. In reality, this may not be the case for various practical reasons. As such, the algorithm needs to be adapted to account for a generic IMU location. To illustrate the issue, the difference between

the specific forces in the UGP (or any arbitrary point on the platform) and at the IMU location can be written as:

$$\boldsymbol{f}_{imu} - \boldsymbol{f}_{ugp} = \boldsymbol{c}_{imu}^{\mathsf{T}} \left(\dot{\Omega}_{\times} + \Omega_{\times}^2 \right) \tag{22}$$

where $\boldsymbol{c}_{imu}$ is the location of the IMU with respect to the UGP and where $\Omega_{\times}$ is given by Equation (15). This equation shows that there is a mismatch between the specific forces in any two points on the moving platform, which arises from the rotation of the platform. An elegant way to correct for this mismatch in the proposed algorithm is to adapt the definition of the joint coordinates $\boldsymbol{a}_i$ appearing in Equation (1) as:

$$\boldsymbol{a}_i^{imu} = \boldsymbol{a}_i - \boldsymbol{c}_{imu} \quad \forall \quad i \in [1, \ldots, 6] \tag{23}$$

This effectively moves the origin of frame E_a in Figure 2 to the IMU location. The consequence of this approach is that the complete estimate of the stochastic process state inferred by the algorithm corresponds to the IMU location. The state vector can, however, be re-calculated with respect to the UGP or any arbitrary location on the platform using Equation (22) and its equivalent for platform velocity and position.

Other issues that affect the performance of the proposed algorithm include: actuator position bias, time-varying IMU biases, measurement time synchronization errors and IMU misalignment. Care was taken to calibrate the actuator position sensors using laser-based distance measuring equipment so as to ensure the biases are no more than a few tenths of a millimetre in magnitude. For the brief motion profiles considered in the current work, time-dependent variations in the IMU biases are negligible. It should be noted, however, that for more prolonged measurements variations in the IMU biases can be accounted for by introducing so-called random walks (e.g., [56]). This entails modeling the evolution of the biases in the stochastic system model as artificial noises. Time delays between the inertial sensors and generic auxilliary position measurements are known to degrade accuracy, e.g., [65], and methods to account for such delays have been proposed in the literature, e.g., [56, 66, 67]. Their application, however, would entail a considerable increase in the complexity of the proposed algorithm and are therefore beyond the scope of the current work. IMU misalignment causes couplings between measurements from the rate gyros in the IMU. This effect can be minimized by careful alignment of the IMU with the orthogonal platform axes, but one could also attempt to incorporate rate gyro misalignment in the stochastic system model. This too is beyond the scope of the current work, however.

5.3. Tuning of the UKF

Before the proposed algorithm was executed on the SRS in real-time, actual IMU and actuator position measurements corresponding to MP1 and MP2 were collected for off-line analysis and tuning. In this analysis, the sensitivity of the algorithm to changes in the governing parameters of the UKF-based algorithm was investigated. It was found that all but one of the UKF tuning parameters could remain unchanged with respect to the settings presented in Section 4. This parameter turned out to be Q_k, i.e., the system noise

covariance matrix. The magnitude of Q_k had to be increased by a factor of a thousand in order to obtain favourable characteristics of the innovation sequences (see Section 4):

$$Q_k = 1000 \cdot \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_p^2, \sigma_q^2, \sigma_r^2) \tag{24}$$

where all the standard deviations used are as given in Table 2. Changes in the magnitude of this matrix were found to have by far the largest impact on the results, whereas the state reconstruction algorithm seemed less sensitive to moderate changes in the magnitude of R_k. It is also not desirable to increase the magnitude of R_k in this case, since the actuator position measurements are the sole basis for estimation of the biases inherent in the IMU measurements. Moreover, as mentioned in Section 5.2, some deficiencies in the IMU measurements are currently unaccounted for, e.g., IMU misalignment and time delays between IMU and actuator position measurements. A more detailed analysis into the magnitude and effect of time delays between the two sets of measurements was performed. This analysis revealed that the IMU measurements are delayed by 10 to 50 ms with respect to the actuator position measurements. The delays were furthermore found to be axis dependent and fluctuating in time. An explanation for the delays on the software level (i.e., DUECA) could not be found and the issue is therefore likely caused by the IMU itself. Regardless of the exact cause, it seems evident that from the perspective of the UKF, the IMU measurements are less reliable than the actuator position measurements. This is reflected in the significantly higher magnitude of Q_k with respect to R_k required to obtain favourable filter performance.

As discussed in Section 5.2, the last parameter required by the algorithm for application on the SRS is the mounting location of the IMU. In the SRS, the IMU is located 0.305 behind and 0.0105 meters above the UGP, such that:

$$\boldsymbol{c}_{imu} = \begin{bmatrix} -0.305 & 0 & -0.0105 \end{bmatrix}^\intercal \tag{25}$$

5.4. Results

The obtained results from real-time execution on the SRS are included in Figures 12, 13 and 14, which show a selection of innovation sequences, state estimates and IMU bias estimates corresponding to both MP1 and MP2. Several conclusions can be drawn from these results.

First, from the innovation sequences, it can be seen that rapid convergence of the algorithm to zero-mean values occurs for both motion profiles. Also, the innovation sequences are largely within the band that is bounded by $2\sigma_\epsilon$. The amplitude of this band is in the range of only several tenths of a millimeter, indicating high accuracy of the estimated states. However, the innovation sequences are evidently not characterized as white and are therefore indicative of sub-optimal filter performance [61]. This could be expected, given the remaining (unmodelled) measurement uncertainties and the fact that the actual measurement noise is non-Gaussian. It can be seen that the innovation sequences corresponding to MP1 clearly exhibit a component that resembles the periodicity of the motion profile. This effect is less pronounced in case of MP2, which is

explained by the smaller amplitude of the motion in this profile relative to the motion in MP1. In addition, a number of short periods with an elevated value of the innovation sequences appear during both MP1 and MP2. The cause of these "periodic beats" was traced to the actuator position measurements, which were found to exhibit an identical periodic elevation in noise level. This increase in noise level was found to occur synchronously over all six actuator position measurements and therefore could not be isolated to a single (faulty) sensor. Consequently, the problem is likely situated in a databus that is affected by some external disturbance. As it does not impact the validity of the presented results, however, a further consideration of the issue is beyond the scope of the current work.

Regardless of all practical issues encountered, the state estimates shown in Figure 13 are highly accurate nonetheless, as also illustrated by the generally small standard deviations calculated from the diagonal elements of the covariance matrix P_k. The estimates of the inertial sensor biases, shown in Figure 14, show a trend of somewhat slower convergence similar as observed in the results from the computer simulations presented in Section 4. A comparison to the static sensor measurements shown in Figure 8, however, reveals that their values are also estimated accurately after a brief period of convergence. Note that a minor mismatch may exist between the estimated IMU biases and the static measurements. This is because the static measurements were recorded at a different date and time than the validation experiments were performed. As such, temperature variations may slightly influence the true values of the inertial sensor biases and small variations in their estimated values may therefore be observed when the experiments are repeated at different times or for an extended amount of time.

The promising results obtained from the validation experiments motivate a comparison of the proposed state reconstruction algorithm to a traditional method to estimate the kinematic state of a Stewart platform. This comparison is the subject of the next section.

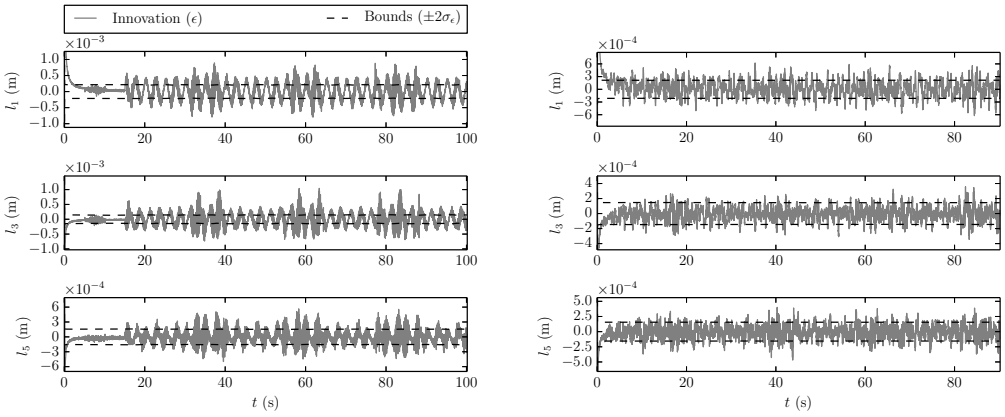


Figure 12: Selection of innovation sequences from real-time sensor data for MP1 (left) and MP2 (right).

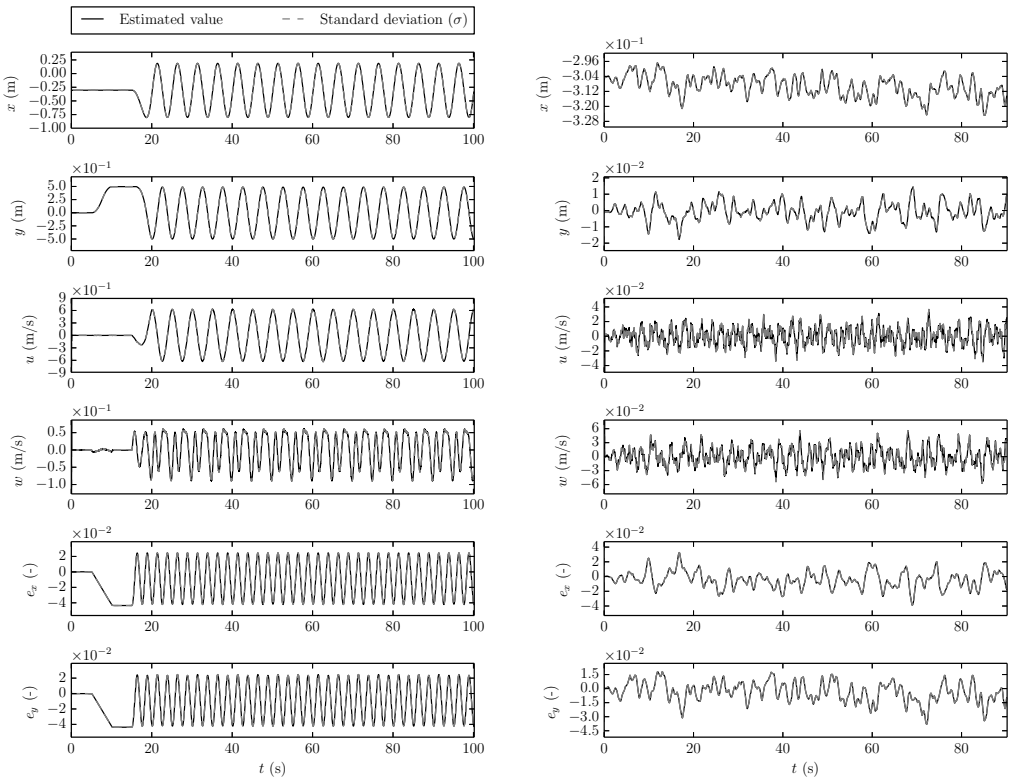


Figure 13: Selection of estimated platform states from real-time sensor data for MP1 (left) and MP2 (right).

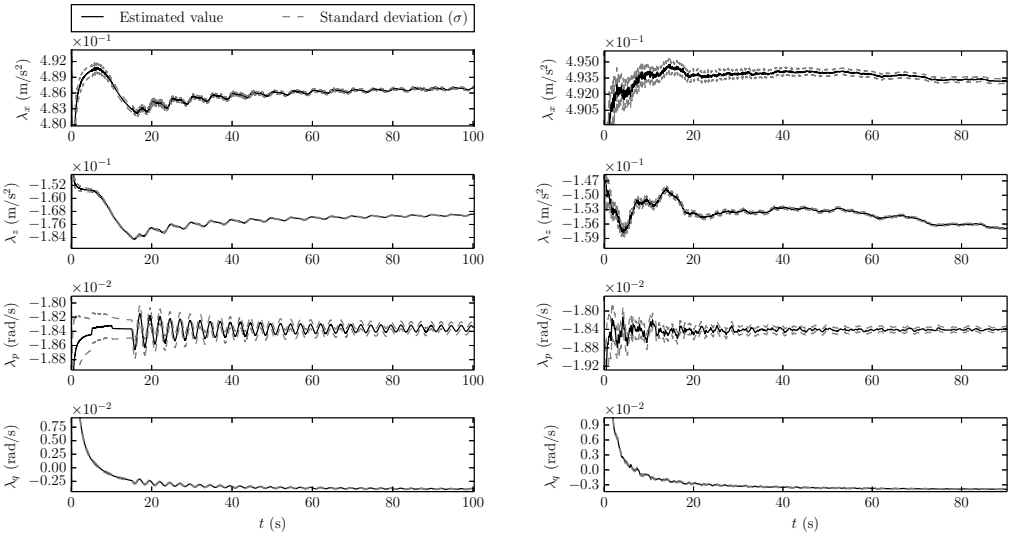


Figure 14: Selection of estimated IMU biases from real-time sensor data for MP1 (left) and MP2 (right).

5.5. Comparison with an iterative method

In Section 5, it was shown that the proposed UKF-based state reconstruction algorithm seems to converge to accurate estimates of the kinematic state of the SRS and is also able to infer inherent IMU biases. This brings about two distinct advantages over contemporary iterative schemes commonly applied to solve the forward kinematics of a Stewart platform (e.g., [27, 10]), which only provide direct estimates of position and attitude. In such traditional methods, to obtain (angular) velocity and acceleration, numerical differentiation is required. As demonstrated, the proposed UKF-based algorithm does not rely on any numerical differentiation to infer velocity and translational acceleration. Translational velocity is an element of the stochastic state vector (see Equation (10)) and is therefore inferred directly. Estimates of translational acceleration and angular velocity can be obtained from raw IMU measurements, corrected for bias using real-time estimates from the proposed UKF-based algorithm.

In order to fully validate the proposed UKF-based state reconstruction algorithm, it is compared to a traditional iterative scheme commonly applied to solve for the forward kinematics of Stewart platforms. The algorithm selected is one that is currently used within the motion control system of the SRS [10], namely the Gauss-Newton (GN) method given by Equation (5). In order to compare the two algorithms, the error with respect to set-points is computed for both MP1 and MP2. Note that this includes *all* errors, i.e., control system errors, transport delays and estimation errors. Some representative results are included in Figure 15. This figure shows excerpts of timetraces of the errors in position, attitude and velocity estimated using both algorithms for both motion profiles.

The GN-based algorithm does not provide a direct estimate of platform velocity. For a fair comparison, use is therefore made of a differentiator combined with a second-order Butterworth filter to compute velocity from estimated position [68]:

$$B_2(s) = \frac{Y(s)}{U(s)} = \frac{s}{\tau_d^2 s^2 + \sqrt{2}\tau_d s + 1} \tag{26}$$

The cut-off frequency of the filter is set to 25 Hz (i.e., τ_d of 0.04 s). Also note that, as the UKF-based state estimates are inferred with respect to the location of the IMU, they must be translated to the platform's UGP for comparison to the GN-based estimates. This is done using the following kinematic entities:

$$\begin{aligned} \hat{\boldsymbol{c}}_{ugp} &= \hat{\boldsymbol{c}} - T_{ba}\boldsymbol{c}_{imu} \\ \hat{\boldsymbol{V}}_{ugp} &= \hat{\boldsymbol{V}} - (\Omega_\times - \hat{\Lambda})^\intercal \boldsymbol{c}_{imu} \end{aligned} \tag{27}$$

where $\Omega_\times$ and Λ are as defined in Equation (15) and $\boldsymbol{c}_{imu}$ is given by Equation (25). The notation $\hat{\square}$ signifies estimates that are derived from the UKF-based algorithm. Also note that $\Omega_\times$ contains *raw* angular rate measurements from the IMU.

From inspection of Figure 15, it seems that the UKF-based algorithm generally attains better accuracy than the GN-based algorithm. The difference is most apparent for MP1, where the error due to the phase

lag incurred by the differentiating filter is clearly visible in the GN-based velocity estimate. This frequency-dependent phase lag can be reduced by increasing the cut-off frequency of the filter, but this results in more noise in the GN-based velocity estimates.

To conclude the comparison between the two algorithms, the corresponding standard deviations of the errors in the estimates are illustrated in Figures 16 and 17 for MP1 and MP2, respectively. It can be seen that the UKF-based algorithm provides slightly improved estimates of position and attitude for both MP1 and MP2. However, by far the largest difference in favor of the UKF-based algorithm is observed in the estimates of platform velocity for MP1. Here, the u and v estimates are found to be approximately twice as accurate for the UKF-based algorithm. For MP2, the difference between the two methods is less pronounced. This can be explained by the different frequency spectrum of MP2 as compared to MP1. In this case, this means that the phase lag induced by the differentiating filter in the GN-based method has a smaller impact on the observed error for MP2 than for MP1. As a final note, it can be seen that the accuracy of both algorithms is equivalent for those kinematic states that are barely excited, i.e., z, ψ and w for MP1 and y, ψ and v for MP2.

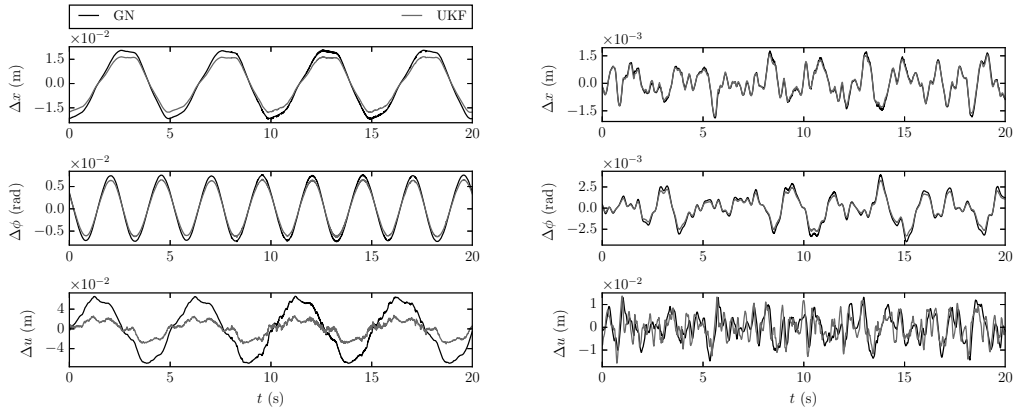


Figure 15: Selection of estimation errors for UKF- and GN-based method to compute platform states in case of MP1 (left) and MP2 (right).

6. Discussion

In this paper, a novel algorithm to reconstruct the kinematic state of a Stewart platform in an operational environment, namely that of the SIMONA Research Simulator (SRS) at TU Delft, was investigated. The proposed algorithm relies on the fusion of inertial and actuator position sensors, realised using the Unscented Kalman Filter (UKF) in conjunction with a purely kinematic model of a Stewart platform. This method enables direct estimation of platform position, attitude and translational velocity. By means of this so-called tightly-coupled sensor fusion, it also becomes possible to account for inherent measurement uncertainties,

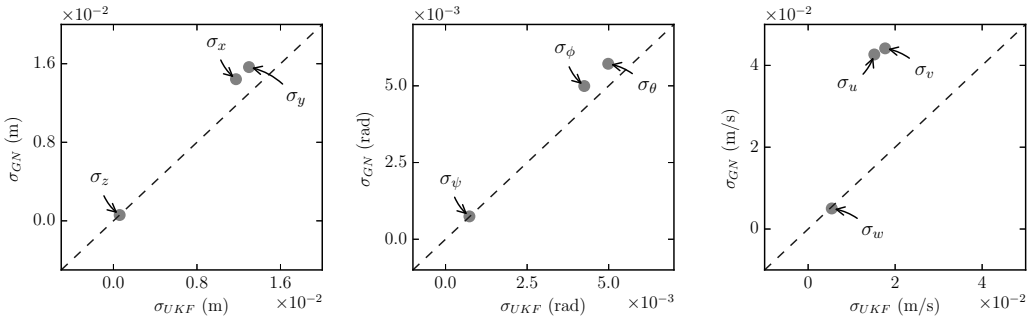


Figure 16: Standard deviations of errors in UKF- and GN-based estimates for MP1.

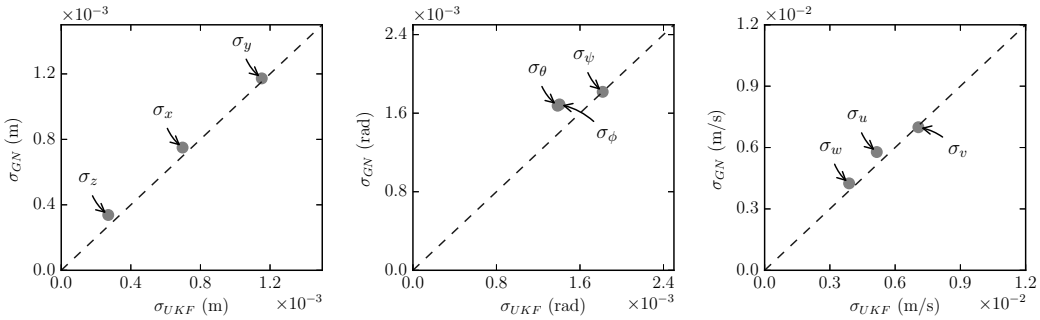


Figure 17: Standard deviations of errors in UKF- and GN-based estimates for MP2.

such as noise and inertial sensor bias. Furthermore, because of the recursive nature of the algorithm, it is possible to execute the algorithm in real-time.

The main contribution of this work is twofold. First, a rigorous verification of the proposed algorithm, using off-line simulations with simulated sensor measurements, was presented. This verification has demonstrated, using two representative motion profiles, the theoretically attainable performance of the algorithm in terms of accuracy and robustness. Some minor signs of sub-optimal filter performance could be observed due to the strongly nonlinear kinematics of the Stewart platform. However, the overall performance of the algorithm is remarkable, yielding highly accurate estimates and exhibiting a rapid rate of convergence without a need for careful tuning of configuration parameters and initial conditions. These results served as the basis for the second contribution of this work, namely a real-time implementation and validation of the novel algorithm on the SRS. To this end, an efficient implementation within DUECA, the software framework of the SRS, was conceived. This implementation was found to easily meet the requirements for real-time execution on the SRS.

During the tuning of the algorithm, a number of practical issues arising from the application in an operational environment were encountered. While some of these issues could be resolved in a relatively straightforward manner, others were deemed beyond the scope of the current work. These issues relate to

measurement time synchronization, IMU misalignment and non-stationary noise levels in measurements. Nonetheless, the overall results obtained from the validation were satisfying. That is, a direct comparison of the results to those obtained from a traditional iterative scheme, commonly applied to solve the forward kinematics of Stewart platforms, were in favor of the proposed UKF-based algorithm.

There are a number of areas in which the proposed approach can be extended and improved. First, potential issues pertaining to the sensory equipment and data acquisition system should be investigated and resolved. These include minimizing measurement time synchronization errors, IMU misalignment, and removing any external disturbances in the data acquisition system that may affect measurement noise levels. Therefore, even though the present results already highlight the potential of the proposed algorithm as an improved estimator for the kinematic state of a Stewart platform, the UKF-based algorithm can be improved even further by, e.g., the use of a more modern and accurate IMU. Second, the robustness of the state reconstruction algorithm itself could be improved as well. In the current work, the UKF system and observation noise covariances were held constant, however, it could be investigated if adaptive modification of these matrices yields improved results (see, e.g., [69, 70]). Finally, a distinct advantage of the proposed algorithm is its extendability with additional sensor measurements. For example, given angular acceleration measurements [71], the kinematic model of the Stewart platform at the heart of the state reconstruction algorithm could easily be modified to enable direct inference of angular velocity and unbiased angular acceleration.

7. Conclusion

In this paper, a novel method to reconstruct the kinematic state of a Stewart platform was presented and validated using real-time sensor measurements on the SIMONA Research Simulator (SRS) at TU Delft. The proposed method relies on an implementation of the Unscented Kalman Filter (UKF) to enable a tighly-coupled fusion of on-platform inertial sensor data with measurements of actuator position. In a direct comparison, the approach is shown to have two main advantages over conventional iterative methods for estimating the state of Stewart platforms in real time, enabling direct and accurate estimation of translational platform velocity, as well as an inherent robustness to measurement uncertainties like sensor noise and bias. The current work therefore demonstrates the effectiveness of the proposed method for Stewart platform state estimation and its potential for supporting improved motion control and simulator motion fidelity verification.

References

[1] D. Stewart, A Platform with Six Degrees of Freedom, Proceedings of the Institution of Mechanical Engineers 180 (1965) 371–386.

[2] Y. Ting, Y. S. Chen, H. C. Jar, Y. Kang, Modeling and Control for a Gough-Stewart Platform CNC Machine, in: Proceedings of the IEEE International Conference on Robotics & Automation, New Orleans, LA, 26 Apr. - 1 May, volume 1, 2004, pp. 535–540.

[3] R. Alves De Sousa, J. Ferreira, J. Sa De Farias, J. Torro, D. Afonso, M. Martins, SPIF-A: On the Development of a New Concept of Incremental Forming Machine, Structural Engineering and Mechanics 49 (2014) 645–660.

[4] M. Wapler, V. Urban, T. Weisener, J. Stallkamp, M. Dürr, A. Hiller, A Stewart Platform for Precision Surgery, Transactions of the Institute of Measurement and Control 25 (2003) 329–334.

[5] L. Kratchman, G. Blachon, T. Withrow, R. Balachandran, R. Labadie, R. Webster, Design of a Bone-Attached Parallel Robot for Percutaneous Cochlear Implantation, IEEE Transactions on Biomedical Engineering 58 (2011) 2904–2910.

[6] M. Girone, G. Burdea, M. Bouzit, V. Popescu, J. Deutsch, Stewart Platform-Based System for Ankle Telerehabilitation, Autonomous Robots 10 (2001) 203–212.

[7] Q. Liu, D. Liu, W. Meng, Z. Zhou, Q. Ai, Fuzzy Sliding Mode Control of a Multi-DOF Parallel Robot in Rehabilitation Environment, International Journal of Humanoid Robotics 11 (2014).

[8] O. Stroosma, M. M. Van Paassen, M. Mulder, Using The SIMONA Research Simulator For Human-Machine Interaction Research, in: AIAA Modeling and Simulation Technologies Conference and Exhibit, Austin, Texas, Aug. 11-14, AIAA-2003-5525, 2003, pp. 1–8.

[9] J. Bürki-Cohen, A. Sparko, M. Bellman, Flight Simulator Motion Literature Pertinent to Airline-Pilot Recurrent Training and Evaluation, in: AIAA Modeling and Simulation Technologies Conference, Portland, OR, Aug. 8-11, 2011, pp. 187–203.

[10] S. H. Koekebakker, Model Based Control of Flight Simulator Motion Bases, Ph.D. thesis, Delft University of Technology, 2001.

[11] I. Davliakos, E. Papadopoulos, Model-Based Control of a 6-dof Electrohydraulic Stewart-Gough Platform, Mechanism and Machine Theory 43 (2008) 1385–1400.

[12] Y. Pi, X. Wang, Trajectory Tracking Control pf a 6-DOF Hydraulic Parallel Robot Manipulator with Uncertain Load Disturbances, Control Engineering Practice 19 (2011) 185–193.

[13] S. H. Chen, L. C. Fu, Output Feedback Sliding Mode Control for a Stewart Platform With a Nonlinear Observer-Based Forward Kinematics Solution, IEEE Transactions on Control Systems Technology 21 (2013) 176–185.

[14] F. A. M. Van Der Steen, Self-Motion Perception, Ph.D. thesis, Delft University of Technology, Faculty of Aerospace Engineering, 1998.

[15] S. K. Advani, R. J. A. W. Hosman, M. Potter, Objective Motion Fidelity Qualification in Flight Training Simulators, in: AIAA Modeling and Simulation Technologies Conference and Exhibit, AIAA 2007-6802, 2007.

[16] O. Stroosma, M. M. van Paassen, M. Mulder, R. J. A. W. Hosman, S. K. Advani, Applying the Objective Motion Cueing Test to a Classical Washout Algorithm, in: Proceedings of the AIAA Modeling and Simulation Technologies Conference, Boston (MA), AIAA-2013-4834, 2013.

[17] Anonymous, ICAO Manual of Criteria for the Qualification of Flight Simulation Training Devices, 2009. Third Edition.

[18] R. J. A. W. Hosman, S. K. Advani, J. Takats, Status of the ICAO Objective Motion Cueing Test, in: Autumn Flight Simulation Conference: Flight Simulation Research - New Frontiers, Royal Aeronautical Society, Royal Aeronautical Society, 2012.

[19] B. Dasgupta, T. S. Mruthyunjayab, The Stewart Platform Manipulator: A Review, Mechanism and Machine Theory 35 (2000) 15–40.

[20] M. Raghavan, The Stewart Platform of General Geometry Has 40 Configurations, ASME Journal of Mechanical Design 115 (1993) 277–282.

[21] C. W. Wampler, Forward Displacement Analysis of General Six-In-Parallel SPS (Stewart) Platform Manipulators using Soma Coordinates, Mechanism and Machine Theory 31 (1996) 331–337.

[22] M. L. Husty, An Algorithm for Solving the Direct Kinematics of General Stewart Gough Platforms, Mechanism and Machine Theory 31 (1996) 365–380.

[23] A. K. Dhingra, A. N. Almadi, D. Kohli, A Gröbner-Sylvester Hybrid Method for Closed-Form Displacement Analysis of Mechanisms, ASME Journal of Mechanical Design 122 (2000) 431–438.

[24] X. S. Gao, D. Lei, Q. Liao, G. F. Zhang, Generalized Stewart-Gough Platforms and Their Direct Kinematics, IEEE Transactions on Robotics 21 (2005) 141–151.

[25] J. P. Merlet, Closed-Form Resolution of the Direct Kinematics of Parallel Manipulators using Extra Sensors, in: Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, 2-6 May, volume 1, 1993, pp. 200–204.

[26] J. P. Merlet, Solving the Forward Kinematics of a Gough-Type Parallel Manipulator with Interval Analysis, The International Journal of Robotics Research 23 (2004) 221–235.

[27] J. E. Dieudonne, R. V. Parrish, R. E. Bardusch, An Actuator Extension Transformation for a Motion Simulator and an Inverse Transformation applying Newton-Raphson's Method, Technical Report D-7067, NASA, 1972.

[28] L. C. T. Wang, C. C. Chen, On the Numerical Kinematic Analysis of General Parallel Robotic Manipulators, IEEE Transactions on Robotics and Automation 9 (1993) 272–285.

[29] W. Zhou, W. Chen, H. Liu, X. Li, A New Forward Kinematic Algorithm for A General Stewart Platform, Mechanism 87 (2015) 177–190.

[30] J. Y. Kang, D. H. Kim, K. I. Lee, Robust Estimator Design for Forward Kinematics Solution of a Stewart Platform, Journal of Robotic Systems 15 (1998) 29–42.

[31] S. H. Chen, H. C. I., L. C. Fu, Applying a Nonlinear Observer to Solve Forward Kinematics of a Stewart Platform, in: Proceedings of the IEEE International Conference on Control Applications, San Antiono, Texas, Sep. 3-5, 2008, pp. 1183–1188.

[32] S. A. Maged, A. M. F. E. Bab, A. Abouelsoud, An Adaptive Observer for a Stewart Platform Manipulator using Leg Position and Force Measurements, International Journal of Modelling, Identification and Control 24 (2015) 62–74.

[33] M. N. Armenise, C. Ciminelli, F. Dell'Olio, V. M. N. Passaro, Advances in Gyroscope Technologies, Springer, 2010.

[34] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Transactions of the ASME, Journal of Basic Engineering 82 (1960) 35–45.

[35] P. C. Lin, H. Komsuoglu, D. E. Koditschek, Sensor Data Fusion for Body State Estimation in a Hexapod Robot With Dynamical Gaits, IEEE Transactions on Robotics 22 (2006) 932–943.

[36] M. M. Shaikh, W. Bahn, C. Lee, T. I. Kim, T. J. Lee, K. S. Kim, D. Cho, Mobile Robot Vision Tracking System Using Unscented Kalman Filter, in: IEEE/SICE International Symposium on System Integration, Kyoto, Dec. 20-22, 2011, pp. 1214–1219.

[37] A. Assa, F. Janabi-Sharifi, A Robust Vision-Based Sensor Fusion Approach for Real-Time Pose Estimation, IEEE Transactions on Cybernetics 44 (2014) 217–227.

[38] H. L. Jonkers, Application of the Kalman Filter to Flight Path Reconstruction from Flight Test Data Including Estimation of Instrumental Bias Error Corrections, Technical Report VTH-162, Delft University of Technology, Faculty of Aerospace Engineering, 1976.

[39] J. J. Horsten, H. L. Jonkers, J. A. Mulder, Flight Path Reconstruction in the Context of Nonsteady Flight Testing, Technical Report LR-280, Delft University of Technology, Faculty of Aerospace Engineering, 1979.

[40] J. A. Mulder, Q. P. Chu, J. K. Sridhar, J. H. Breeman, M. Laban, Non-Linear Aircraft Flight Path Reconstruction: Review and New Advances, Progress in Aerospace Sciences 35 (1999) 673–726.

[41] P. Lu, L. Van Eykeren, E. J. Van Kampen, C. C. De Visser, Q. P. Chu, Double-Model Adaptive Fault Detection and Diagnosis Applied to Real Flight Data, Control Engineering Practice 36 (2015) 39–57.

[42] A. Vaccarella, E. De Momi, A. Enquobahrie, G. Ferrigno, Unscented Kalman Filter Based Sensor Fusion for Robust Optical and Electromagnetic Tracking in Surgical Navigation, IEEE Transactions on Instrumentation and Measurement 62 (2013) 2067–2081.

[43] G. Hovland, T. Von Hoff, E. Gallestey, M. Antoine, D. Farruggio, A. Paice, Nonlinear Estimation Methods for Parameter Tracking in Power Plants, Control Engineering Practice 13 (2005) 1341 – 1355.

[44] M. A. Louda, D. C. Rye, M. W. M. G. Dissanayake, H. F. Durrant-Whyte, INS-based Identification of Quay-Crane Spreader Yaw, in: Proceedings of the IEEE International Conference on Robotics & Automation, Leuven, Belgium, May 16-20, 1998, pp. 3310–3315.

[45] D. M. Pool, Q. P. Chu, M. Mulder, M. M. van Paassen, Optimal Reconstruction of Flight Simulator Motion Cues using Extended Kalman Filtering, in: Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Honolulu, Hawai, Aug. 18-21, AIAA-2008-6539, 2008.

[46] A. Gelb, Applied Optimal Estimation, The M.I.T. Press, 1974.

[47] I. Miletović, D. M. Pool, O. Stroosma, Q. P. Chu, M. M. van Paassen, Using Iterated Extended Kalman Filtering for Estimation of a Hexapod Flight Simulator Motion State, in: Proceedings of the 11th PEGASUS-AIAA Student Conference, Salon de Provence, France, 2015.

[48] S. J. Julier, J. K. Uhlmann, A New Extension of the Kalman Filter to Nonlinear Systems, Proceedings of the International Symposium on Aerospace/Defence Sensing, Simulation and Controls, Orlando, Florida, Apr. 20-25 (1997) 182–193.

[49] S. J. Julier, J. K. Uhlmann, Unscented Filtering and Nonlinear Estimation, Proceedings of the IEEE 923 (2004) 401–422.

[50] W. F. Phillips, C. E. Hailey, Review of Attitude Representations Used for Aircraft Kinematics, Journal of Aircraft 38 (2001) 718–737.

[51] M. W. Soijer, Rotations as Double Reflections and Geometrical Derivation of Euler-Rodrigues Parameters, AIAA Journal of Guidance, Control and Dynamics 32 (2009) 313–318.

[52] A. Papoulis, S. U. Pillai, Probability, Random Variables and Stochastic Processes (Fourth Edition), McGraw-Hill, 2002.

[53] S. J. Julier, The Scaled Unscented Transformation, Proceedings of the American Control Conference, Anchorage, Alaska, May 8-10 (2002) 4555–4559.

[54] E. A. Wan, R. Van Der Merwe, The Unscented Kalman Filter for Non-linear Estimation, in: Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, Alberta, Canada, Oct. 1-4, 2000, pp. 153–158.

[55] E. A. Wan, R. Van Der Merwe, Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, in: Proceedings of the Workshop on Advances in Machine Learning, Montreal, Canada, Jun. 8-11, 2003.

[56] R. Van Der Merwe, Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, Ph.D. thesis, Oregon Health & Science University, 2004.

[57] T. E. Oliphant, Python for Scientific Computing, Computing in Science and Engineering 9 (2007).

[58] J. D. Hunter, Matplotlib: A 2D Graphics Environment, Computing in Science and Engineering 9 (2007).

[59] P. M. T. Zaal, D. M. Pool, Multimodal Pilot Behavior in Multi-Axis Tracking Tasks with Time-Varying Motion Cueing Gains, in: Proceedings of the AIAA Modeling and Simulation Technologies Conference, National Harbor, Maryland, Jan. 13-17, AIAA-2014-0810, 2014.

[60] F. Ferrari, D. Marioli, A. Flammini, A. Taroni, Magnetostrictive Position Sensor, Technical Report, European Patent Office, 2006.

[61] B. D. O. Anderson, J. B. Moore, Optimal Filtering, Prentice Hall Information and System Sciences Series, 1979.

[62] H. M. Heerspink, W. R. Berkouwer, O. Stroosma, M. M. van Paassen, M. Mulder, J. A. Mulder, Evaluation of Vestibular Thresholds for Motion Detection in the SIMONA Research Simulator, in: Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, August 15–18, San Francisco (CA), AIAA-2005-6502, 2005.

[63] M. M. Van Paassen, O. Stroosma, J. Delatour, DUECA - Data-Driven Activation In Distributed Real-Time Computation, in: Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Denver, Colorado, Aug. 14-17, AIAA-2000-4503, AIAA, 2000.

[64] J. Benoît, G. Guennebaud, Eigen C++ Template Library for Linear Algebra, Accessed: Apr. 14, 2016. `http://eigen.tuxfamily.org/`.

[65] I. Y. Bar-Itzhack, Y. Vitek, The Enigma of False Bias Detection in a Strapdown System During Transfer Alignment, Journal of Guidance, Control, and Dynamics 8 (1985) 175–180.

[66] H. K. Lee, J. G. Lee, G. I. Jee, Calibration of Time Synchronisation Error in GPS/SDINS Hybrid Navigation, in: Proceedings of the IFAC Symposium on Automatic Control in Aerospace, Bologna/Forli, Italy, Sep. 2-7, 2001.

[67] J. Fox, Synchronisation of Input and Output Measurements using a Kalman Filter, Proceedings of the IASTED International Conference on Modeling, Indentification, and Control, Lanzarote, Spain, Feb. 6-8 (2006) 435–440.

[68] S. Butterworth, On the Theory of Filter Amplifiers, Experimental Wireless and The Wireless Engineer 7 (1930) 536–541.

[69] S. Parija, S. Hasan, A Modified Adaptive Filtering Algorithm for Identification of a Dynamic System, in: Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on, 2015, pp. 1–6.

[70] Y. Meng, S. Gao, Y. Zhong, G. Hu, A. Subic, Covariance Matching Based Adaptive Unscented Kalman Filter for Direct Filtering in INS/GNSS Integration, Acta Astronautica 120 (2016) 171–181.

[71] S. Ovaska, S. Vliviita, Angular Acceleration Measurement: A Review, IEEE Transactions on Instrumentation and Measurement 47 (1998) 1211–1217.