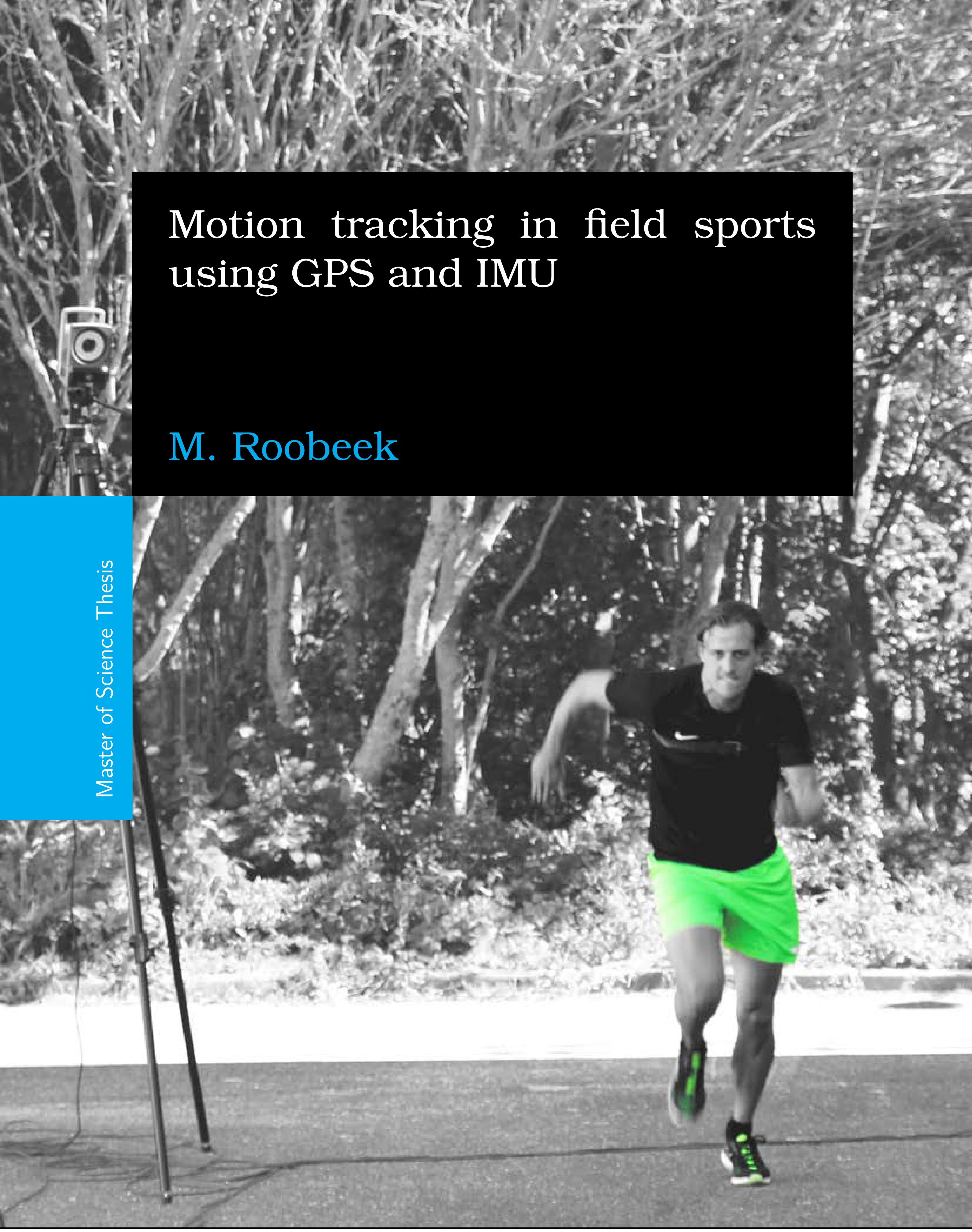


Motion tracking in field sports using GPS and IMU

M. Roobeek

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Motion tracking in field sports using GPS and IMU

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems & Control at Delft University of Technology

M. Roobeek

February 16, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was conducted on behalf of, and carried out in cooperation with JOHAN Sports. Their cooperation is hereby gratefully acknowledged.

Abstract

Injuries are a sportsman's worst nightmare: They withhold players from playing, they weaken teams and they force clubs to buy a bench-full of substitutes. On top of that, injuries can cause an awful lot of pain.

Regulating match and training intensity can drastically decrease the risk of getting injured. Furthermore, it can optimally prepare players for matchday. To enable player load regulation, accurate measurements of player positions, velocities and especially *accelerations* are required. To measure these quantities, JOHAN Sports develops a sports player motion tracking system.

The device that is used for motion tracking contains a 9-DoF MEMS Inertial Measurement Unit (IMU) and a GPS receiver. These low-cost MEMS sensors are combined via sensor fusion. The challenge in this filtering problem lies in the limited and low-quality sensor measurements combined with the high-dynamic player motion consisting of rapid orientation changes and sensed impacts in every step. Being able to overcome these challenges paves the way for injury prevention, saving sports clubs, teams and players a lot of misery.

To estimate player motion from the measurements by the tracking device, four sensor fusion algorithms are developed. For estimating rotational motion, the quaternion-based Unscented Kalman filter as described by Kraft and the Madgwick filter are devised. For estimating translational motion, a traditional linear Kalman filter and an Unscented Kalman filter are designed. These filters are combined to solve the sensor fusion problem.

On simulated data, it is shown that the Madgwick filter outperforms Kraft's quaternion-based unscented Kalman filter in both estimation accuracy and computational load. In estimating translational motion, the simulations show that the UKF and the linear Kalman filter achieve similar estimation accuracy.

Subsequently the filters are tested on real data in different experiments. It is shown that, due to the on-chip filtering operation of the GPS sensor, position estimates do not benefit from sensor fusion. Furthermore, it is shown that the filters in their current state are unable to accurately estimate player accelerations from the sensors in the JOHAN Sports tracking device.

Keywords: Motion tracking, IMU, GPS, GNSS, magnetometer, inertial navigation, sensor fusion, Unscented Kalman filter, Madgwick filter.

Table of Contents

Acknowledgements

Throughout the course of the thesis project, several people have been of great assistance. I hereby would like to take the opportunity to express my gratitude.

A big thank you to JOHAN Sports for providing the opportunity to perform this research on their behalf. The discussions we had over lunch or during carpool rides from Amsterdam to Noordwijk were often amusing, sometimes useful and always colorful. The after-lunch games of 'tens' provided me with the perfect amount of relaxation and distraction during the research. Thank you Jelle Reichert, Hidde van der Eijk, Niels van der Linden, Antonis Kalou and Mátyás Fodor and especially Robin van Kappel, for his support on the process and reporting of the research.

I would also like to thank the European Space Agency (ESA), for making the VICON system available for use. Thank you Eloise Matheson for the assistance and advice prior to and during the VICON experiment. Many thanks to Jesus Gil Fernandez for the help in designing and tuning the UKF. The support sessions we had were intense, high-paced, but very helpful. They have led to a deeper understanding of filtering techniques and the UKF in particular. Thanks to Thomas Bouvet for his efforts in arranging the support provided by ESA employees.

I would like to thank Professor Robert Babŭska for the guidance, patience and knowledge-sharing throughout the thesis project. The meetings were always pleasant and always fruitful. They kept me on track and helped me to complete the thesis that is lying in front of you.

Thanks to Joris Sijs (PhD) for the support in early filter design. The assistance has led to valuable insights and is very much appreciated.

Last but not least, I would like to thank my family for their unconditional love, support and encouragement.

Delft, University of Technology M. Roobeek
February 16, 2017

“Je gaat het pas zien als je het door hebt.”
— *H. J. Cruijff*

Chapter 1

Introduction

1-1 Motivation

In sports, injuries are highly undesirable. The joy of playing is decreased by the pain and the inability to play. Team performance suffers from the absence of injured players, fans miss out on watching their favorite player and clubs lose money on operations and wasted salary. In case of injuries, there are no winners. Nevertheless, injuries occur very often. During the 2009-2010 season, 62.7% of the soccer players in the Dutch Eredivisie suffered from one or more injuries [1].

Luckily, the likelihood and the severity of injuries can be reduced by properly regulating players' physical load [2]. To reduce injury risk and to improve performance, the company JOHAN Sports is developing a monitoring system for outdoor field sports. Sports player motion is measured via a wearable tracking device. By combining motion data with sports science models, the company is able to provide sports coaches and physiotherapists with advice on how to adapt training strategies for each individual player, reducing injury risk and optimizing performance.

For reliable training advice and effective injury prevention, accurate motion estimates are required in terms of acceleration, velocity and position with respect to the playing field. The JOHAN Sports tracking device contains microelectricomechanical systems (MEMS) sensors measuring specific forces, angular velocities, magnetic field strength and the sensor's location on Earth. The challenge in the thesis is to accurately estimate the player state with low-grade sensors.

1-2 Background

Sports players are being monitored through a tracking device, see Figure 1-1a. The devices come in a suitcase that serves as a charging station and is used to upload the measurement data to the JOHAN servers. During training sessions and matches, the tracking device is worn

(a) The JOHAN Sports tracker and suitcase.

(b) The JOHAN Sports vest.

Figure 1-1: The JOHAN Sports tracking device comes in a suitcase that serves as a charging station. It is worn in a vest on players' upper back, causing the least discomfort while playing. Figures by courtesy of JOHAN Sports.

in a vest on player's upper back, as depicted in Figure 1-1b. The tracking device contains sensors with which player motion is to be estimated. The available sensors are a GPS receiver, an accelerometer, a gyroscope and a magnetometer. Some properties of these sensors that are of influence on player state estimation are briefly discussed.

GPS The GPS receiver uses sattelites orbiting the Earth to measure the tracking device's location on Earth. From the measured locations, an on-chip filter derives a directionless speed estimate. Due to the filtering operation, the error on position estimates is not Zero-Mean White Noise (ZMWN).

IMU The 3-axis accelerometer measures accelerations in the tracking device coordinate frame. It measures both the gravitional acceleration and the player acceleration. An accurate estimate of the tracking device 3D orientation is required to distinguish between the part of the total measured acceleration that can be attributed to gravity and the part that is caused by translational motion. []

The angular velocity of the tracking device is measured by a 3-axis gyroscope. An ideal gyroscope would be able to take the initial orientation and determine each next orientation by integrating the angular velocity over time. This process is called gyro dead-reckoning. Real gyroscopes suffer from biases and noise, causing cumulative errors. []

A 3-axis magnetometer measures the flux of the magnetic fields surrounding it, in the tracking device coordinate system. Ideally it would measure only the Earth's magnetic field, resulting in a vector pointing towards the magnetic North. In practice, multiple magnetic fields exist around the sensor, disturbing the measurements. []

Applications The gyroscope, accelerometer and magnetometer combined are called a 9-Degrees of Freedom (DoF) Inertial Measurement Unit (IMU). These sensors are commonly used together and are often contained on the same chip. In applications like quadcopters, submarine drones and virtual reality headsets, 9-DoF IMU's are used to estimate the device orientation. The gyroscope is then used to measure the change in orientation, the accelerometer and magnetometer are used as a ‘gravity sensor’ and ‘Earth magnetic field sensor’, respectively.

In the previously described applications, sensor fusion is applied to estimate the orientation, position and sometimes velocity. In the application of sports player state estimation, determining the player *acceleration* is of great importance, since player acceleration and deceleration is found to be very indicative for the load on muscles [3].

Filtering techniques The rotational motion of tracking devices causes the system to be nonlinear. Probably the most widely applied estimation technique used for nonlinear applications is the Extended Kalman filter (EKF) [4]. In the EKF, the nonlinear models are linearized, enabling the use of the linear Kalman filter equations. A new extension to nonlinear Kalman filters was developed by Van der Merwe [5] and Julier [6], known as the Unscented Kalman filter (UKF). In the UKF, a set of cleverly chosen points around the state estimate is propagated through the nonlinear models. From the transformed points, the new state estimate is determined in similar fashion as in the traditional Kalman filter equations. The UKF has been found to outperform the EKF in state estimation for highly nonlinear systems [7], [8].

Another technique that is commonly used in orientation estimation with IMU’s is the Madgwick filter [9]. This filter uses one-step gradient descent optimization to find the orientation that best fits the gyroscope, accelerometer and magnetomter measurements.

The Unscented Kalman filter can be designed to estimate the full state at once. However, it is also possible to split the state estimation in a rotational and a translational part. The possible advantage is that the accelerometer can be used as a ‘gravity sensor’ in the orientation estimation and as an ‘acceleration sensor’ in estimating the translational part of the state vector.

The rotational state can then be either estimated with a UKF referred to as the Kraft filter, or with a Madgwick filter. The model of translational motion can simply be linear, making it possible to use a linear Kalman filter. By including the directionless speed estimate provided by the GPS receiver as a measurement in the model, the translational model also becomes nonlinear. A UKF can be used for this part of the state vector.

1-3 Problem statement

The reliablity of the training advice that JOHAN Sports provides, relies on accurate player motion estimates in terms of position, velocity and acceleration. These estimates are to be obtained from measurements with the JOHAN Sports tracking device.

Noise on position not ZMWN The position is measured via a GPS module. Although the position estimates do not drift over time, they do contain offsets in the short term. Furthermore, the noise on the position estimates is not zero-mean and not white due the filtering action that is done on the chip, making it not straightforward to be filtered out.

Estimating velocity: noise amplification vs. cumulative errors The velocity can be estimated by either taking the derivative of the position estimates, or by integrating the accelerations. When deriving the velocity from position estimates, sudden jumps in the position caused by measurement noise result in problematic peaks in the estimated velocity. This is referred to as noise amplification. On the other hand, integrating the accelerations, referred to as dead-reckoning, is subject to cumulative errors.

Estimating orientation: In-plane accelerations contaminate gravity measurements The orientation of a *non-moving* rigid 3D body is fully determined by the vectors describing the Earth's gravitational field and the Earth's magnetic field, measured by the accelerometer and magnetometer, respectively. Furthermore, it can be determined by integrating the angular velocity (gyro dead-reckoning), provided that the previous orientation is known. However, the object whose orientation is to be estimated tends to be *moving*, contaminating the gravity measurements. And as stated before, dead-reckoning is subject to cumulative errors.

High signal-to-noise ratio in estimating accelerations In most applications, the accelerometer is primarily used to determine the direction of the gravity vector, which in turn is used in the orientation estimate. In the sports player motion tracking problem, player accelerations are a desired estimated quantity. Note that player accelerations vary roughly between -3 and 3 $\mathrm{m/s^2}$, while gravity is in the order of ≈ 10 $\mathrm{m/s^2}$. This results in a small signal to noise ratio, making it difficult to retrieve the desired measured signal.

1-4 Research question

The problems described in section 1-3 give rise to the following research question:

> Which sensor fusion algorithm is most suitable for sports player motion estimation from a 9-DoF IMU and GPS receiver?

The emphasis will lie on the estimation of accelerations, since these are most indicative for a players' physical load.

1-5 Hypothesis

It is expected that estimating the rotational and translational part of the state vector separately is beneficial to the estimation accuracy. Furthermore, it is expected that the Kraft filter is able to estimate the orientation of a 3D rigid body in space more accurately than the Madgwick filter. Lastly, it is expected that a UKF, by taking into account speed estimates from the GPS receiver as a measurement, outperforms a linear Kalman filter in estimating translational motion.

1-6 Approach

The approach to investigate the best suitable sensor fusion algorithm for sports player motion tracking is described in this section.

Five different filters are created. The *full UKF* estimates the full state at once. For estimating rotational motion, the *Kraft* filter, a quaternion-based unscented Kalman filter and the *Madwgick* filter are compared. For estimating translational motion, the *UKF* and linear *Kalman filter* are compared.

The filters are combined in different configurations:

- Full UKF
- Kraft + UKF
- Madgwick + UKF
- Kraft + KF
- Madgwick + KF

Using simulated data, the filters are developed. Weak spots and difficulties of each filter are identified. The filters that perform well in simulation are applied in real world experiments.

The real world experiments are designed in such a way that the estimation accuracy of the full state vector is assessed.

Based on the experimental results, the best suitable filter is chosen out of the above-described configurations.

Chapter 2

Methods and materials

When working with rotations in a 3 dimensional space, coordinate frame conventions are important. The conventions around the use of the different coordinate frames in the thesis are clarified in section 2-1. To be clear on the equipment that is used in the research, the type of sensors that are used are clarified in Section 2-2. A short description of their working principle is given. Throughout the thesis, a number of filters is used to estimate sports player motion from sensor measurements. In Section 2-3, the filters that the reader is expected not to be familiar with are explained in some depth. Section 2-4 describes the different models used in the (unscented) Kalman filters. In Section 2-5, the method to determine the optimal filter configuration is described.

2-1 Coordinate systems

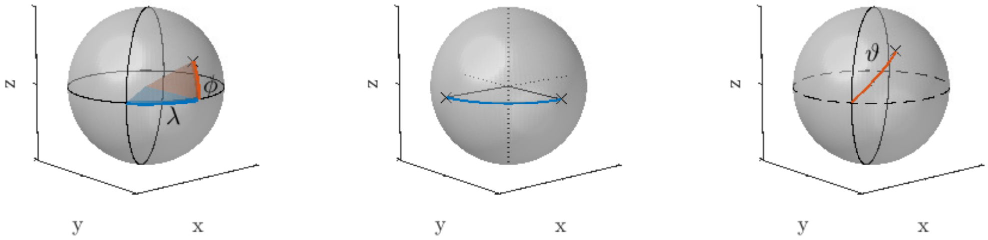
The same motion can be described in different coordinate frames. In the application of field sports motion tracking, it is first and foremost desired to know how players move with respect to the playing field. However, the GPS sensor measures locations with respect to the center of the Earth and the IMU sensor measures rotations, accelerations and the direction of the magnetic north with respect to the tracking device.

The sensor measurements are transformed to the field coordinate frame via coordinate transformations. When transforming from one coordinate frame to the other, it is important to be aware of how each coordinate frame is defined. In this section all different coordinate frames are defined.

Note on notation To denote the relative frames of orientations and vectors, the notation of leading subscripts and superscripts adopted from Craig [10] is used. For example, the rotation matrix describing the orientation of the body frame Ψ_B in the world frame Ψ_W is denoted as ${}^W_B R$ and a vector v described with respect to the body frame Ψ_B is written as ${}^B v$. In Appendix A, commonly used attitude representations are described.

2-1-1 Earth coordinate frame

The Earth coordinate frame (Ψ_E) is an Earth-Centered Earth-Fixed (ECEF) coordinate system, meaning that it's center lies at the center of the Earth and that it rotates with the Earth. The Earth frame describes the location of a point on Earth in degrees latitude (ϕ) and longitude (λ). For any given location on the Earth's surface, the latitude describes the vertical angle between that point and the equatorial plane, where North is the positive direction. The longitude describes the horizontal angle between the point and the reference meridian crossing through Greenwich (UK), where East is the positive direction. The Earth frame is depicted in Figure 2-1a.



(a) Latitude and longitude **(b)** Great circle distance. **(c)** Bearing.

Figure 2-1: Earth coordinate frame. For a point on the Earth's surface, the latitude ϕ describes the angle between the equator and the point. The longitude λ describes the angle between the prime meridian and the point. The great circle distance is the length of an arc between two points, measured over the surface of a sphere. The bearing is the angle between a line and the meridian crossing that line.

2-1-2 World coordinate frame

The world coordinate frame Ψ_W is fixed on the Earth's surface. It has its origin at the location of the first GPS measurement in the dataset. The x-axis points towards the magnetic North, the y-axis points towards the 'magnetic west', parallel to the Earth's surface and orthogonal to x. The z-axis points upwards. Positions in the world frame are defined in meters. The coordinate frame is depicted in Figure 2-2.

Transforming Earth frame to world frame

The GPS sensor measures locations in degrees latitude and longitude in the Earth frame. To transform these locations into (x, y)-positions in the world frame, the Haversine formula [11] is used. Assume point $p_0 = (\phi_0, \lambda_0)$ and point $p_1 = (\phi_1, \lambda_1)$. The Haversine formula calculates the great cirlce distance and bearing between the two points on a sphere.

The great circle distance d is the distance between the points, measured over the sphere

(a) World coordinate frame, global view. **(b)** World coordinate frame, local view.

Figure 2-2: World coordinate frame. The x-axis points towards the magnetic North, the origin lies in the first measured location in the dataset.

surface, as displayed in Figure 2-1b. It is calculated via Equation 2-3.

$$c_1 = \sin^2(\frac{\Delta\phi}{2}) + \cos(\phi_0) \cdot \cos(\phi_1) \cdot \sin^2(\frac{\Delta\lambda}{2}) \tag{2-1}$$

$$c_2 = 2 \cdot \text{atan2}(\sqrt{c_1},\ \sqrt{1 - c_1}) \tag{2-2}$$

$$d = r \cdot c_2, \tag{2-3}$$

where r is the radius of the sphere.

The bearing θ is the angle between the line through p_0 and p_1 and the meridian crossing through p_0, see Figure 2-1c. It is calculated via Equation 2-4.

$$\theta = \text{atan2}\left(\sin(\Delta\lambda)\cos(\phi_1),\ \cos(\phi_0)\sin(\phi_1) - \sin(\phi_0)\cos(\phi_1)\cos(\Delta\lambda)\right) \tag{2-4}$$

With the distance and bearing of all points with respect to the origin, the $(x,\ y)$-position can be determined at all measured time instances.

2-1-3 Field coordinate frame

Ultimately, the coordinate frame that matters most is the field coordinate frame. The field coordinate frame, denoted Ψ_F, has its origin at the center point of the playing field. The x-axis points towards the North-most goal. Looking in x direction, the y-axis points to the left, running over the center line of the field. The z-axis points upwards. The coordinate frame is depicted in Figure 2-3b. Positions on the field are measured in meters. The centerpoint and dimensions of the field are determined from the GPS locations of the corner points.

Transforming world frame to field frame

The transformation from world frame to field frame is a fixed rotation and translation. First, the data is rotated over the angle between the field x-axis and the magnetic North. Subsequently, the translation is calculated via the Earth frame. The position of the world frame origin with respect to the field frame is determined. The data is than translated to the field frame.

$$^{F}x = {}^{F}_{W}Rx + p \tag{2-5}$$

(a) Field coordinate frame, global view. **(b)** Field coordinate frame, local view.

Figure 2-3: The field coordinate frame. The origin lies at the center of the field. The x-axis points towards the North-most goal, the y-axis runs to the left, over the centerline and the z-axis points upwards.

where R is a rotation matrix over the z-axis and p is the position shift between the field origin and the world frame origin.

2-1-4 Body coordinate frame

The body coordinate frame is denoted Ψ_B. The body frame is defined in such a way that if a player stands up straight, the x-axis points forwards out of his chest, the y-axis points left and the z-axis points upwards. The origin of the of the body frame lies at the center of the tracking device.

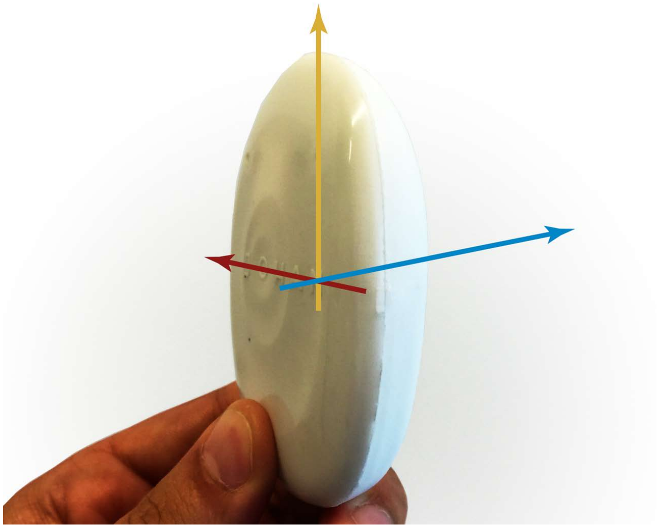


Figure 2-4: The body coordinate frame Ψ_B is defined such that if the player stands up straight, the x-axis (blue) points forwards, the y-axis (red) points to the left and the z-axis (yellow) points upwards.

2-1-5 Sensor coordinate frames

The sensors are mounted on the circuit board of the tracking device. The coordinate frames as defined by the chip manufacturers do not coincide with the tracking device body coordinate frame. For the GPS receiver, this obviously does not matter. For the other sensors it does.

The difference in orientation between the IMU coordinate frame (Ψ_I) and the body coordinate frame is a rotation of 90 degrees over the y-axis.

$$^{B}v = {}^{B}_{I}R\ ^{I}v = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} {}^{I}v \tag{2-6}$$

The coordinate frame of the magnetometer (Ψ_M) is defined as a left-handed coordinate frame. It is transformed to the (right-handed) body coordinate frame via Equation 2-7.

$$^{B}v = {}^{B}_{M}R\ ^{M}v = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} {}^{M}v \tag{2-7}$$

These fixed coordinate transformations are performed as a pre-processing step.

2-2 Sensors

The sensors that the JOHAN Sports tracking device contains are an accelerometer, a gyroscope, a magnetometer and a Global Positioning System (GPS) receiver. The sensors and their working principles are described in this section.

(a) Invensense MPU-6050 IMU module.

(b) NXP MAG3110 magnetometer.

(c) u-blox PAM 7Q GPS receiver.

Figure 2-5: The sensors that are contained in the JOHAN Sports tracking device.

The 3-axis accelerometer and 3-axis gyroscope are contained on the same chip (the InvenSense MPU-6050[1]), displayed in Figure 2-5a. Together with the 3-axis magnetometer (NXP MAG-3110[2], Figure 2-5b), this sensor is viewed as a 9-DoF IMU. The GPS receiver (u-blox PAM 7Q[3]) is displayed in Figure 2-5c.

---

[1] www.invensense.com/

[2] www.nxp.com/

[3] www.u-blox.com/

2-2-1 Inertial Measurement Unit (IMU)

The Inertial Measurement Unit chip in the tracking device houses both the accelerometer and the gyroscope. Their working principle and calibration technique are described in this section.

Accelerometer

The accelerometer measures accelerations in each of the three axes. In free fall, an ideal accelerometer would measure zero acceleration in all axes. If the accelerometer is kept in a fixed upright position on Earth, it senses the normal force counteracting gravity, resulting in a measured acceleration in positive z-direction.

The MEMS accelerometer in the tracking device is a piezoelectrical accelerometer. It consists of a proof mass connected to a piezoelectric element. A force on the sensor causes a displacement of the proof mass, leading to deformation of the piezoelectric element, which in turn generates a voltage. The voltage is directly proportional to the sensed force [12].

MEMS accelerometers suffer from multiple sources of errors, the most important being scale factors, coupling factors due to axis misalignment and biases.

Gyroscope

The gyroscope uses the Coriolis effect to measure angular velocities ($[^\circ/s]$) around each of the three axes. A proof mass is resonating in one direction. If the sensor is rotated, the Coriolis effect will generate a force in perpendicular direction. This causes a piezoelectric element to deform, generating a voltage. This voltage is proportional to the rotational rate [12].

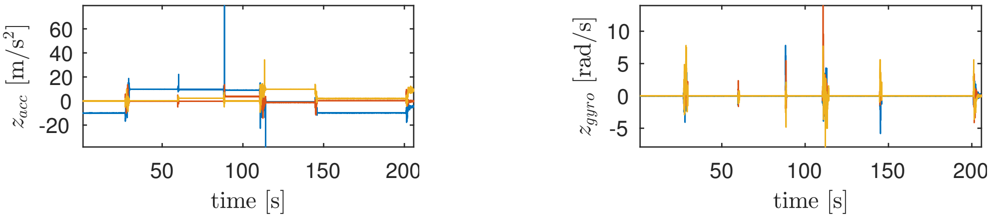
The gyroscope is in strapdown configuration, meaning that it moves with the object that it is attached to. MEMS gyroscopes suffer from the similar sources of errors as accelerometers; scale factors, coupling factors and biases.

Calibration

When arriving from the factory, low-cost MEMS IMU's are usually poorly calibrated. A factory calibration has been performed, but the resulting measurements often still contain non-negligable systematic errors [13]. To compensate for these systematic errors, calibration is required.

To calibrate the accelerometer and gyroscope, usually a mechanical platform rotating the IMU into different precisely controlled orientations and angular rates is used [14]. The measured rotational velocity and measured accelerations can then be compared to the known motion and an optimization problem can be solved to determine the biases, coupling factors and scale factors. Since such a platform was not available, a simpler method described by is chosen.

When an IMU is lying still, the accelerometer is supposed to only measure gravity and the gyroscope is expected to measure zero rotational velocity. For calibration, the tracking device is held still in different orientations during different time intervals, see Figure 2-6.



(a) Raw accelerometer data. **(b)** Raw gyroscope data.

Figure 2-6: Raw data for IMU calibration. Note that the data consists of intervals where the tracking device is kept in a fixed orientation with transitions in between. The calibration is performed on the first 'still' interval.

To quantify the error before calibration, the RMSE over the norm of the sensor measurements is calculated. For the accelerometer, the norm should be 1g, or 9.81 m/s^2. The norm of the gyroscope measurements should be 0. The root-mean-square errors of the norm are calculated via Equation 2-8.

$$\text{RMSE}_{acc} = \sqrt{\frac{\sum_{i=1}^{n}(|a| - g)^2}{n}}, \qquad \text{RMSE}_{gyro} = \sqrt{\frac{\sum_{i=1}^{n}|\omega|^2}{n}}, \tag{2-8}$$

In the first time interval, the tracker is put on a table, with the x-axis pointing upwards. Now, by subtracting the gravity vector from the original accelerometer measurements, the expected accelerometer measurements become zero in all axes. The expected gyroscope measurements are zero as well, since no rotational motion is present.

The biases of both sensors are determined by taking the mean of the gravity-corrected accelerometer measurements and the gyroscope measurements over the first time interval. The other time intervals are used to quantify the accuracy improvement.

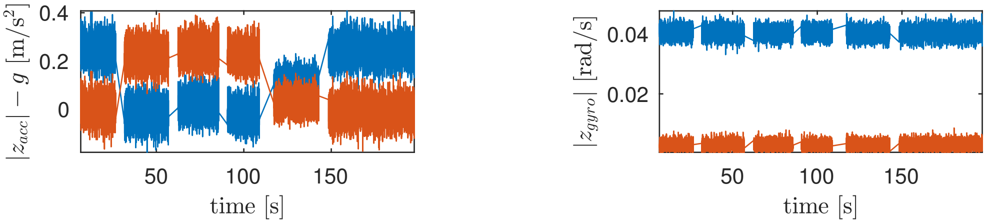
$$a_{bias} = \frac{\sum_{i=1}^{n} a_i - Rg}{n}, \qquad \omega_{bias} = \frac{\sum_{i=1}^{n} \omega_i}{n}, \tag{2-9}$$

where a_{bias} and ω_{bias} are the accelerometer bias and gyroscope bias, respectively, R is the rotation matrix correcting for the orientation of the tracking device, g is the gravity vector and n the number of still samples. The biases of the accelerometer and gyroscope are presented in Table 2-1.

Biases	Accelerometer			Gyroscope		
Tracking device 1	−0.25	−0.05	0.06	−0.02	0.03	−0.00
Tracking device 2	−0.10	−0.11	0.11	−0.02	0.00	0.02

Table 2-1: Accelerometer and gyroscope bias for two tracking devices.

To calibrate the accelerometer and gyroscope, their respective biases are subtracted from the raw measurements. To quantify the improvement, the RMSEs of the norm are calculated again after calibration. The resulting RMSEs are presented in Table 2-2. The deviation of the expected norm before and after calibration is depicted in Figure 2-7 for both the accelerometer and the gyroscope.



(a) Accelerometer norm error before (blue) and after (red) calibration.

(b) Gyroscope norm error before (blue) and after (red) calibration.

Figure 2-7: The deviation of the expected norm for both the accelerometer and the gyroscope measurements before (blue) and after calibration (red). Note that the improvement in accelerometer accuracy is minor, while the improvement in gyroscope accuracy is large. The data that is displayed stems from tracking device 1.

RMSE	Accelerometer			Gyroscope		
	raw	calibrated	improvement	raw	calibrated	improvement
Tracking device 1	0.170	0.157	7.42%	0.040	0.003	92.45%
Tracking device 2	0.122	0.103	16.05%	0.037	0.006	83.00%

Table 2-2: The RMSEs of the raw and calibrated accelerometer and gyroscope measurements. The numbers show that the improvement in accelerometer accuracy is in the order of 10%. The gyroscope accuracy improves with ±90% via calibration.

As Figure 2-7 and Table 2-2 show, the accelerometer measurements and especially the gyroscope measurements improve by applying this simple calibration technique. To calibrate the sensors to higher accuracy, more involved calibration techniques are required.

2-2-2 Magnetometer

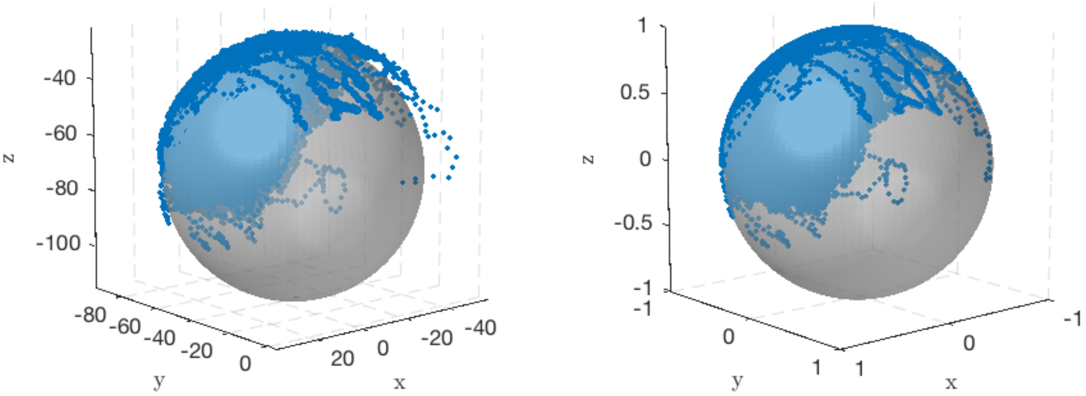
The 3-axis magnetometer measures the flux Φ in μT of the Earth's magnetic field, serving as a 3D compass. The chip works on the Magnetic Tunnel Junction (MTJ) principle. In an MTJ there are two ferromagnets separated by a thin insulating layer, about 1 nm thick. The magnetization and resistance change as a function of magnetic field [15].

The magnetometer does not perfectly measure only the Earth's magnetic field. The measurements are distorted by external magnetic objects. These distortians are divided into hard-iron distortions and soft-iron distortions. Hard-iron distortions are caused by magnetic sources that are fixed in the body coordinate frame. They rotate with the sensor, so assuming that their magnetic field strength remains constant, they cause a fixed offset in the measurements. Soft-iron distortions are distortions that cause a deflection or alteration in the existing magnetic field, for example the presence of metals in the soil or magnetic sources in buildings. These errors stretch or deform the magnetic field, since they alter the magnetic field strength.

To compensate for hard- and soft iron errors, the magnetometer data is calibrated before it enters the filters. This action yields the measurements dimensionless. A more detailed description of the magnetometer pre-processing is found in Section 2-2-2.

Calibration

For an ideal magnetometer with no magnetic sources around, each measurement of the magnetometer would lie on a sphere centered around $(0,0,0)$ with the Earth's magnetic field strength as a radius. The magnetometer in the JOHAN Sports tracking device is not ideal. In Figure 2-8a, the raw magnetometer measurements of a soccer match are depicted. The figure shows that the data points are not centered around the origin and that not all measurements lie on a perfect sphere.



(a) Raw magnetometer measurements. **(b)** Calibrated magnetometer measurements.

Figure 2-8: Raw and calibrated magnetometer measurements. The raw magnetometer measurements are not centered around $(0,0,0)$ due to hard-iron errors. Due to soft-iron errors, not all measurements lie on a sphere. By removing the offset and projecting the measurements on the unit sphere, the hard-iron errors and soft-iron errors are removed.

To determine the hard-iron and soft-iron errors in the magnetometer and account for them, a sphere is fitted through the complete set of measurements. Fitting a sphere through data is done by solving the optimization problem defined in Equation 2-10. In Figure 2-8b, the fit of the sphere through the measurements is displayed.

$$\underset{c,\ r}{\text{minimize}}\ f(c,r,m),\ \text{with} \tag{2-10}$$

$$f(c,r,m) = \sum_{i=1}^{n}\left(\sqrt{(m_{x,i}-c_x)^2+(m_{y,i}-c_y)^2+(m_{z,i}-c_z)^2}-r\right)^2 \tag{2-11}$$

where $c \in \mathbb{R}^3$ represents the center of the sphere, $r \in \mathbb{R}$ is the radius of the sphere, corresponding to the Earth's magnetic field strength and $m \in \mathbb{R}^3$ is the magnetometer measurement.

The center c of the fitted sphere represents the hard-iron error. With this bias removed, the magnetometer measurements are now centered around $(0,0,0)$. However, soft-iron errors still exist. These errors are created by deviations in the local magnetic field. This occurs for example when players walk into the dressing rooms, where magnetic materials that are present distort the measurements of the Earth's magnetic field. To minimize the effect of measurements inside buildings, only measurements during periods with GNSS measurements available are used for the optimization.

There are several approaches to account for soft-iron errors. The data points with deviant measured magnetic field strength can be completely rejected, they can be weighed less heavily in the filter or they can be projected onto the unit sphere. Projecting each measurement onto the sphere results in the adjusted magnetometer measurements depicted in Figure 2-8b. Each of these data points now represent the estimated direction of the magnetic north at time t in tracker coordinate frame.

2-2-3 Global Positioning System (GPS)

GNSS is the collective name for all satellite-based navigation systems, of which the American Global Positioning System (GPS) is most commonly known. GPSs consists of multiple satellites, emitting identifiable signals: pseudo ranges. By measuring the travel time of these signals to the receiver, the distance between the satellite and the receiver can be calculated, using the Doppler effect [16]. The working principle is displayed in Figure 2-9. The accuracy of a GPS can be affected by deviations in the satellite orbit, disturbances in the atmosphere (e.g. clouds) or signal jamming, where the signal is blocked by for example tall buildings.

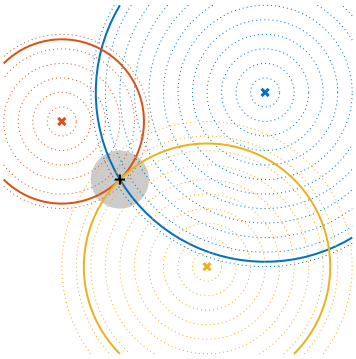


Figure 2-9: Working principle of Global Positioning System. Satellites transmit a package of information with their position and a timestamp. From the traveling time of the information, the position of the GPS receiver is determined. In a 2D space, three satellites fully determine the location of the receiver. In a 3D space, three satellites limit the possible location of the receiver to two points, one of which lies close to Earth. A fourth satellite is used to correct for clock error.

The u-blox PAM 7Q module filters the measurements on-chip with a dual SAW (surface acoustic wave) filter and outputs latitudes, longitudes, speed and time measurements at 10 Hz. The filter has two purposes. It enables users with limited knowledge on the functioning of GPS to use the module without having to do calculations with pseudo ranges and it increases the filters immunity to signal jamming. The downside of the filtering action is that the noise on the position estimates becomes non-zero-mean and non-white.

The GPS receiver supports Satellite-Based Augmentation System (SBAS)s [17]. These systems increase GPS accuracy by involving external information into the calculation process. The European version of a Satellite-Based Augmentation System (SBAS) is called European Geostationary Navigation Overlay Service (EGNOS). The EGNOS system consists of four geostationary satellites and a network of ground stations. EGNOS improves position estimates to about 1 meter [18].

2-3 Filters

Probably the best known state estimator is the Kalman filter [19]. For nonlinear applications, extensions to the Kalman filter are developed. The Extended Kalman filter (EKF) [4], [20] is such a nonlinear extension that is widely used in industry. A less common nonlinear extension to the Kalman filter is the Unscented Kalman filter (UKF). An explanation of its working principle is given in Section 2-3-1.

A filter that is often applied for IMU-based orientation estimation by hobbyists is the Madgwick filter. An algorithm that estimates only the 3D orientation of a rigid body in space is referred to as an Attitude and Heading Reference System (AHRS). The Madgwick filter is popular for its ease of implementation and satisfactory results. In Section 2-3-2, a description is given.

2-3-1 Unscented Kalman filter (UKF)

The Unscented Kalman filter is a nonlinear version of the well-known Kalman filter. Like the standard Kalman filter, the UKF uses a propagation model f that predicts the next state estimate, defined by Equation 2-12. The observation model h transforms the state estimate into the expected measurements via Equation 2-13.

$$x_{k+1} = f(x_k, u_k, w_k) \tag{2-12}$$

$$y_k = h(x_k, u_k, v_k), \tag{2-13}$$

where x and u denote the state vector and external inputs and w and v represent the process- and measurement noise, respectively.

Unscented transformation

The Unscented Kalman filter uses the unscented transformation to be able to deal with nonlinearity in models. In the unscented transformation, a set of so-called *sigma points* is calculated in such a way that they lie around the current state estimate distributed according to the current covariance matrix. The sigma points are propagated through a nonlinear model, resulting in transformed sigma points. From the transformed sigma points, the new covariance matrix and new mean are calculated. The unscented transformation is depicted in Figure 2-10. A numerical example is presented in Appendix C.

Update equations

The Unscented Kalman filter uses the unscented transformation in the prediction step and the measurement update.

In the prediction step, assume the previous state estimate and covariance matrix to be x_{k-1} and P_{k-1}, respectively. The sigma points $\mathcal{X}_k$ are calculated via Equation 2-14, such that the mean of the sigma points equals x_{k-1} and such that the covariance matrix of the sigma points equals P_{k-1}:

$$\mathcal{X}_k = \left[x_{k-1} - \zeta\sqrt{P_{k-1}}, \quad x_{k-1}, \quad x_{k-1} + \zeta\sqrt{P_{k-1}} \right] \tag{2-14}$$

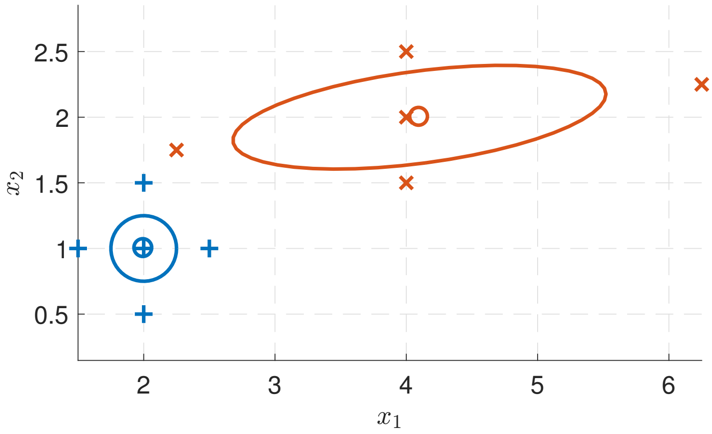


Figure 2-10: Unscented transformation. Around the mean $(2,\ 1)$, sigma points (blue plus signs) are chosen according to the covariance matrix (blue circle). Each sigma point is individually propagated trough some nonlinear function, resulting in the transformed sigma points (red 'x' signs). From the transformed sigma points, the mean (red 'o') and covariance (red ellipse) are calculated.

where $\zeta = \sqrt{2n+1+\kappa}$ with n the state dimension and κ a tuning parameter.

Now, the unscented transformation takes place. The set of sigma points is propagated through the propagation model f, using Equation 2-15, resulting in a priori sigma points $\mathcal{X}^-$. The a priori state estimate x^- is found by taking the mean of the a priori sigma points via Equation 2-16, the a priori state covariance $P^-_{x_k}$ is calculated via Equation 2-17.

$$\mathcal{X}^-_k = f(\mathcal{X}_k, w) \tag{2-15}$$

$$x^-_k = \frac{1}{L}\sum_{i=1}^{L} \mathcal{X}^-_{i,k} \tag{2-16}$$

$$P^-_{x_k} = (\mathcal{X}^-_k - x^-_k) \cdot (\mathcal{X}^-_k - x^-_k)^T + Q \tag{2-17}$$

where L is the number of simga points and Q is the covariance matrix of process noise w. Note that in this implementation, the process noise is assumed to be additive.

By definition, the set of a priori sigma points $\mathcal{X}^-_k$ is distributed according to covariance matrix $P^-_{x_k}$ around its mean x^-_k. After all, the mean and covariance are calculated *from* the a priori sigma points. The mean however, does not necessarily coincide with a sigma point in the set. This is the case in for example Figure 2-10, where the mean (red 'o') lies slightly to the right of the middle a priori sigma point (red 'x'). Optionally, a new set of a priori sigma points can be calculated by plugging x^-_k and $P^-_{x_k}$ in Equation 2-14, reassuring that the mean is part of the set of sigma points.

The set of a priori sigma points is now used in the measurement update. The sigma points are propagated through the measurement model via Equation 2-18, yielding the measurement sigma points $\mathcal{Y}_k$. Via Equation 2-19, the expected measurement y is calculated. The measurement covariance matrix P_y is obtained via Equation 2-20.

$$\mathcal{Y}_k = h(\mathcal{X}_k^-, v) \tag{2-18}$$

$$y_k = \frac{1}{L}\sum_{i=1}^{L} \mathcal{Y}_{i,k} \tag{2-19}$$

$$P_{y_k} = (\mathcal{Y}_k - y_k) \cdot (\mathcal{Y}_k - y_k)^T + R \tag{2-20}$$

where R represents the measurement noise covariance. Again, the noise is assumed to be additive.

Subsequently, the cross-covariance matrix P_{xy} and Kalman gain K are calculated.

$$P_{x_k,y_k} = (\mathcal{X}_k^- - x_k^-) \cdot (\mathcal{Y}_k - y_k)^T \tag{2-21}$$

$$K_k = P_{x_k,y_k} P_{y_k}^{-1} \tag{2-22}$$

Finally, the new state estimate and state covariance matrix are obtained via Equation 2-23 and Equation 2-24.

$$x_k = x_k^- + K_k(z_k - y_k) \tag{2-23}$$

$$P_k = P_{k-1}^- - K_k P_{y_k} K_k{}^T \tag{2-24}$$

where z_k is the measurement at time k.

The structure of the UKF is similar to that of the linear Kalman filter. For comparison, see Appendix B. The additional steps concerning the unscented transformation make the UKF able to deal with nonlinearities.

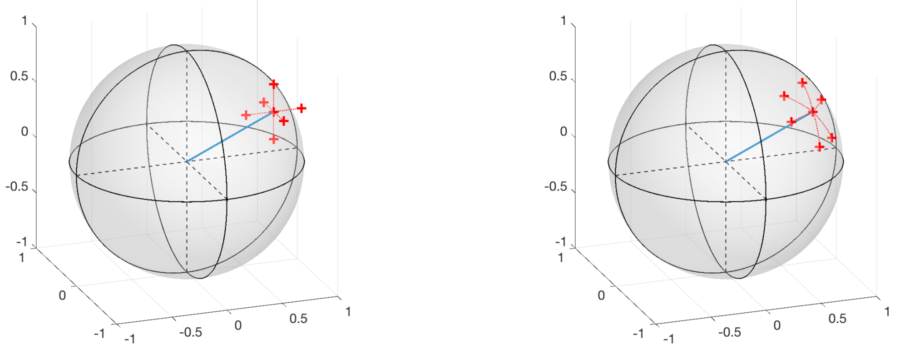
Handling quaternions in the unscented transformation

In the Unscented Kalman filter, the use of quaternions requires some extra attention. What distinguishes quaternions from states like position or angular velocity, is their unity constraint. The problem occurs in the calculation of sigma points in the UKF, as explained in [21] and [22]. The UKF assumes the state space is a vector space. However, the unity constraint deprives the quaternion of one degree of freedom. This turns the quaternion state space into a 3D unit sphere manifold instead of a 4D vector space. A graphical interpretation of this problem is given in Figure 2-11.

Since the unit quaternions $q \in \mathbb{R}^4$ have only three degrees of freedom, their corresponding uncertainty should also have dimension 3, resulting in a state covariance matrix with dimension $n - 1$. From a three-dimensional uncertainty vector w_q, a disturbed quaternion is calculated via:

$$\begin{aligned} \alpha_w &= |w_q| \\ e_w &= \frac{w_q}{|w_q|} \\ q_w &= \begin{pmatrix} \cos(\frac{\alpha_w}{2}) \\ e_w \sin(\frac{\alpha_w}{2}) \end{pmatrix} \end{aligned} \tag{2-25}$$

In calculating the sigma points, the quaternion part of the state vector for each sigma point is calculated via Equation 2-25. The other part of the state vector is calculated via the standard approach.



(a) Incorrect quaternion sigma points. Note that the 4th dimension is not displayed.

(b) Correct quaternion sigma points.

Figure 2-11: Quaternion sigma points. As is visible in Figure 2-11a, sigma points calculated by simple addition do not satisfy the unity constraint. Figure 2-11b shows that sigma points calculated via disturbance rotations as described in [21] do satisfy the unity constraint.

2-3-2 Madgwick filter

The Madgwick filter [9] is an Attitude and Heading Reference System (AHRS) algorithm. This means that it estimats the 3D orientation of a rigid body in space. The Madgwick filter creates two estimates of the orientation quaternion in every update step. One estimate is calculated from the angular velocity (gyroscope) and the previous orientation estimate. The other quaternion uses one-step gradient descent to estimate the orientation with respect to homogeneous fields (gravity and the Earth's magnetic field). Both orientation estimated are fused into one by a weighted average. Due to clever simplifications in the fusion process, the Madgwick filter has a lower computational load than the UKF.

Orientation from angular velocity

Assume the current estimated orientation quaternion to be ${}^B_W q$, describing the orientation of the world frame as viewed from the body frame. To predict the next orientation, the angular velocity is integrated and added to the current orientation.

The angular velocity is taken directly from the gyroscope measurements and is rewritten as a quaternion via Equation 2-26. In contrast to quaternions describing an orientation, the angular velocity quaternion is not subject to the unity constraint. The quaternion derivative, describing the rate at which the orientation of the world frame changes with respect to the body frame, is calculated by Equation 2-27, where the subscript ω indicates that it is derived from angular velocity. The estimate of the new quaternion is obtained via Equation 2-28.

$$ {}^B\omega = \begin{pmatrix} 0 & \omega_x & \omega_y & \omega_z \end{pmatrix}^T \tag{2-26} $$

$$ {}^B_W\dot{q}_{\omega,k} = \frac{1}{2}\,{}^B_W q_{k-1} \otimes {}^B\omega_k \tag{2-27} $$

$$ {}^B_W q_{\omega,k} = {}^B_W q_{k-1} + {}^B_W\dot{q}_{\omega,k} T, \tag{2-28} $$

where ω denotes the gyroscope measurements, q denotes the attitude quaternion and T represents the sample time.

Provided that the initial orientation is known, predictions of the next orientation via the above-described method are correct in the short term. However, due to noise on the gyroscope measurements, errors will accumulate. The cumulative errors will cause the orientation estimate to drift away from the actual orientation quaternion.

Orientation from a homogeneous field

To correct for cumulative errors that occur in orientation estimates derived from the gyroscope measurements, a second orientation estimate that is reliable over the long term is desired. Such an orientation estimate can be derived from homogeneous vector fields. A vector field is called homogeneous if its direction is constant over the complete workspace. The type of workspaces in the application of player motion tracking are sports fields. There, both gravity and the Earth's magnetic field are assumed to be homogeneous.

Consider a homogeneous field with known direction in the world frame, denoted ${}^{W}d$. The sensor measurement of that homogeneous field in the body frame is called ${}^{B}s$. The quaternion ${}^{B}_{W}q$ that describes the rotation between the two satisfies:

$$ {}^{B}s = {}^{B}_{W}q^{\star} \otimes {}^{W}d \otimes {}^{B}_{W}q \tag{2-29} $$

To find a quaternion that satisfies this equation, the optimization in Equation 2-30 problem is formulated. The objective function is defined by Equation 2-31.

$$ \min_{{}^{B}_{W}q \in \mathbb{R}^4} f({}^{B}_{W}q,\ {}^{W}d,\ {}^{B}s) \tag{2-30} $$

$$ f({}^{B}_{W}q,\ {}^{W}d,\ {}^{B}s) = {}^{B}_{W}q^{\star} \otimes {}^{W}d \otimes {}^{B}_{W}q - {}^{B}s \tag{2-31} $$

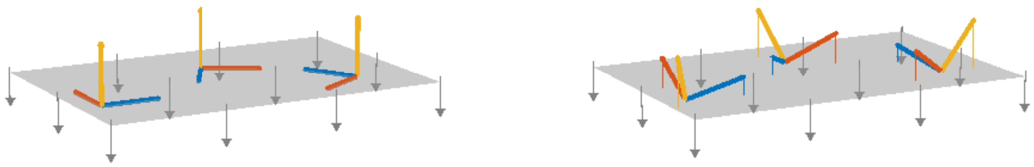
The optimization problem is solved using gradient descent. In Equation 2-32, the gradient descent algorithm is described for n iterations, yielding an attitude estimate ${}^{B}_{W}q$ based on initial orientation ${}^{B}_{W}q_0$ and variable step size μ.

$$ {}^{B}_{W}q_{k+1} = {}^{B}_{W}q_k - \mu \frac{\nabla f({}^{B}_{W}q,\ {}^{W}d,\ {}^{B}s)}{||\nabla f({}^{B}_{W}q,\ {}^{W}d,\ {}^{B}s)||}, \text{ for } k = 0, 1, 2, \dots n \tag{2-32} $$

$$ \nabla f({}^{B}_{W}q,\ {}^{W}d,\ {}^{B}s) = J^T({}^{B}_{W}q,\ {}^{W}d) f({}^{B}_{W}q,\ {}^{W}d,\ {}^{B}s) \tag{2-33} $$

There is no unique solution ${}^{B}_{W}q$. Instead, an infinite amount of possible orientations exist that satisfy Equation 2-29, as clarified in Figure 2-12.

It is shown that one homogeneous field does not suffice to fully determine the sensor orientation. Luckily, the JOHAN tracking device is assumed to always be used on Earth, in the presence of both gravity and the Earth's magnetic field. First consider the accelerometer



(a) Coordinate frames with different heading. **(b)** Coordinate frames with different orientation.

Figure 2-12: The 3D orientation of a rigid body cannot be fully determined from measuring one homogeneous field. While all three coordinate frames in Figure 2-12a have a different orientation, it is clear to see that the resulting sensor measurements are the same. Although it may be less obvious, the same holds true for the coordinate frames in Figure 2-12b. By taking into account a second homogeneous vector field, different from the first one, the orientation is fully determined.

acting as a 'gravity sensor'. The vector indicating the direction of gravity is normalized and becomes ${}^{W}g$. The accelerometer measurement is also normalized and denoted ${}^{B}a$.

$$ {}^{B}_{W}q = \begin{pmatrix} q_w & q_x & q_y & q_z \end{pmatrix}^T \tag{2-34} $$

$$ {}^{W}g = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T \tag{2-35} $$

$$ {}^{B}a = \begin{pmatrix} 0 & a_x & a_y & a_z \end{pmatrix}^T \tag{2-36} $$

The quaternion describing the rotation between ${}^{W}g$ and ${}^{B}a$ is found by minimizing the objective function in Equation 2-37. The Jacobian, required in solving the problem via gradient descent, is calculated via Equation 2-38.

$$ \begin{aligned} f_g({}^{B}_{W}q,\ {}^{B}a) &= {}^{B}_{W}q^{\star} \otimes {}^{W}g \otimes {}^{B}_{W}q - {}^{B}a \\ &= \begin{pmatrix} 2(q_xq_z - q_wq_y) - a_x \\ 2(q_wq_x + q_yq_z) - a_y \\ 2(\frac{1}{2} - q_x{}^2 - q_y{}^2) - a_z \end{pmatrix}, \end{aligned} \tag{2-37} $$

$$ J_g({}^{B}_{W}q) = \begin{pmatrix} -2q_y & 2q_z & -2q_w & 2q_x \\ 2q_x & 2q_w & 2q_z & 2q_y \\ 0 & -4q_x & -4q_y & 0 \end{pmatrix} \tag{2-38} $$

The same approach is applied to the magnetometer measurements ${}^{B}m$ of the Earth's magnetic field ${}^{W}b$. Assume that the Earth's magnetic field only has a vertical and a horizontal component and that the x-axis of the world frame Ψ_W coincides with the magnetic north. This implies that $b_y = 0$. Both the vector describing the magnetic field and the magnetometer measurement are normalized.

$$ {}^{W}b = \begin{pmatrix} 0 & b_x & 0 & b_z \end{pmatrix}^T \tag{2-39} $$

$$ {}^{B}m = \begin{pmatrix} 0 & m_x & m_y & m_z \end{pmatrix}^T \tag{2-40} $$

The objective function is given in Equation 2-41, the Jacobian in Equation 2-42.

$$\begin{aligned} f_b({}^{B}_{W}q,\ {}^{W}b,\ {}^{B}m) &= {}^{B}_{W}q^{\star} \otimes {}^{W}b \otimes {}^{B}_{W}q - {}^{B}m \qquad (2\text{-}41) \\ &= \begin{pmatrix} 2b_x(\frac{1}{2} - {q_y}^2 - {q_z}^2) - 2b_z(q_w q_y + q_x q_z) - m_x \\ 2b_x(q_x q_y - q_w q_z) + 2b_z(q_w q_x + q_y q_z) - m_y \\ 2b_x(q_w q_y + q_x q_z) + 2b_z(\frac{1}{2} - {q_x}^2 - {q_y}^2) - m_z \end{pmatrix}, \\ J_b({}^{B}_{W}q,\ {}^{W}b) &= \begin{pmatrix} 2b_z q_y & 2b_z q_z & -4b_x q_y - 2b_z q_w & -4b_x q_z + 2b_z q_x \\ -2b_x q_z + 2b_z q_x & 2b_x q_y + 2b_z q_w & 2b_x q_x + 2b_z q_z & -2b_x q_w + 2b_z q_y \\ 2b_x q_y & 2b_x q_z - 2b_z q_x & 2b_x q_w - 2b_z q_y & 2b_x q_x \end{pmatrix} \end{aligned}$$

$$(2\text{-}42)$$

The optimization problem that is formulated from this set of equations does have a unique solution. The optimization problem is formulated as:

$$\min_{{}^{B}_{W}q \in \mathbb{R}^4} f({}^{B}_{W}q,\ {}^{B}a,\ {}^{W}b,\ {}^{B}m) \qquad (2\text{-}43)$$

$$f_{g,b}({}^{B}_{W}q,\ {}^{B}a,\ {}^{W}b,\ {}^{B}m) = \begin{pmatrix} f_g({}^{B}_{W}q,\ {}^{B}a) \\ f_b({}^{B}_{W}q,\ {}^{W}b,\ {}^{B}m) \end{pmatrix}, \qquad (2\text{-}44)$$

$$J_{g,b}({}^{B}_{W}q,\ {}^{W}b) = \begin{pmatrix} J_g({}^{B}_{W}q) \\ J_b({}^{B}_{W}q,\ {}^{W}b) \end{pmatrix} \qquad (2\text{-}45)$$

Usually, solving the optimization problem would require a number of iterations. However, provided that the orientation estimation converges faster than the rate at which the orientation itself changes, one iteration per update step suffices. The one step gradient descent algorithm calculates the homogeneous field-based orientation estimation via Equation 2-46, with the gradient ∇f defined by Equation 2-47.

$$\begin{aligned} {}^{B}_{W}q_{\nabla,k} &= {}^{B}_{W}q_{k-1} - \mu_k \frac{\nabla f}{||\nabla f||} \qquad (2\text{-}46) \\ \nabla f &= J^{T}_{g,b}({}^{B}_{W}q_{k-1},\ {}^{W}b) f_{g,b}({}^{B}_{W}q_{k-1},\ {}^{B}a,\ {}^{W}b,\ {}^{B}m) \qquad (2\text{-}47) \end{aligned}$$

The value of stepsize μ_k is chosen in such a way that the orientation estimate always converges faster than the physical orientation rate. It is calculated as Equation 2-48:

$$\mu_k = \alpha ||{}^{B}_{W}\dot{q}_{\omega,k}|| T, \quad \text{where } \alpha > 1 \qquad (2\text{-}48)$$

where α is a tuning parameter, $||{}^{B}_{W}\dot{q}_{\omega,tk}||$ is the physical rate of change as measured by the gyroscope and T is the sample period.

Fusion process

The fusion process of the gyro-based orientation estimate ${}^{B}_{W}q_{\omega,k}$ and the homogeneous field-based orientation estimate ${}^{B}_{W}q_{\nabla,k}$ simply comes down to a weighted mean:

$${}^{B}_{W}q_t = \gamma_k\ {}^{B}_{W}q_{\nabla,k} + (1 - \gamma_k)\ {}^{B}_{W}q_{\omega,k}, \quad 0 \leq \gamma_k \leq 1 \qquad (2\text{-}49)$$

Now, γ_k is chosen in such a way that the weighted divergence rate of ${}^B_W q_\omega$ due to integral drift equals the weighted convergence rate of ${}^B_W q_\nabla$. The convergence rate of ${}^B_W q_\nabla$ is $\frac{\mu_k}{T}$ and the divergence rate of ${}^B_W q_\omega$ is represented by β. Via Equation 2-50 and Equation 2-51, the optimal value for γ_k is calculated.

$$(1-\gamma_k)\beta = \gamma_k \frac{\mu_k}{T} \tag{2-50}$$

$$\gamma_k = \frac{\beta}{\mu_k/T + \beta} \tag{2-51}$$

In Equation 2-48, $\alpha > 1$ ensures that the orientation estimation converges faster than the rate of change of the physical orientation. However, it does not prescribe how much faster. In other words, α has no upper bound. By choosing a very large α, μ_k also becomes very large. Equation 2-46 then simplifies to:

$${}^B_W q_{\nabla,k} \approx -\mu_k \frac{\nabla f}{||\nabla f||} \tag{2-52}$$

A very large μ_k yields β negligable in Equation 2-51. The formula for γ_k can now be rewritten as Equation 2-53, from which it can also be deduced that $\gamma_k \approx 0$.

$$\gamma_k \approx \frac{\beta T}{\mu_k} \tag{2-53}$$

Now, by substituting in Equations 2-28, 2-52, 2-53 into Equation 2-49, Equation 2-54 is obtained.

$${}^B_W q_t = \frac{\beta T}{\mu_k}\left(-\mu_k \frac{\nabla f}{||\nabla f||}\right) + (1-0)\left({}^B_W q_{k-1} + {}^B_W \dot{q}_{\omega,k} T\right) \tag{2-54}$$

This in turn, can be simplified to Equation 2-55, where ${}^B_W \dot{q}_k$ is defined by Equation 2-56.

$${}^B_W q_t = {}^B_W q_{k-1} + {}^B_W \dot{q}_k T \tag{2-55}$$

$${}^B_W \dot{q}_k = {}^B_W \dot{q}_{\omega,k} - \beta \frac{\nabla f}{||\nabla f||} \tag{2-56}$$

The final equations are computationally very efficient, directly using the sensor measurements. On top of that, the only tuning parameter that remains is β.

2-4 Models

The sports player motion is defined as the player's position, velocity, acceleration and orientation. Different filter configurations can be applied to estimate these quantities from 9-DoF IMU's and GPS. The most common approach is to try and estimate the full state at once. This can be done using an Unscented Kalman filter, referred to as the full state UKF. However, in sports player motion, rotations and translations are independent of each other. It is therefore possible, and maybe advantageous, to estimate rotations and translations separately. The accelerometer is then used as a 'gravity sensor' in the orientation estimation. The orientation is then used to turn the accelerometer into an 'acceleration' sensor in estimating the translational motion. For estimating translational motion, a linear Kalman filter is compared to a UKF. For estimating rotational motion, a quaternion-based Unscented Kalman filter and a Madgwick filter are investigated.

In this section, first the models for translational motion is described. Next, the model for rotational motion is described. Finally, these models are merged together to a full state model.

2-4-1 Linear Kalman filter

Translational motion is linear. It is therefore possible to use a linear Kalman filter [] to estimate these motions.

The state vector of the translational motion is defined as $x = \begin{pmatrix} p & v & a \end{pmatrix}^T$, with $p, v, a \in \mathbb{R}^3$. The propagation model predicts the next state, based on the current state and inputs. The model is simple; the acceleration is modeled to be constant, the velocity and position are obtained by integrating the acceleration. A block diagram is presented in Figure 2-13.

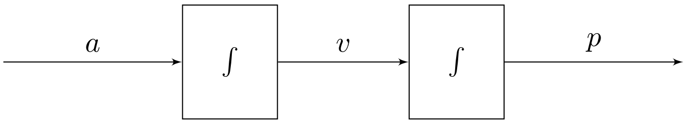


Figure 2-13: Translational model assuming constant acceleration

For the linear Kalman, the propagation model $x_{k+1} = Fx_k + v_k$ is described in Equation 2-57.

$$x_{k+1} = \begin{pmatrix} I_3 & I_3T & 0 \\ 0 & I_3 & I_3T \\ 0 & 0 & I_3 \end{pmatrix} x_k + v_k \tag{2-57}$$

The observation model is used to transform the state (x) into a form (y) that is comparable to sensor measurements (z). To observe translational motion, several sensors are available.

The GPS sensor provides measurements of the positions in $\mathbb{R}^2$, denoted $p_{x,GPS}$ and $p_{y,GPS}$. The vertical position is not measured accurately by GPS and thus not outputted. A barometer is also not present in the tracking device, so no measurement of p_z is available. However, the nature of sports player movements limits the vertical position between 0 meters (lying on the ground) and ±2.5 meters (really high jump). To deal with the absence of accurate height estimations, p_z measurements are simulated to be a constant value with high uncertainty. According to [23], the P-50 value for male shoulder height is 1.46 meters, meaning that 50% of the measured population has a shoulder heigth below this value. The value for the p_z-measurements is set to be 1.5 meters.

The accelerometer provides acceleration measurements in $\mathbb{R}^3$. The accelerations are measured in the body frame and contaminated with the measurement of gravitational acceleration. For now, assume that a correct orientation quaternion is available. In subsection 2-3-2, it is shown that world frame accelerations can be transformed into the body frame via Equation 2-37.

The transformation of body frame accelerometer measurements into world frame acceleration estimates is given by:

$$^{W}a = {}^{B}_{W}q \otimes {}^{B}a \otimes {}^{B}_{W}q^{\star} + \begin{pmatrix} 0 & 0 & g \end{pmatrix}^T \tag{2-58}$$

By performing the transformation in Equation 2-58 outside the filter, the observation model can remain linear.

The observation model for the linear Kalman filter, $y_k = Hx_k + w_k$ is given in Equation 2-59.

$$y_k = \begin{pmatrix} I_3 & 0 & 0 \\ 0 & 0 & I_3 \end{pmatrix} x_k + w_k \tag{2-59}$$

In the filter, y_k is compared to z_k, where

$$z_k = \begin{pmatrix} p_{GPS}{}^T & 1.5 & {}^{W}a^T \end{pmatrix}^T \tag{2-60}$$

2-4-2 Translational UKF

The translational UKF is an extension to the above-described linear Kalman filter in the observation model. The reason for this extension is the fact that the GPS receiver outputs directionless speed estimates. When this information is incorporated in the filter, the observation model becomes nonlinear.

The propagation model $x_{k+1} = f(x_k, w_k)$ for the translational UKF is equal to the propagation model for the linear Kalman filter (Equation 2-57). It is rewritten into Equation 2-61.

$$x_{k+1} = \begin{cases} p_{k+1} & = & p_k + v_k \cdot dt & + & w_{p,k} \\ v_{k+1} & = & v_k + a_k \cdot dt & + & w_{v,k} \\ a_{k+1} & = & a_k & + & w_{a,k} \end{cases} \tag{2-61}$$

where $w_k \in \mathbb{R}^9$ represents the process noise.

As stated before, the GPS sensor also outputs directionless speed estimates v_{GPS}. These speed estimates are calculated as the length of the horizontal velocity vector, via Equation 2-62. When incorporating the speed estimates in the observation model, it becomes nonlinear.

$$v = \sqrt{v_x{}^2 + v_y{}^2} \tag{2-62}$$

The observation model $y_k = h(x_k, v_k)$, with $y_k \in \mathbb{R}^7$, is described by Equation 2-63.

$$y_k = \begin{cases} y_{pos,k} & = & p_k & + & v_{p,k} \\ y_{vel,k} & = & \sqrt{v_{x,k}{}^2 + v_{y,k}{}^2} & + & v_{v,k} \\ y_{acc,k} & = & a_k & + & v_{a,k} \end{cases} \tag{2-63}$$

In the measurement vector z_k, the speed estimate provided by the GPS is added.

$$z_k = \begin{pmatrix} p_{GPS}{}^T & 1.5 & v_{GPS} & {}^{W}a^T \end{pmatrix}^T \tag{2-64}$$

It is expected that taking into account the speed estimate will improve the acceleration estimates.

2-4-3 Rotational UKF

Previously, models for translational motion are described. In these observation models, it was assumed that correct orientation estimates were available. The model for the filter with which the orientation estimates are to be obtained is described here.

In modeling translational motion, the accelerations are modeled to be constant. Similarly, in modeling rotational motion, the angular rate is modeled to be constant. A block diagram is provided in Figure 2-14.

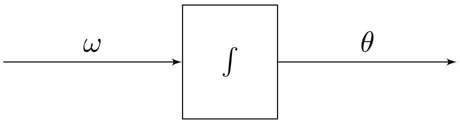


Figure 2-14: Rotation model assuming constant angular velocity

The propagation model $x_{k+1} = f(x_k, w_k)$ for the rotational UKF (Kraft UKF) is similar to the gyroscope-based orientation estimate in the Madgwick filter. The main difference is that in the Madgwick filter, the gyroscope measurements are directly fed into the filter as ω. In the Kraft UKF, ω is considered a state and is assumed to be constant. The rotational propagation model is defined in Equation 2-65.

$$x_{k+1} = \begin{cases} q_{k+1} & = & q_k \otimes q_{w_q,k} \otimes q_{\nabla,k} & & \\ \omega_{k+1} & = & \omega_k & + & w_{\omega,k} \end{cases} \tag{2-65}$$

where q_w denotes the rotation due to process noise and q_Δ represents the differential rotation by $\omega \cdot dt$.

The observation model $y_k = h(x_k, v_k)$ in the Kraft UKF consists of three elements.

$$y = \begin{cases} y_{rot,k} & = & \omega_k & + & v_{rot,k} \\ y_{acc,k} & = & {}^B a_k & + & v_{acc,k} \\ y_{mag,k} & = & {}^B m_k & + & v_{mag,k} \end{cases} \tag{2-66}$$

where v_{rot}, v_{acc} and v_{mag} denote the measurement noise in the gyroscope, accelerometer and magnetometer, respectively.

The first element of the observation model concerns angular velocity: $z_{rot} = \omega_k + v_{rot}$. The values for z_{rot} are compared to the gyroscope measurements.

The remaining submodels both concern a sensor that measures a homogeneous vector field with respect to the body frame. The accelerometer measures gravity g and the magnetometer measures the Earth's magnetic field b, both assumed to remain constant. The Earth's magnetic field is location-dependent and is defined by the angle of declination ϑ_d and the angle of inclination ϑ_i. The angle of declination represents the angle between the local meredian and the magnetic North, where West is defined to be positive. The angle of inclination is the angle at which the magnetic North vector points into the ground. In the area of Noordwijk, the Netherlands, the angle of declination and inclination are $\vartheta_d = -2.24°$ and $\vartheta_i = 66.90°$, respectively.

$${}^W g = \begin{pmatrix} 0 \\ 0 \\ 9.81 \end{pmatrix}, \qquad {}^W b = \begin{pmatrix} \cos(\vartheta_i) \\ -\sin(\vartheta_i)\sin(\vartheta_d) \\ -\sin(\vartheta_i)\cos(\vartheta_d) \end{pmatrix} \tag{2-67}$$

To observe the current orientation, the homogeneous fields are rotated into body frame via the current quaternion estimate, in similar fashion as in subsection 2-3-2. The expected body frame measurements for the accelerometer and magnetometer are given in Equation 2-68 and Equation 2-69, respectively.

$$\begin{pmatrix} 0 \\ {}^{B}a \end{pmatrix} = q \otimes \begin{pmatrix} 0 \\ {}^{W}a \end{pmatrix} \otimes q^{\star} \tag{2-68}$$

$$\begin{pmatrix} 0 \\ {}^{B}m \end{pmatrix} = q \otimes \begin{pmatrix} 0 \\ {}^{W}m \end{pmatrix} \otimes q^{\star} \tag{2-69}$$

Note that the expected world frame accelerations ${}^{W}a$ include both translational and gravitational acceleration:

$${}^{W}a = {}^{W}a_{translational} + \begin{pmatrix} 0 & 0 & a_{gravity} \end{pmatrix}^{T} \tag{2-70}$$

The measurement vector z_k is built up from the inertial sensors contained in the IMU, as denoted in Equation 2-71.

$$z_k = \begin{pmatrix} {\omega_{gyro}}^{T} & {a_{acc}}^{T} & {m_{mag}}^{T} \end{pmatrix}^{T} \tag{2-71}$$

2-4-4 Full state

The full state UKF combines the quaternion-based Kraft UKF and the translational UKF. This yields a state vector $x \in \mathbb{R}^{16}$ consisting of $x = \begin{pmatrix} q^T & \omega^T & p^T & v^T & a^T \end{pmatrix}^T$.

The propagation model for the full state UKF is described in Equation 2-72. It merges the propagation models from the rotational and translational UKF.

$$x_{k+1} = f(x_k, w_k) = \begin{cases} q_{k+1} & = & q_k \otimes q_{w_q,k} \otimes q_{\nabla,k} & & \\ \omega_{k+1} & = & \omega_k & + & w_{\omega,k} \\ p_{k+1} & = & p_k + v_k \cdot dt & + & w_{p,k} \\ v_{k+1} & = & v_k + a_k \cdot dt & + & w_{v,k} \\ a_{k+1} & = & a_k & + & w_{a,k} \end{cases} \tag{2-72}$$

The observation model for the full state UKF inherits elements of the Kraft UKF and the translational UKF.

$$y_k = h(x_k, v_k) = \begin{cases} y_{rot,k} & = & \omega_k & + & v_{rot,k} \\ y_{acc,k} & = & {}^{B}a_k & + & v_{acc,k} \\ y_{mag,k} & = & {}^{B}m_k & + & v_{mag,k} \\ y_{pos,k} & = & p_k & + & v_{p,k} \\ y_{vel,k} & = & \sqrt{{v_{x,k}}^2 + {v_{y,k}}^2} & + & v_{v,k} \end{cases} \tag{2-73}$$

Note that the world-frame acceleration 'measurement' from the translational UKF has disappeared. Instead, the accelerometer is used both as a 'gravity sensor' and an 'acceleration' sensor.

2-4-5 Filter configurations

Except for the full state UKF, the described filters are a combination of a filter estimating the rotational part, and a filter estimating the translational part of the state vector. Figure 2-15 schematically presents the way the filters are implemented. Each time-step, the simulated measurements are first fed into a rotational estimation filter. This is either the Kraft quaternion-based unscented Kalman filter, or the Madgwick filter. Then, using the estimated orientation q, the accelerometer measurements are rotated into world-frame. The gravity vector is subtracted and the acceleration measurements are, together with the GPS measurements, fed into the translational filter. The translational filter is either the UKF or the linear Kalman filter. These filters output position, velocity and acceleration estimates in world frame.

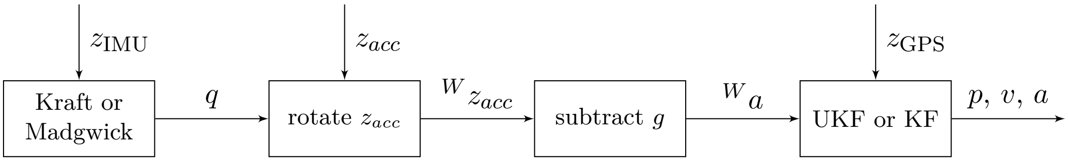


Figure 2-15: The rotational filter (Kraft or Madgwick) takes sensor measurements z_{IMU} as an input and yields orientation estimate q. With orientation estimate q, the accelerometer measurements are rotated into world frame after which the gravity vector is subtracted. The resulting world-frame acceleration measurements and GPS measurements z_{GPS} are fed into the translational filters (UKF and KF) to obtain p, v and a.

2-5 Determining the most suitable filter

The end goal is to determine the filter most suitable for outdoor field sports player motion estimation with the JOHAN tracking device. To determine the most suitable filter, the estimation accuracy and computational load are reviewed. The Pareto optimality [24] is determined for each filter and each part of the state vector. The filter that scores best on both estimation accuracy and computational load is selected as the most suitable filter.

2-6 Summary

The most important coordinate frames used in this thesis are the world frame and the body frame. With the word ‘orientation’, the difference in orientation between these two coordinate frames is referred to. The Earth coordinate frame is used to transform GPS measurements into world frame, the field frame is used in the post-processing to enable positional comparison between players.

The sensors used in the thesis are a GPS receiver and a 9-DoF IMU (combined gyroscope, accelerometer and magnetometer). The GPS receiver is primarily used to measure position. The IMU is primarily used to determine the tracking device orientation. The orientation is used to transform accelerometer measurements from the body frame into the world frame. When subtracting the gravity vector from these rotated accelerometer measurements, world-frame accelerations are obtained.

The most important filters in the thesis are the Unscented Kalman filter (UKF) and Madgwick filter. Via the unscented transformation, the UKF is capable of state estimation in nonlinear systems. The Madgwick filter is popular for its ease of implementation and low computational load. It estimates only the orientation, using IMU measurements.

Models are developed for estimating rotational and translational motion using the Unscented Kalman filter (UKF). The filter can estimate the full state at once, but due to the independence of rotational and translational motion in field sports, these can also be estimated separately. The linearity of translational motion enables the use of the linear Kalman filter for translational state estimation. The use of quaternions for orientation estimation cause difficulties in the UKF. An approach to deal with these is presented.

The benefit of two-step estimation lies in the different use of the accelerometer. During the estimation of rotational motion, little trust is put in the accelerometer measurements, since these measurements are disturbed by translational accelerations. Subsequently in the translational motion estimation, when a relatively accurate orientation estimate is present, the accelerometer is more heavily relied on.

The method used for deciding upon which filter is most suitable is Pareto optimization. The computational load and the estimation accuracy of each filter is used to create a Pareto front.

Chapter 3

Simulation

To design a sensor fusion filter for player state estimation from IMU and GPS data, it is desired to have a dataset containing both the sensor data and the corresponding motion data. Such a dataset can be obtained by conducting experiments, or by simulating motion. In the early design phase simulated data is beneficial, for all signals are then known, including noise.

In the simulation, reference states x_{ref} and their corresponding expected sensor measurements z are defined. The simulated sensor measurements are fed into the filters to estimate the original states x. By comparing the estimated states with the simulated reference states, filter performance is assessed. The process is depicted in Figure 3-1.

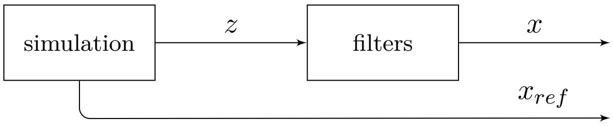


Figure 3-1: Setup of the simulation experiment

First some simple motions consisting of only pure rotations and pure translations are simulated. These simple simulations illustrate some difficulties that arise in the motion estimation problem at hand. Subsequently, a more complex motion is simulated by modeling the angular velocities and accelerations as a random walk. Finally, a motion representative for a sports player is simulated based on external measurements from a real-life experiment.

The sample frequency is chosen to be 100Hz, resembling that of the real IMU. A reference state is obtained by designing the angular velocities ω_{ref} and accelerations a_{ref}. The remainder of the reference state is determined using the propagation model in Equation 2-72. The corresponding expected measurements are calculated using the observation model as in

Equation 2-73.

$$x_{\omega,k} = \omega_{\text{ref},k} \tag{3-1}$$
$$x_{a,k} = a_{\text{ref},k} \tag{3-2}$$
$$x_{\text{ref},k+1} = f(x_{\text{ref},k}, 0) \tag{3-3}$$
$$z_k = h(x_{\text{ref},k}, v_k), \tag{3-4}$$

where $x_{\omega,k}$ and $x_{a,k}$ represent the parts of the state vector corresponding to angular velocity and accelerations, respectively. The measurement noise v_k is defined to be zero mean white noise with variances corresponding to measurement noise on the real sensors, derived in the calibration process, Equation 3-5.

$$v_{\text{gyro}} = 1 \cdot 10^{-4}, \quad v_{\text{acc}} = 1 \cdot 10^{-2}, \quad v_{\text{mag}} = 1 \cdot 10^{-4}, \quad v_{\text{p}_{GPS}} = 1 \cdot 10^{-2}, \quad v_{\text{v}_{GPS}} = 1 \cdot 10^{-3}. \tag{3-5}$$

The set of simulated measurements is fed into the different filter configurations. The resulting state estimates x_{est} are compared to the simulated reference states x_{ref}.

3-1 Simulating pure rotations and translations

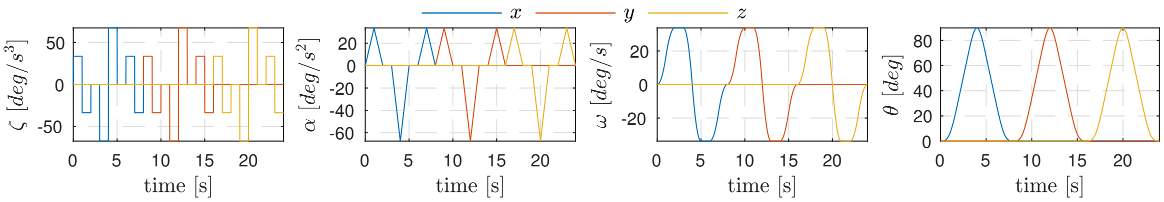
First pure rotations without any translational motion are simulated. Subsequently, pure translations without any rotational motion are simulated. The goal is to see how accurate the filter configurations can estimate these simple motions.

3-1-1 Rotations

Angular jerk ζ is modeled as a piecewise constant signal. Via integration angular acceleration α, angular velocity ω and Euler angles θ are obtained:

$$\alpha = \int \zeta dt, \quad \omega = \int \alpha dt, \quad \theta = \int \omega dt \tag{3-6}$$

The designed rotational motion is depicted in Figure 3-2.



(a) Angular jerk **(b)** Angular acceleration **(c)** Angular velocity **(d)** Euler angles

Figure 3-2: Designed rotational motion.

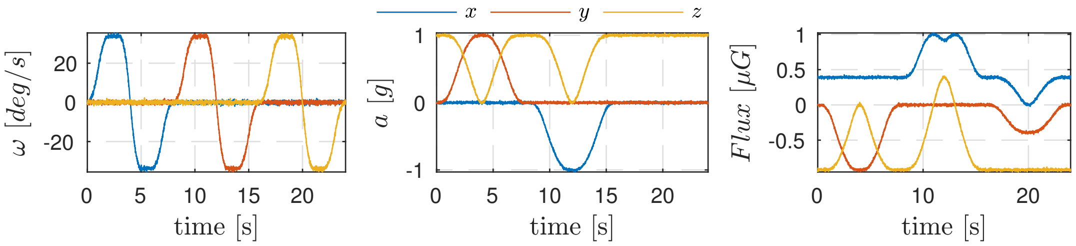
Simulating reference states

The designed angular velocity is used to simulate the reference state x_{ref} via the propagation model f defined in Equation 2-72. The initial states are zeros, except for the orientation: $q_0 = \begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}^T$. The angular velocity in the reference states is defined to be the designed angular velocity ω. The remainder of the state is obtained via Equation 3-3. Note that no process noise is added to the reference state, to obtain a smooth reference state.

Simulating sensor measurements

Since no translational motion is simulated, the GPS sensor is simulated to measure only measurement noise v. The gyroscope is expected to measure ω_{ref}. The accelerometer and magnetometer are expected to measure the gravity vector and magnetic field vector, rotated by q_{ref}. To obtain these simulated sensor measurements, the reference states x_{ref} and measurement noise v are fed in into the observation model defined in Equation 2-73, as is done in Equation 3-4.

The simulated IMU measurements are depicted in Figure 3-3. The simulated GPS measurements are disregarded, since only the rotation is estimated for now.



(a) z_{gyro} **(b)** z_{acc} **(c)** z_{mag}

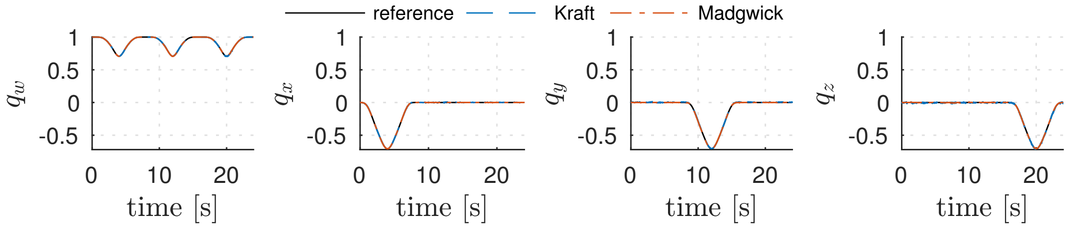
Figure 3-3: Simulated sensor measurements corresponding to the rotational motion designed in Figure 3-2.

Estimation accuracy

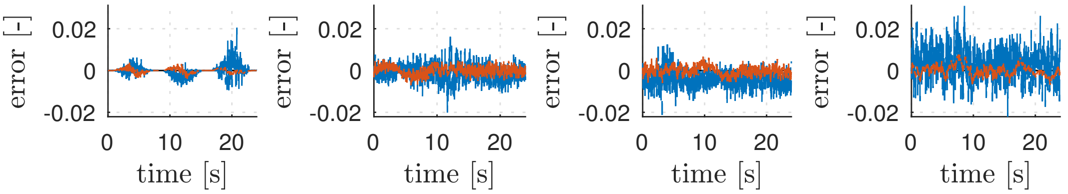
The simulated sensor data is fed into the Kraft UKF and the Madgwick filter. The quaternion estimates and tracking errors are displayed in Figure 3-4.

Since quaternions are not that intuitive, it makes sense to convert the orientation estimates and tracking errors back into Euler angles, as displayed in Figure 3-5. Note the spike in the estimation error around 12 seconds. This error spike is caused by a singularity in the conversion between quaternions and Euler angles. If one was to calculate the tracking error from the Euler angles, the error would be heavily influenced by this singularity peak.

To obtain a more accurate performance indicator of the orientation estimators, the absolute angular error is calculated. The absolute angular tracking error $\theta_{error} \in \mathbb{R}$ is determined from



(a) Quaternion estimates.



(b) Errors in the quaternion estimates.

Figure 3-4: Quaternion estimates and corresponding tracking error of the Kraft UKF and Madgwick filter on simulated rotational motion.

the scalar value in of the error quaternion, as in Equation 3-7.

$$\begin{aligned} q_{error} &= q_{\text{ref}}^{\star} \otimes q_{est} \\ \theta_{error} &= 2\arccos(q_{w,error}) \end{aligned} \tag{3-7}$$

The RMSE's of the absolute angular tracking errors are presented in Table 3-1.

RMSE	θ [degrees]
Kraft	1.25
Madgwick	0.46

Table 3-1: RMSE's in estimating simple rotations.

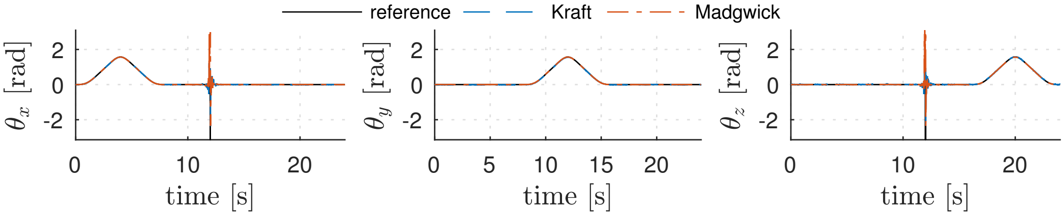
It is shown that the Madgwick orientation estimates are more accurate than the Kraft filter estimates.

3-1-2 Translations

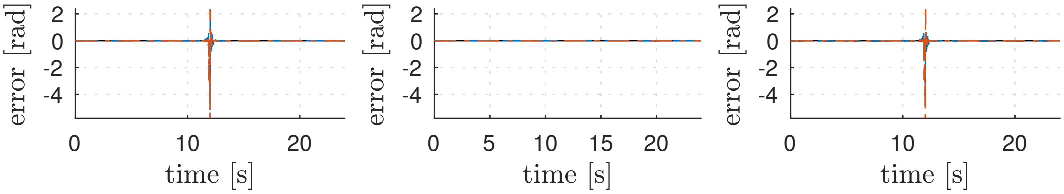
In similar fashion as the simulation of pure rotations, now pure translations are simulated. Again, a piecewise constant jerk is assumed, see Figure 3-6a. From the jerk, the accelerations, velocity and positions are derived.

$$a = \int j dt, \quad v = \int a dt, \quad p = \int v dt \tag{3-8}$$

The translational motion that is simulated is depicted in Figure 3-6.

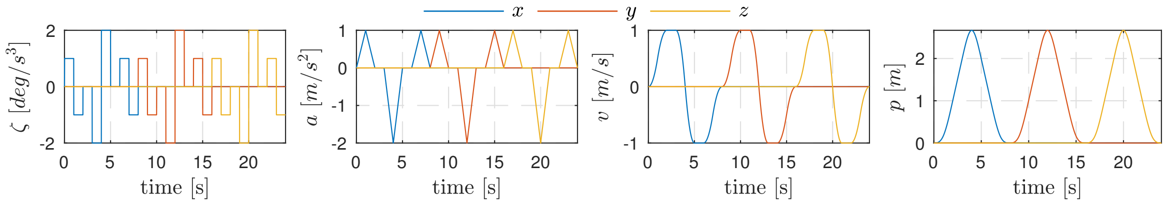


(a) Euler angle estimates.



(b) Errors in the Euler angle estimates.

Figure 3-5: Euler angle estimates and corresponding tracking errors of the Kraft filter and Madgwick filter on simulated rotational motion.



(a) Jerk **(b)** Acceleration **(c)** Velocity **(d)** Position

Figure 3-6: Simulated translational motion. From a piecewise constant jerk, the acceleration, velocity and position in x-, y- and z direction are determined.

Simulating reference states

With the acceleration a defined, the remainder of the state vector is obtained using the propagation model in Equation 2-72, see Equation 3-9.

$$x_{\text{ref},k+1} = f(x_{\text{ref},k}, 0) \tag{3-9}$$

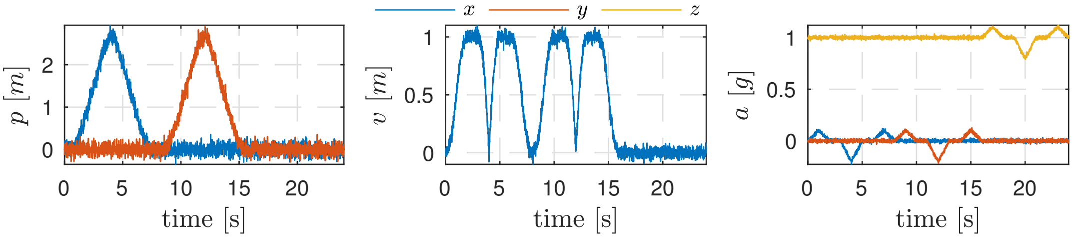
Simulating sensor measurements

To simulate sensor measurements corresponding to the reference motion, again the observation model in Equation 2-73 is used, see Equation 3-10. However, since the GPS sensor is unable to measure the z-position, this simulated measurement is replaced by 1.5 m.

$$z_k = h(x_{\text{ref},k}, v_k) \tag{3-10}$$

$$z_{p_z,k} = 1.5 \tag{3-11}$$

The simulated sensor measurements are depicted in Figure 3-7.

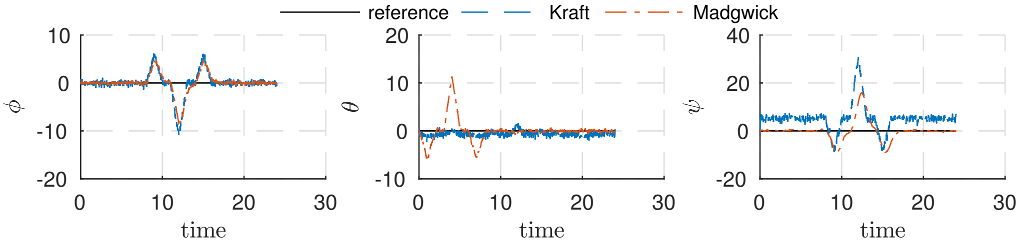


(a) z_p **(b)** z_v **(c)** z_a

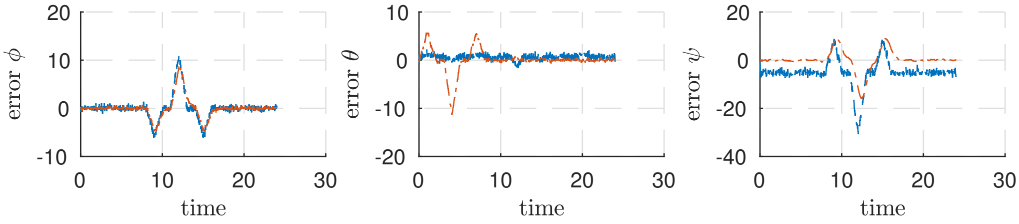
Figure 3-7: Simulated sensor data corresponding to translations described in Figure 3-6.

Estimation accuracy

The simulated sensor measurements are first fed into the rotational motion estimation filters, to obtain quaternion estimates. These quaternion estimates are used to rotate the accelerometer measurements into the world frame and rid them of gravitational acceleration. By doing so, world frame acceleration quasi-measurements are obtained. These accelerations, together with the simulated position and velocity measurements, are fed into the translational filters UKF and KF.



(a) Estimated Euler angles.



(b) Errors in estimated Euler angles.

Figure 3-8: Euler angle estimates and corresponding tracking error by Kraft (blue) and Madgwick (red) on simulated translational motion.

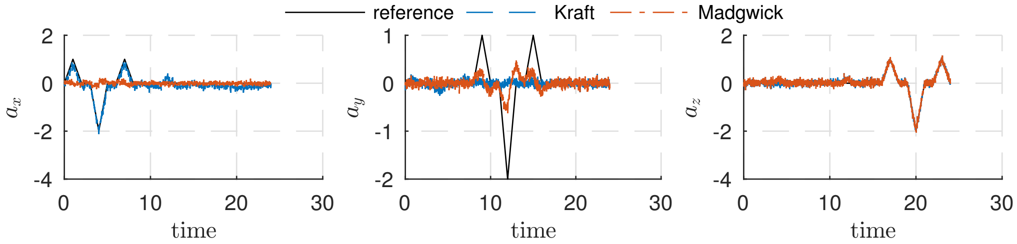
As is visible in Figure 3-8, the rotational filters are being fooled by the translational acceleration. Due to the translational acceleration, the direction of the total acceleration vector changes. Since the rotational UKF views the accelerometer as a 'gravity sensor', it assumes that the rotation is changing. An error in the orientation estimate introduced by this phenomenon will harm subsequently created translational motion estimates.

The RMSE's of the absolute angular tracking errors are presented in Table 3-2. The large absolute angular RMSE in the Kraft filter stems mostly from the sustained error in the ψ-angle.

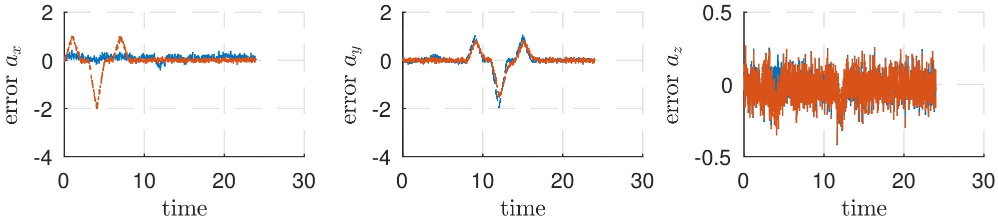
RMSE	θ [degrees]
Kraft	8.00
Madgwick	6.08

Table 3-2: Angular RMSE's in estimating simple translations.

The translational filter uses quaternion estimates to rotate accelerometer measurements into world frame. Gravity is then substracted, yielding an 'acceleration' sensor. The effect of erroneous orientation estimates on acceleration estimates is visible in Figure 3-9. The acceleration is partially filtered out, since it credited to a fictitious rotation. For the Kraft filter, this effect is clearly visible in the acceleration estimate in the y-axis, for the Madgwick filter it is visible in both the x and y-axis.



(a) Acceleration estimates



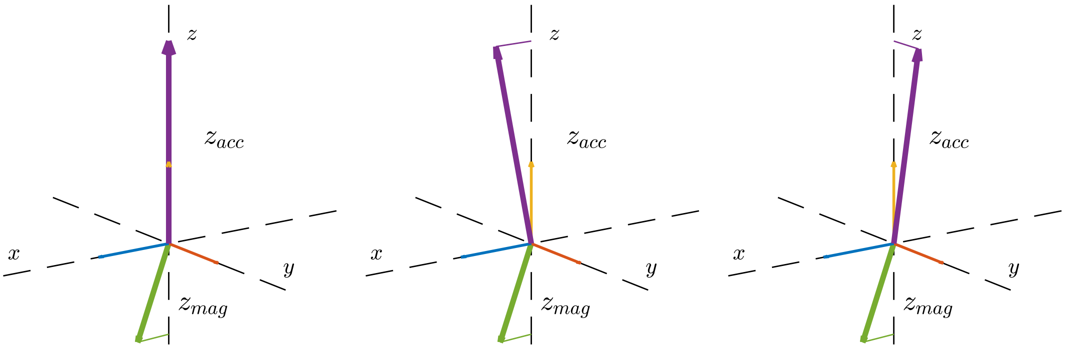
(b) Errors in acceleration estimates

Figure 3-9: Acceleration estimates and their tracking errors. The filters used to estimate rotations are UKF (blue) and Madgwick (red). The translational filter used is the UKF.

The difference in estimation accuracy per axis is remarkable. One would expect that the accuracy in x-axis and y-axis would be similar. The reason for this lies in the fact that the simulated orientation of the tracking device happens to coincide with the world frame. The Madgwick filter views the magnetometer measurements as a horizontal component and a vertical component, while the Kraft filter views it as a three-dimensonal vector. Therefore, a pure acceleration in x-direction is sensed by the Kraft filter, while it is not sensed by the Madgwick filter. In y-direction, both filters interpret the changing accelerometer measurements as a change in orientation.

The RMSE's of the different filter configurations on position, velocity and acceleration is presented in Table 3-3.

The RMSE's in Table 3-3 show that despite the larger absolute orientation error, the configurations with the Kraft filter estimate accelerations more accurate than those with the Madgwick filter. This is explained by the fact that the Madgwick filter is fooled by the translational motion in x-direction.



(a) No acceleration. **(b)** Acceleration in x-direction. **(c)** Acceleration in y-direction.

Figure 3-10: The direction of the vector measured by the accelerometer and the magnetometer when subject to translational accelerations. When an acceleration in x-direction is applied, the Madgwick filter assumes a rotation over the y-axis. The Kraft filter is able to distinguish the translational motion. When an acceleration is applied in the y-direction, both filters are fooled.

		Position [m]			Velocity [m/s]			Acceleration [m/s^2]		
		x	y	z	x	y	z	x	y	z
Kraft	UKF	0.053	0.067	1.361	0.350	0.501	0.153	0.235	0.419	0.085
	KF	0.038	0.045	1.372	0.271	0.428	0.331	0.240	0.419	0.081
Madgwick	UKF	0.054	0.067	1.364	0.583	0.603	0.166	0.410	0.395	0.088
	KF	0.039	0.044	1.374	0.322	0.411	0.332	0.412	0.395	0.085

Table 3-3: The RMSE's on translational motion estimates in simple simulated translations.

3-2 Simulating a random walk

The simulated pure rotations and pure translations have displayed some weaknesses in the filters. However, in real player motion pure translations and pure rotations are very unlikely to occur. In real player motion, translational acceleration is almost always accompanied by rotational motion. To simulate combined rotations and translations, a random walk is simulated in both the angular velocity and accelerations.

3-2-1 Simulating reference states and measurements

The propagation model f and observation model h as described in subsection 2-4-4 are used to simulate combined rotations and translations. An initial state x_0 is chosen to be zeros, except for the orientation quaternion. Furthermore, process noise and measurement noise are modeled as white noise. The variance of the process noise w is defined as in Equation 3-12, the variance of the measurement noise v is given in Equation 3-13.

$$w_q = 1 \cdot 10^{-16}, \quad w_\omega = 1 \cdot 10^{-3}, \quad w_p = 1 \cdot 10^{-6}, \quad w_v = 1 \cdot 10^{-4}, \quad w_a = 1 \cdot 10^{-2} \tag{3-12}$$

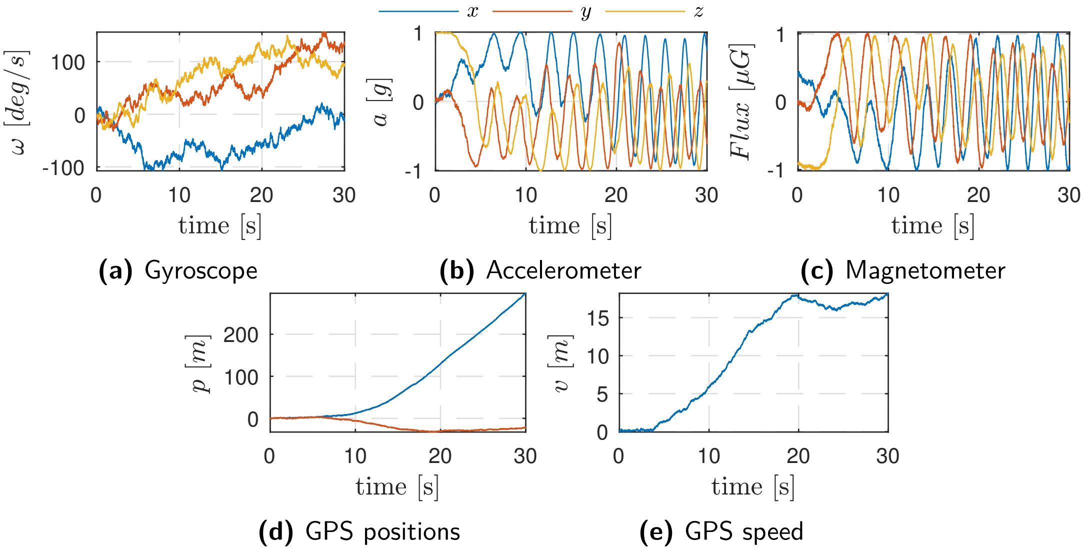
$$v_{\text{gyro}} = 3 \cdot 10^{-3}, \quad v_{\text{acc}} = 1 \cdot 10^{-4}, \quad v_{\text{mag}} = 1 \cdot 10^{-4}, \quad v_{p\text{GPS}} = 1 \cdot 10^{-1}, \quad v_{v\text{GPS}} = 1 \cdot 10^{-2} \tag{3-13}$$

The reference states are now obtained via Equation 3-14, the measurements are simulated via Equation 3-15.

$$x_{\mathrm{ref},k} = f(x_{\mathrm{ref},k-1}, w_k) \tag{3-14}$$

$$y_k = h(x_k, v_k) \tag{3-15}$$

In each step, the process noise vector is added to the state vector. Due to the relatively large process noise in the angular velocities and accelerations, these states are affected most by the random walk. The resulting simulated measurements are presented in Figure 3-11.



(a) Gyroscope **(b)** Accelerometer **(c)** Magnetometer

(d) GPS positions **(e)** GPS speed

Figure 3-11: Simulated sensor measurements corresponding to random walk generated random motion.

The simulated measurements are fed into the different filter configurations. The resulting state estimates are compared to the reference state to assess the filter estimation accuracy.

3-2-2 Results

On the random walk simulated data, the filters for estimating rotational motion and the filters for estimating translational motion are tested.

Rotational motion First, the accuracy of the orientation estimate is investigated. Figure 3-12 shows quaternion estimates and the tracking error for both the Kraft filter and Madgwick filter. The Kraft filter slightly outperforms the Madgwick filter. The angular RMSE is defined as the RMSE of the absolute angular deviation as described in Equation 3-7, The angular RMSE is 17.8 degrees for the Kraft filter and 25.0 degrees for the Madgwick filter. These values are very large, indicating that something is wrong.

In the simulation that is depicted in Figure 3-12, the Kraft filter seems to outperform the Madgwick filter. To check whether this is the case in general, a set of 50 repetitions of the random walk simulation is created. Taking the mean angular RMSE over all repetitions yields angular RMSE's of 47.2 and 47.6 degrees for the Kraft filter and Madgwick filter, respectively.

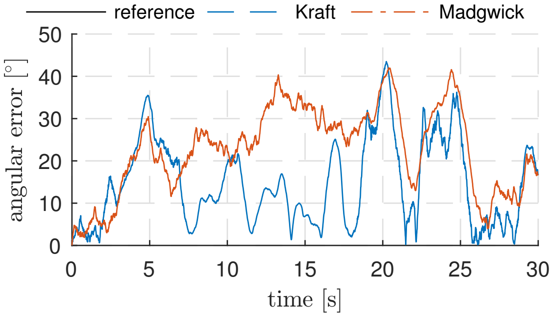


Figure 3-12: Angular error for both the Kraft filter and the Madgwick filter. Note that for both filters, the angular error reaches above 30 degrees.

Translational motion Now the translational UKF and linear Kalman filter are compared. The reference orientation quaternions are used to perfectly rotate the accelerometer measurements into world frame. The resulting RMSE's on the translational part of the state vector are presented in Table 3-4. On all states the RMSE's are small. The estimation accuracy of the UKF and KF is similar. This performance was expected, since measurements for p_x, p_y and a_x, a_y and a_z are available.

	Position [m]			Velocity [m/s]			Acceleration [m/s^2]		
	x	y	z	x	y	z	x	y	z
UKF	0.157	0.211	0.026	0.102	0.117	0.015	0.076	0.029	0.008
KF	0.211	0.213	0.034	0.135	0.131	0.016	0.042	0.042	0.009

Table 3-4: RMSE's of translational motion estimates in random walk simulations, with correct world-frame acceleration measurements available.

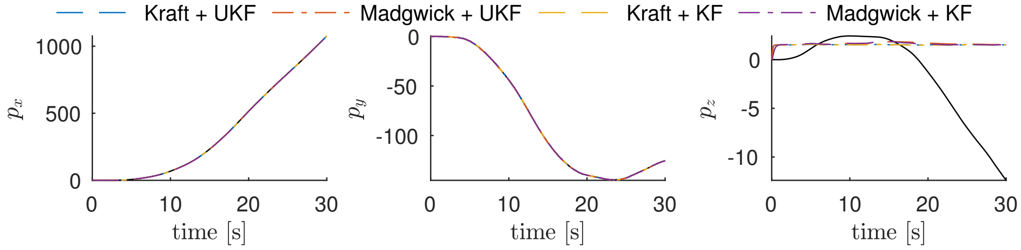
In real life, the true orientation is not available. The orientation estimate is provided either by the Kraft filter, or by the Madgwick filter and, as Figure 3-12 displays, these orientation estimates contain quite large errors. The effect of a poor orientation estimate on the estimates of translational motion is investigated next.

As Figure 3-13 shows, the position estimates suffer from orientation estimation errors. The filters prove unable to accurately estimate the z-position.

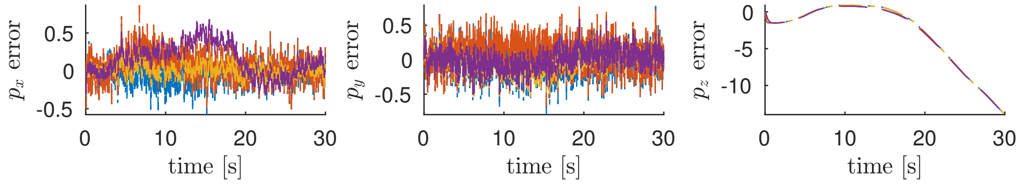
Over to the velocity estimates. The estimates of each filter configuration are presented in Figure 3-14. The filters are unable to accurately track the velocities. When the reference velocity stabilizes, around $t = 25$ seconds, the error decreases. However, the filter performance is not satisfactory.

In Figure 3-15, the acceleration estimates are displayed. The filter configurations that use Madgwick as an orientation estimator filter out almost all acceleration in x- and y-direction. The Kraft-based filter configurations estimate the x-acceleration remarkable more accurate. This indicates that the orientation error in the Kraft estimation probably does not affect the direction of the x-axis.

Figures 3-13, 3-14 and 3-15 show that the estimation accuracy deteriorates severely by errors in orientation estimates. The RMSE's over the complete set of 50 random walk simulations



(a) Position estimate.



(b) Errors in the position estimate.

Figure 3-13: Position estimates and tracking errors of the simulated random walk. Note the error on the z-position. Since no measurements of vertical position are available, a dummy measurement of 1.5 m is assumed. In this simulation, the reference z grows to -12, causing problems in the filters.

confirm this. The RMSE's over the translational part of the state vector are presented in Table 3-5.

		Position [m]			Velocity [m/s]			Acceleration [m/s^2]		
		x	y	z	x	y	z	x	y	z
Kraft	UKF	0.191	0.216	30.765	5.746	7.157	2.796	1.621	2.787	0.811
	KF	0.162	0.274	30.781	1.787	10.530	2.864	1.620	2.787	0.811
Madgwick	UKF	0.192	0.215	30.786	6.598	6.248	3.823	3.233	2.680	1.238
	KF	0.268	0.277	30.806	3.765	10.669	3.151	3.234	2.680	1.237

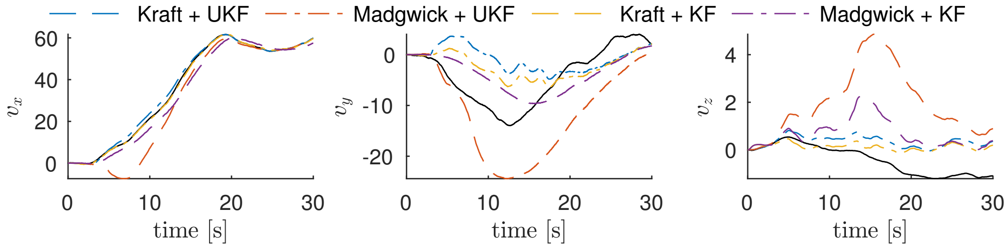
Table 3-5: RMSE's of translational motion estimates in random walk simulations. When comparing the RMSE's to those in Table 3-4, the degradation of accuracy due to inaccuracy in orientation estimates is striking. Especially the errors in the z-direction are large. This is caused by the absence of z-position measurements.

3-2-3 Remarks

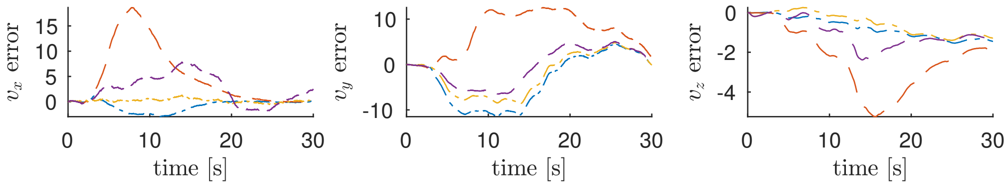
A major limitation in the random walk simulations is the fact that the motion simulated with random walks is often far from representative for player motion.

The small estimation errors in Table 3-4 show that the absence of p_z measurements can be corrected for by accurate orientation estimates.

Simulated accelerations are in $-18 < a_{\text{sim}} < 16$ m/s^2, while real player accelerations should vary roughly between $-4 < a_{\text{player}} < 4$ m/s^2. These simulated high accelerations are maintained for a longer period of time, while in reality high accelerations only last for a short time.



(a) Velocity estimate.



(b) Errors in the velocity estimate.

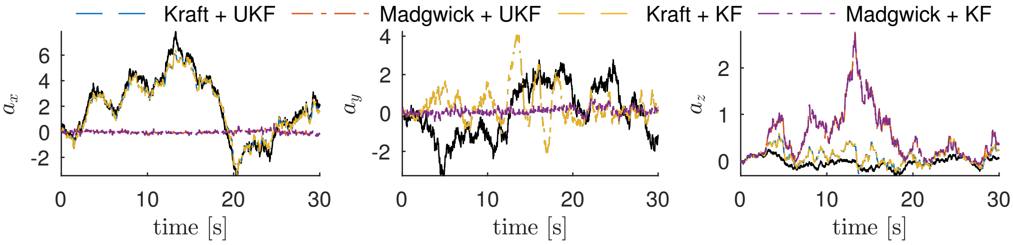
Figure 3-14: Velocity estimate and corresponding tracking error on simulated random walk. Both the UKF and linear Kalman filter are unable to accurately track velocities.

The high accelerations are simulated to be measured by the accelerometer. The accelerometer measurements are fed into the rotational filter. Both the Madgwick filter and Kraft filter are implemented to view the accelerometer measurements as a 'gravity' sensor. For real data, this is viable, since translational accelerations tend to remain smaller than gravity. However, with the high accelerations in the simulation, this assumption is violated. This leads to poor orientation estimates in the simulation. As is shown, errors in the orientation estimate have major consequences for the estimation accuracy of translational motion.

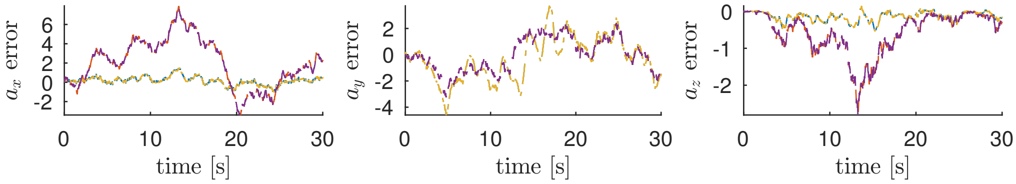
Due to the high accelerations, the simulated velocities reach unrealistic values as well: $-200 < v_{sim} < 200$ m/s. These high velocities in turn lead to large traveled distances. Now for x and y, it is no problem if positions become large. For the z-position however, this is a problem, since it violates the assumption that the z-position lies around 1.5 m. Because the simulated z-position is allowed to grow boundless, the estimation error in this state can become very large, negatively influencing estimation accuracy in position, velocity and acceleration.

Regarding rotational motion, the simulated angular velocity reaches a maximum value of 280 degrees per second. In reality, player rotational rates may also reach such values, but only for a very short period of time. In the simulation these high angular rates are maintained for a stretch of multiple seconds.

Summarized, the simulated motion is not representative for real sports player motion. Since the filters are tailored to tracking real player motion, they are unable to accurately estimate the non-representative motions in this simulation. To properly investigate filter performance, motions need to be simulated that better represent sports player motion. In Section section 3-3 an attempt is made to do so.



(a) Acceleration estimate.



(b) Errors in the acceleration estimate.

Figure 3-15: Acceleration estimates and corresponding tracking error on simulated random walk. Taking into account that the world-frame acceleration measurements are fed into the filters, it can be concluded that the estimation errors are huge.

3-3 Simulating VICON-measured motion

The random walk simulations described in Section 3-2 are of limited use, since the simulated motions are not directly representative for sports player motion. To simulate a motion that is representative for sports player motion, data from the VICON experiment is used. Details of the VICON experiment are found in Section 4-2, for now it suffices to know that the measurements exist of 200 Hz positions and Euler angles. These external measurements are used to simulate the corresponding tracking device measurements.

3-3-1 Method

The angular velocity is obtained by approximating the derivative of the Euler angles. This is done calculating the differences between adjacent measurements and dividing the result by the sample time, as in Equation 3-16.

$$\omega'_{\text{ref},i} = \frac{[\theta_{\text{VICON},i+1} - \theta_{\text{VICON},i}]}{T} \tag{3-16}$$

When approximating the discrete derivative via this approach, peaks arise due to noise amplification. To deal with this problem, the *Savitsky-Golay filter* is applied [25]. The Savitsky-Golay filter regards the datapoints within windowsize ws. Through these points, an n-th order polynomial is fit. Then, the window is shifted one place and the process repeats itself. The filter is applied via the MATLAB-command $\omega_{\text{ref}} = \texttt{sgolayfilt}(\omega'_{\text{ref}}, n, ws)$.

The same approach is applied to the position measurements to obtain reference velocities. It is then applied once more to obtain reference accelerations. The VICON system measures at 200 Hz. The sample frequency is reduced to 100 Hz using the `decimate()`-function.

From the obtained angular velocities and accelerations, the reference states and corresponding measurements are simulated via Equations 3-1 to 3-4.

Using this approach, 27 datasets with a total of 13084 datapoints of reference states x_{ref} and measurements z are obtained. The datasets are divided into different categories defined by the pace at which the subject travels. Some simulations are derived from measurements of a *walking* or *jogging* motion, while others correspond to a *running* or *sprinting* motion. The different motion types are presented in Table 3-6.

Pace [km/h]	Category
$0 < v \leq 7$	Walking
$7 < v \leq 14$	Jogging
$14 < v \leq 20$	Running
$20 < v$	Sprinting

Table 3-6: The different categories of simulated motion are defined by the pace at which the subject travels.

In analyzing the resulting motion estimates, this differentiation in motion type is retained. It is investigated what the estimation accuracy is of each filter configuration on sports player representative motion data.

3-3-2 Results

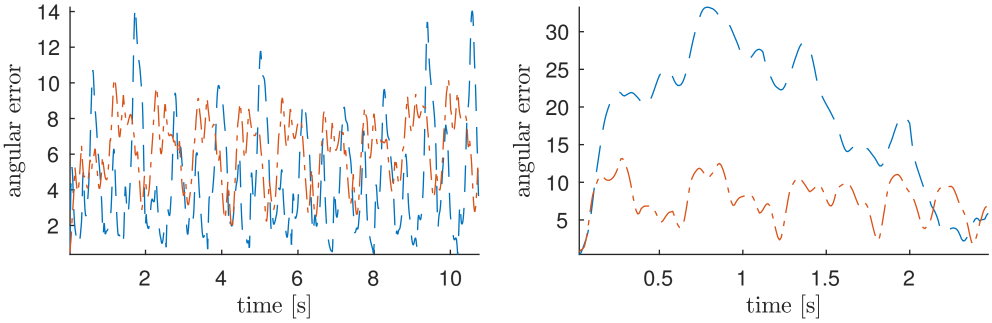
Via the method described in Section 3-3-1, motions are simulated that should be representative for sports player motion. The simulated measurements are again fed into the different filter configurations to assess estimation accuracy. In these simulations, the computational load is assessed as well.

Estimation accuracy

From the random walk simulations in Section 3-2, it has become clear that erroneous orientation estimates and z-positions deviating too far from 1.5 meters are factors that greatly reduce estimation accuracy. In this section, first the estimation accuracy of rotational motion is investigated, followed by the estimation accuracy of translational motion.

Rotational motion The absolute angular error is calculated for each simulated motion and filter configuration. The angular errors of both a simulated walking motion and a simulated sprinting motion are depicted in Figure 3-16.

The resulting RMSE's are presented in Table 3-7. The Madgwick filter shows superior performance over the Kraft filter in accuracy of estimating the orientation. For both filters, it is shown that the estimation error grows when the pace in the simulated motion increases. The error on the Kraft estimates even grows to 101.2° in the sprinting simulations. Since the Kraft filter performs best on simulated *walking* motions, one could suspect that the filter is tuned to perform better on slower motions. This is however not the case. For the tuning method, the reader is directed to Section D-2.



(a) Angular error on a simulated *walking* motion.

(b) Angular error on a simulated *sprinting* motion.

Figure 3-16: Angular error for both a walking and a sprinting motion. The errors at walking pace are way lower than those at sprinting pace.

$\text{RMSE}_{\lvert\theta\rvert}$ [°]	Walking	Jogging	Running	Sprinting
Kraft	6.67	29.69	25.65	101.21
Madgwick	7.49	7.14	9.02	10.29

Table 3-7: The RMSE's on the rotational motion estimates in the VICON-based simulations.

Keeping in mind the influence of orientation errors that was found in Section 3-2, it is expected that angular errors like these are problematic for the accuracy in translational motion estimates. To check whether this is the case, the translational state estimates are investigated.

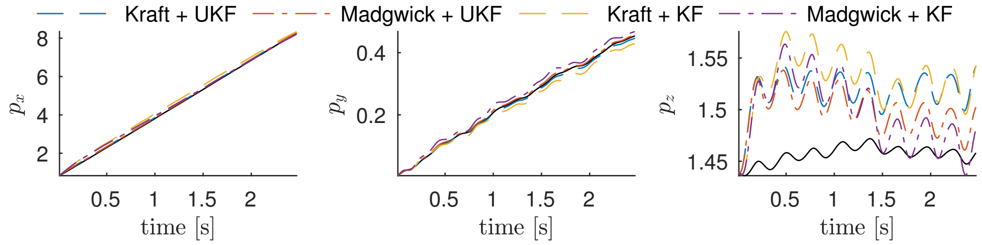
Translational motion The position estimates for a simulated sprinting motion are depicted in Figure 3-17. The estimation errors are all within acceptable range. As expected, the reference z-position remains close to 1.5 meters. As a consequence, the error on the z-position remains small. The rhytmic motion of a sprinting sports player is reflected in the sine-shaped z-position, in the reference state as well as in the estimates.

The accuracy of velocity estimates depends on orientation estimates, as Figure 3-18 shows. A clear difference between accuracy of Kraft-based and Madgwick-based velocity estimates is visible, expecially in y-direction.

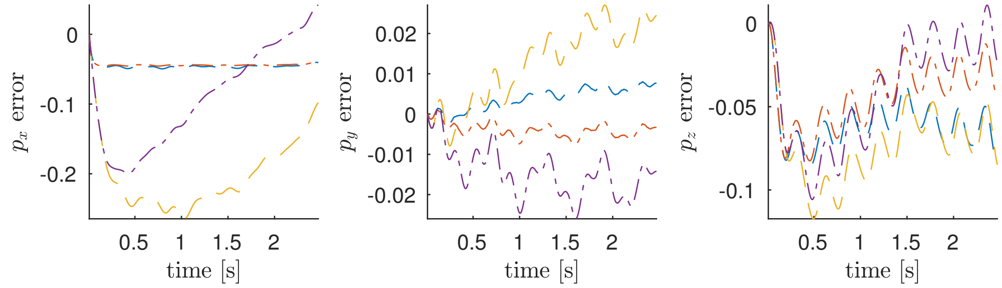
In Figure 3-19, a few remarkable features can be noted. First of all, the range in which the accelerations live. It was assumed that player acceleration ranges roughly from -4 to 4 m/s^2. This simulation suggests that, at least the tracking device, moves more violently. It is important to keep in mind that the reference acceleration is obtained via discrete differentiation and that its accuracy is therefore questionnable. The errors in x and y direction are larger than those in z-direction.

The acceleration estimates vary roughly between -10 and 10 m/s^2. It goes without saying that a post-processing operation is required to obtain values that are representative for player acceleration (as opposed to tracking device accelerations). Such an operation could be applying a simple Butterworth filter.

The RMSE's corresponding to the translational part of the state vector are presented per motion type in Table 3-9. The position estimates outputted by each filter configuration are



(a) Position estimate.



(b) Errors in the position estimate.

Figure 3-17: Position estimates and corresponding tracking errors on simulated sprinting motion. The UKF outperforms the linear Kalman filter in estimation accuracy.

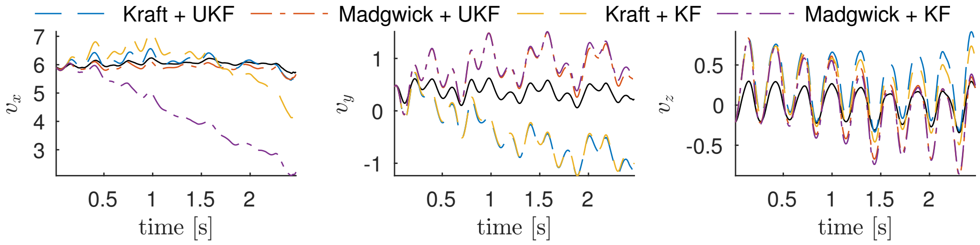
equally accurate. In the velocity estimates, the UKF shows slightly more accurate estimation than the linear KF. In the acceleration estimates, the filter configurations with the Kraft filter as orientation estimator are outperformed by the configurations with the Madgwick filter. This makes sense, given the difference in orientation estimation accuracy. Although better than the Kraft-based acceleration estimates, the estimation errors in the Madgwick-based accelerations are still quite large: Assuming that player acceleration values lies within $-4 \leq v \leq 4$ m/s^2, the RMSE on x-acceleration in sprints is ±25% of the complete range.

Computational load

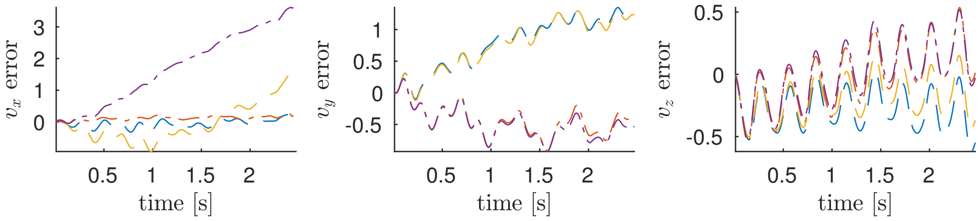
The computational load of each filter configuration is determined by measuring the time it takes to calculate the state estimates of 1 second of simulated data, equivalent to 100 measurements. The computational load per filter configuration is determined over 50 sets of simulation data and presented in Table 3-8.

		Computational load	
		μ	(σ)
Kraft	UKF	0.340	(0.010)
	KF	0.306	(0.012)
Madgwick	UKF	0.053	(0.001)
	KF	0.027	(0.001)

Table 3-8: Computational load per filter configuration. The computational load is expressed as the computation time per second of simulated data. The data shows that the Madgwick filter has a much lower computational load than the Kraft filter. The linear Kalman filter is slightly faster than the linear UKF.



(a) Velocity estimates.



(b) Errors in the velocity estimates.

Figure 3-18: Velocity estimates and corresponding tracking errors on simulated sprinting motion. The UKF outperforms the linear Kalman filter in estimation accuracy.

Whether the measurements fed into the filters is simulated or real-world data does not influence computational load. The computational load presented in Table 3-8 is thus representative for all simulations and experiments.

Pareto optimization

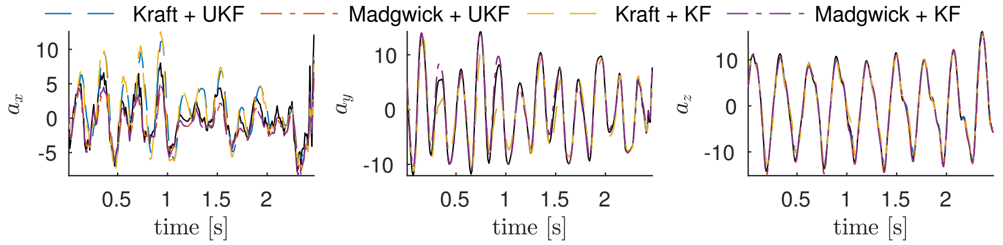
For comparison, the computational load is plotted against the estimation accuracy. Each part of the state vector is viewed separately. Furthermore, a differentiation is made between the simulated motion types. The estimation accuracy is defined as the mean RMSE over the x, y and z-direction of that particular part of the state vector. The results are presented in Figure 3-20 and Figure 3-21.

It is shown that the orientation estimates outputted by the Madgwick filter are more accurate than those of the Kraft filter. The more accurate orientation estimate is visible in the acceleration estimates. At walking pace, both filters estimate accelerations with similar accuracy. When the pace in the simulated is increased however, the acceleration estimates from Kraft-based filter configurations heavily deteriorate. Accuracy in acceleration estimates of the Madgwick-based filter configurations does also decrease with increasing pace in the simulated motion, but less severe.

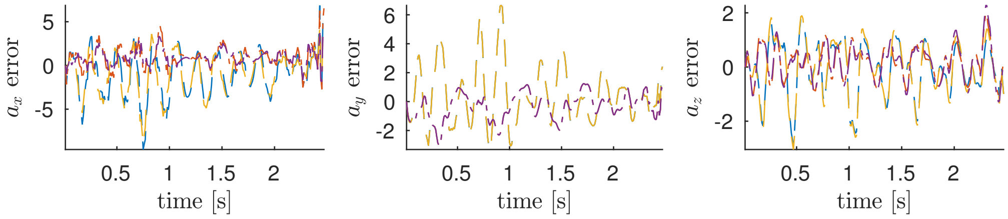
The figures clearly show that the Kraft-based filter configurations are far from Pareto optimal. When looking at the Madgwick-based filter configurations, performances of the UKF and KF are not far apart.

3-3-3 Remarks

Like the random walk simulations from Section 3-2, the VICON-based simulations also have their limitations. The Savitsky-Golay filter is used to derive velocities from positions and



(a) Acceleration estimates.



(b) Errors in the acceleration estimates.

Figure 3-19: Acceleration estimates and corresponding tracking errors on simulated sprinting motion. The superior orientation estimation accuracy of the Madgwick filter reflects in the acceleration estimation accuracy of Madgwick-based filter configurations.

accelerations from velocities. The degree of similarity of the simulated accelerations and the actual accelerations in the experiment is thus dependent on the chosen order and windowsize in the Savitsky-Golay filter. It is likely that the simulated motion is smoother than the actual motion.

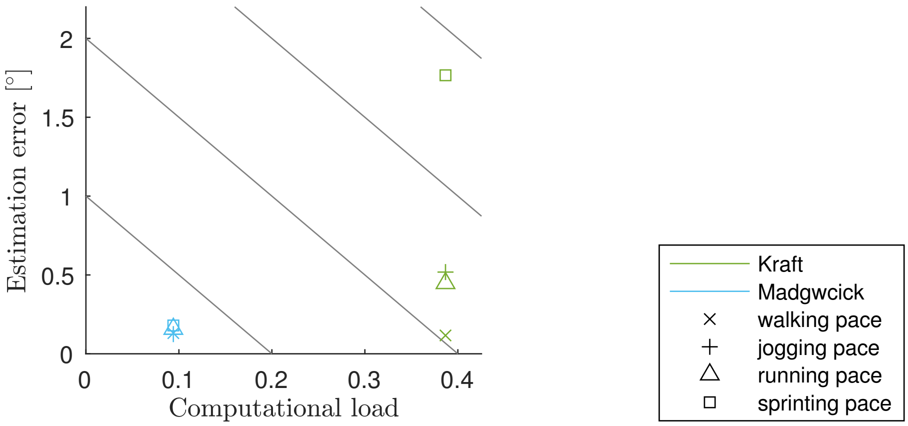
The angular error of both the Madgwick filter and the Kraft filter is larger than 5 degrees, even for te simulated motions at walking pace. This is not a good sign for the real world experiments, as it can be assumed that the real sensors will suffer from more complex disturbances than their simulated counterparts.

3-4 Summary

Simple rotational and translational motion is simulated. It is shown that translational accelerations can disturb orientation estimates. When this is the case, the false orientation estimate disqualifies the accelerometer as a reliable ‘acceleration’ sensor.

The simulation in which the angular velocities and accelerations are modeled as a random walk stresses the importance of accurate orientation estimates. The absence of height measurements can be corrected for as long as accurate orientation estimates (and thus accurate acceleration measurements) are available. As soon as orientation estimates become inaccurate, errors on estimates of the translational part of the state vector start to grow.

By simulating motions measured in a real-life experiment (that is described in Section 4-2), representative player motions are simulated. The simulations suggest that the accelerations of the tracking device vary roughly between -10 and 10, rather than between -4 and 4 as was previously assumed. Furthermore, it is shown that orientation estimates get less accurate as the pace of the simulated motion increases. Finally, the simulations show that the Madgwick filter outperforms the Kraft filter both in estimation accuracy and in computational load.

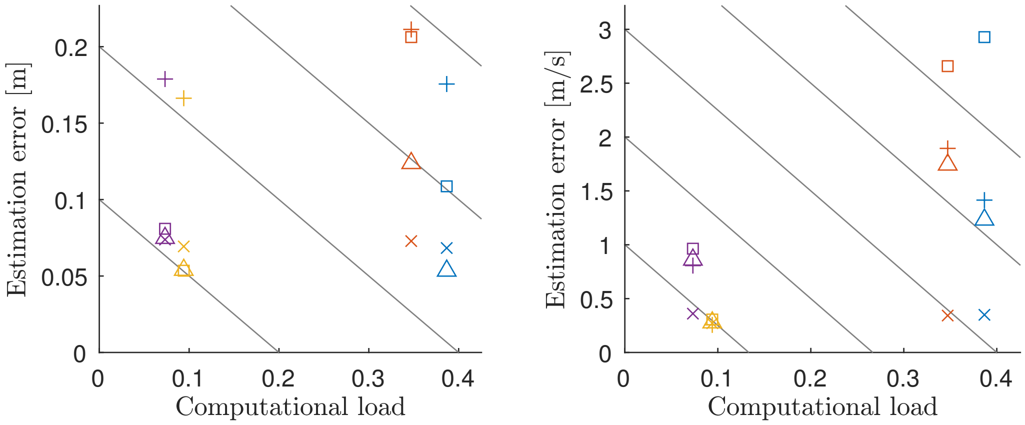


(a) Pareto front of orientation estimates. **(b)** Legend

Figure 3-20: Pareto front of the computational load and estimation accuracy of rotational motion. The Kraft filter and Madgwick filter are compared.

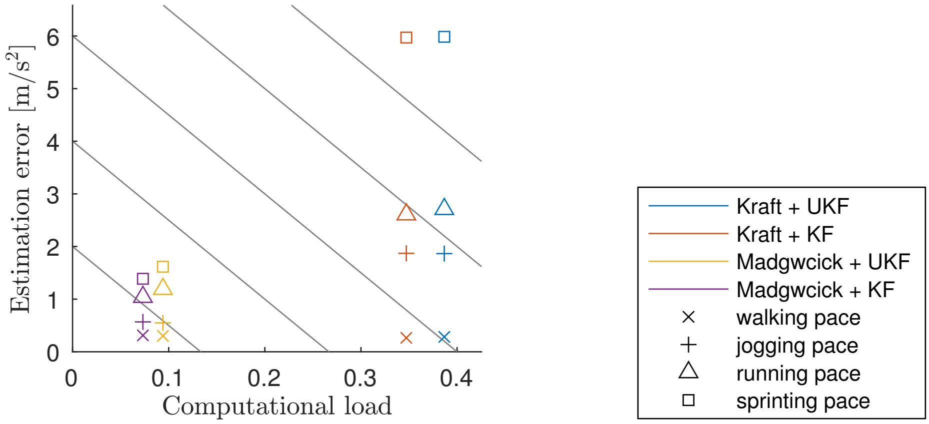
		Position [m]			Velocity [m/s]			Acceleration [m/s^2]		
		x	y	z	x	y	z	x	y	z
walking										
Kraft	UKF	0.005	0.004	0.196	0.365	0.548	0.136	0.335	0.442	0.073
	KF	0.016	0.006	0.196	0.690	0.221	0.117	0.260	0.442	0.094
Madgwick	UKF	0.007	0.002	0.198	0.302	0.345	0.167	0.524	0.306	0.077
	KF	0.017	0.006	0.198	0.720	0.234	0.128	0.533	0.306	0.097
jogging										
Kraft	UKF	0.018	0.013	0.495	1.278	2.180	0.785	2.127	2.563	0.906
	KF	0.065	0.069	0.500	1.863	3.198	0.623	2.048	2.563	1.002
Madgwick	UKF	0.025	0.002	0.472	0.076	0.203	0.489	0.738	0.429	0.478
	KF	0.055	0.009	0.472	1.607	0.334	0.479	0.659	0.429	0.615
running										
Kraft	UKF	0.033	0.013	0.114	0.772	2.105	0.820	2.940	3.365	1.826
	KF	0.168	0.072	0.131	1.380	3.264	0.585	2.568	3.365	1.890
Madgwick	UKF	0.037	0.003	0.121	0.112	0.449	0.272	2.040	0.886	0.649
	KF	0.099	0.010	0.116	1.911	0.408	0.263	1.491	0.887	0.746
sprinting										
Kraft	UKF	0.050	0.019	0.256	3.087	3.181	2.515	5.827	7.666	4.461
	KF	0.171	0.067	0.381	2.904	3.139	1.932	5.757	7.666	4.486
Madgwick	UKF	0.044	0.003	0.113	0.137	0.451	0.330	2.929	0.968	0.951
	KF	0.119	0.009	0.115	2.190	0.398	0.302	2.200	0.969	0.993

Table 3-9: The RMSE's on translational motion estimates in the VICON-based simulations of each motion type. The position estimates of all filter configurations are of similar accuracy, regardless of the pace that is simulated. In the velocity estimates, the slightly superior performance of the UKF over the linear Kalman filter is visible. In estimating accelerations, the filter configurations relying on the Madgwick filter for orientation estimation clearly outperform those with the Kraft filter.



(a) Pareto front of position estimates.

(b) Pareto front of velocity estimates.



(c) Pareto front of acceleration estimates.

(d) Legend

Figure 3-21: Pareto fronts of computational load and estimation accuracy of the translational parts the state vector. A differentiation is made based on the pace at which the subject in the simulated motion moves. In the position estimates, similar estimation accuracy is shown for each filter configuration. In estimating velocities, the Madgwick-based filter configurations show superior estimation accuracy over the Kraft-based filter configurations. The difference in estimation accuracy is particularly evident in Figure 3-21c. The estimation accuracy of each filter configuration is similar at walking pace, but with increasing pace, the Kraft-based filter configuration estimation accuracy heavily deteriorates. The computational load per filter configuration equal for each motion type.

Chapter 4

Experiments and results

In simulation, the filters have shown to be capable of tracking sports player motion. However, the real world, effects occur that have not been simulated.

The measurement noise on each of the sensors was simulated as zero-mean white noise (ZMWN). On the real sensors, the noise is not ZMWN. The noise on the GPS measurements is better modeled as a piece-wise constant signal (subsection 2-2-3), the accelerometer and gyroscope suffer from biases, scale errors and coupling errors (subsection 2-2-1) and the magnetometer measurements contain hard-iron errors and soft-iron errors (subsection 2-2-2). Furthermore, motions in the real world may differ from those that are simulated. To assess filter performance on real sensor data, two different experiments are carried out.

The first experiment is the *reference trajectory experiment.* The subject travels along a trajectory, of which the positions are determined via Google Maps. Subsequently, the deviations of the estimated positions with respect to the real positions are determined.

The second experiment is the *VICON experiment.* In the experiment, the subject moves in similar fashion as a sports player, imitating actions from a sports game. The VICONMotion Capture system, with its infrared cameras and reflective markers, is used to measure the performed motions in terms of position and orientation.

4-1 Reference trajectory experiment

In the reference trajectory experiment, a predefined trajectory is traveled while being tracked with the JOHAN tracking device. The reference trajectory coordinates are obtained via Google Maps. The measurements are fed into the different filter configurations, yielding filtered position estimates. The raw GPS measurements and the filtered position estimates are compared to the reference trajectory. It is investigated if the filtering operation improves the position estimates.

4-1-1 Obtaining reference coordinates from Google Maps

The reference trajectory is defined by straight line segments on a truck parking lot, clearly visible on Google Maps. The locations (lat, long) of several points per line segment are selected in Google Maps. A straight line is fit through the locations and the points in which the line segments cross are calculated. The resulting reference trajectory is displayed in Figure 4-1. To express the position error in meters, the trajectory is transformed from Earth frame (lat, long) to world frame (x, y). Google Maps provides latitudes and longitudes with 6 decimal places, corresponding to 0.11 m accuracy.

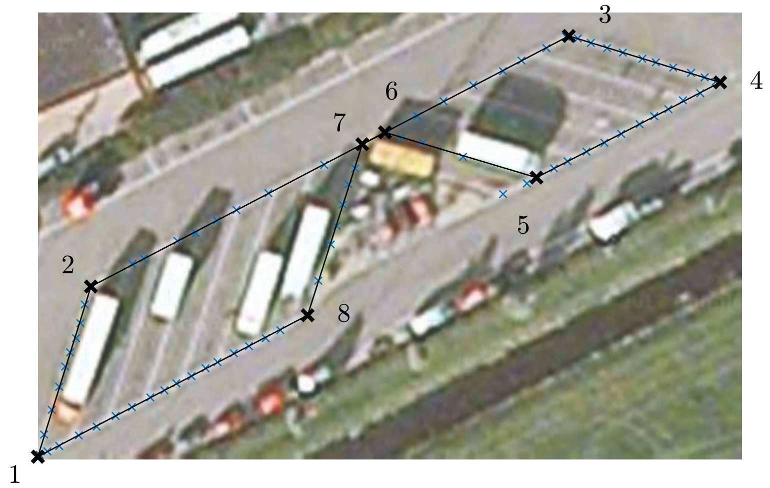


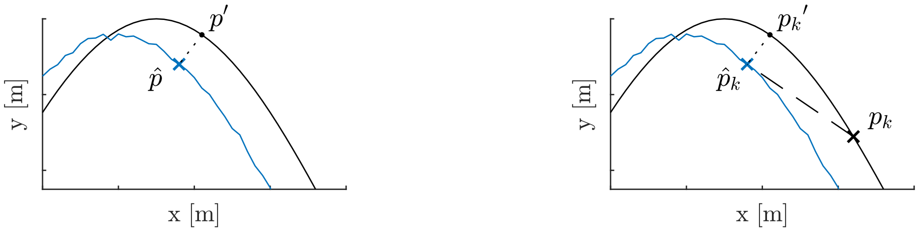
Figure 4-1: The reference trajectory in the reference trajectory experiment. Locations of points on the reference trajectory line segments (blue x's) are obtained via Google Maps. A straight line is fit through the selected points. The points are determined in which the line segments cross (black x's), yielding the complete reference trajectory (full line). The direction of the trajectory is in ascending order of crossing points.

4-1-2 Defining the estimation error

The error is defined as the distance between a position estimate $\hat{p}$ and the closest point on the reference track p', as displayed in Figure 4-2a. The RMSE of all position estimates is calculated via Equation 4-1.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{p}_i - p'_i)^2}{n}} \tag{4-1}$$

Note that the position error as defined is independent of time. This time independency is an important weakness. Assume the actual position at time instance k to be p_k. The position estimate and the closest reference point are now denoted $\hat{p}_k$ and p'_k, respectively. The actual position does not necessarily coincide with the closest reference point ($p_k \neq p'_k$). This results in a smaller position error estimate than the actual position error, as is illustrated by Figure 4-2b.



(a) Position error as defined

(b) Actual position error

Figure 4-2: Reference trajectory and its estimate. The error is defined to be the distance between the estimated position $\hat{p}$ and the closest point on the reference trajectory p' (4-2a), while the actual position p_k may be another point on the reference trajectory (4-2b).

4-1-3 Results

With two tracking devices, the reference trajectory is traveled four times, yielding 8 measurements. The resulting dataset contains 122.624 position measurements in total. The raw measurements are displayed in Figure 4-3. The figure shows that the error on the position measurements is not ZMWN.

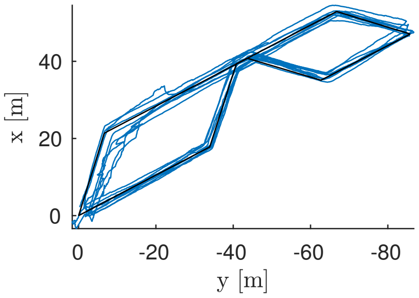


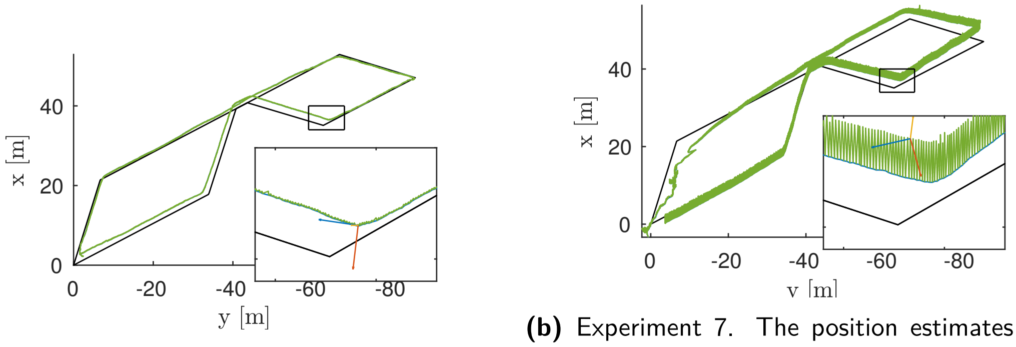
Figure 4-3: Raw position measurements in the reference trajectory experiment.

The RMSE of the raw position measurements is 1.03 m. The measurements are fed into the filter configurations to assess the filter performance. The RMSE's per filter configuration are displayed in Table 4-1.

filter configuration	RMSE [m]
raw measurements	1.03 (0.04)
Kraft + UKF	1.21 (0.32)
Madgwick + UKF	1.23 (0.29)
Kraft + KF	1.21 (0.32)
Madgwick + KF	1.22 (0.29)

Table 4-1: Root mean square errors on the position estimates with raw and filtered measurements. The filtering action deteriorates the position estimates.

The table shows that the position estimates after filtering are worse than the raw GPS estimates. This is caused by erroneous orientation estimates disturbing the position estimates, as Figure 4-4b shows.



(a) Experiment 2. The position estimates closely follow the raw GPS measurements, the RMSE is 0.867 m.

(b) Experiment 7. The position estimates diverge between subsequent GPS measurements, due to poor orientation estimates. RMSE is 1.75 m.

Figure 4-4: Position- and orientation estimates in the reference trajectory experiment for two different repetitions. The reference trajectory, raw GPS measurements and position estimates are represented by the black, blue and green line, respectively. A coordinate frame is plotted to indicate the orientation estimate. It is shown that, next to bad GPS measurements, erroneous orientation estimates are a cause for large RMSE's.

4-1-4 Remarks

The GPS measurements as outputted by the tracking device are shown to measure positions with 1.03 *meter accuracy*. That is, assuming that the reference trajectory perfectly describes the traveled trajectory.

Furthermore, the experiment shows that no matter which filter configuration is applied, the *filtering operations bring no improvement* to position estimates. Due to the on-chip filtering operation in the GPS receiver, hardly any noise appears to be present on the measurements. It is thus not possible to estimate a smooth line through the cloud of measurements, as would be the case with ZMWN. Whether or not the filtering operation improves or deteriorates the position estimates with respect to the raw measurements is solely dependent on the direction in which the filter estimates deviate from the measurements: towards the reference trajectory or not.

4-2 VICON experiment

The VICON experiment is executed with the aim of assessing the accuracy of the *orientation* estimate. In typical sports player motion, the orientation rapidly changes. To test the filter estimation accuracy during representative motion, the VICON system is used. The VICON Motion Capture system uses infrared cameras to estimate the position of retroflective markers. Motion capture systems are well-known for their application in the filming of animation movies (Figure 4-5), but are also used in technical applications like validation and control in computer vision and robotics [26]. Infrared cameras are set up to cover a workspace. A collection of markers is defined to be an object. As long as the object is in the line of view of at least two VICON cameras, the position of the object can be determined up to a sub-millimeter accuracy [27]. From the positions of each marker, the orientation of the object is determined.

Figure 4-5: Motion Capture systems are well-known for their use in creating animation movies.

4-2-1 Experiment set-up

For the experiment, two JOHAN tracking devices and a set of reflective markers are attached to a rigid plate (Figure 4-6). The plate is mounted on the subject's upper back and the set of markers is defined as an object in the VICON system. Cameras are set up to cover the workspace and the system is calibrated. The subject then performs typical field sports moves, while being tracked by the JOHAN system as well as the VICON system. The JOHAN measurements are fed into the different filter configurations. The resulting orientation estimates are compared to those provided by the VICON system.

(a) VICON camera and JOHAN device. **(b)** Rigid plate with reflective markers.

Figure 4-6: The VICON system uses infrared cameras (4-6a) to capture motion of objects defined by reflective markers (4-6b) in terms of position and orientation.

Representative moves that are performed during the experiment are for example jogging, running, sprinting, jumping, turning and running sideways. The moves are performed within the workspace so that they are captured by the VICON system. The experiment is measured with two tracking devices for reasons of redundancy and consistancy. If one tracker turns out to have failed, the other device serves as a backup. Furthermore, by measuring with two tracking devices the number of trials per experiment is doubled.

The VICON Motion Capture system outputs measurements at 200 Hz, containing a time-stamp t, position p, orientation ϑ in Euler angles and a flag indicating whether or not the object is in view, as denoted in Equation 4-2.

$$z_{\text{VICON}} = \begin{pmatrix} t & p^T & \vartheta^T & \text{flag} \end{pmatrix}^T \tag{4-2}$$

where the time-stamp t is the UNIX time in milliseconds as provided by the computer running the VICON software.

Based on the VICON measurements, a set of reference states is created as is done in Section 3-3.

4-2-2 Synchronize timing

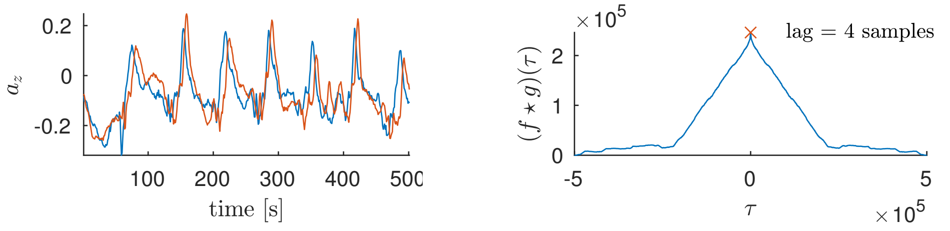
To enable comparison between the JOHAN orientation estimates and the orientations measured by the VICON Motion Capture system, the timing of all measurements has to be synchronized. Both the time-difference between both JOHAN tracking device measurements and the difference between the JOHAN and VICON measurements need to be determined.

JOHAN to JOHAN The first step of synchronizing the timing of both JOHAN tracking devices is done using the GPS time stamp. However, since the GPS measures at 10 Hz, this synchronization lacks accuracy. To find a more accurate time-difference between both JOHAN tracking devices, the measurements are compared. Since both tracking devices are attached to the same rigid plate, it is expected that they measure similar motion. The time-difference is determined by calculating the cross-correlation between measurements of both tracking devices. The cross-correlation of functions f and g is defined as Equation 4-3.

$$(f \star g)[n] \triangleq \sum_{m=-\infty}^{\infty} f^*[m]g[m+n] \tag{4-3}$$

where f^* is the complex conjugate of f and n denotes the lag. The value of n for which the cross-correlation is largest represents the number of samples that one tracking device lags the other.

The signals that are used to calculate the cross-covariation are the acceleration measurements in z-direction, since these are expected to coincide most. In Figure 4-7, the z-accelerations of the two tracking devices and the cross-correlation between both signals are depicted. The lag is determined to be 4 samples, corresponding to 0.04 seconds.



(a) Acceleration measurements in z-direction from both tracking devices.

(b) Cross-correlation between measurements from both tracking devices.

Figure 4-7: Calculating the time-difference between two JOHAN tracking devices. The cross-correlation of acceleration measurements in z-direction from both trackers is calculated. The value for τ with the highest cross-correlation represents the time-difference between the two tracking devices.

JOHAN to VICON To determine the time-difference between the JOHAN tracking devices and the VICON system, the time-stamps are compared. The VICON time-stamp is the UNIX-time in milliseconds as provided by the computer running the VICON software. This time-stamp is compared to the JOHAN GPS-time to synchronize the timing.

For each experiment, the start-time t_0 is defined as the VICON time-stamp corresponding to the first measurement where the object is in view. The end-time is defined as the last VICON measurement with the object in view. To synchronize the VICON measurements with the JOHAN data, the time between two GPS time-stamps is interpolated. The interpolated time-stamp closest to $t_{0,\text{VICON}}$ is chosen as the starting index.

To overcome the difference in sample frequency, the VICON measurements are decimated using the `decimate()`-function over the samples where the object is in view.

4-2-3 Align coordinate frames

JOHAN to JOHAN Although the coordinate frames of both tracking devices do not coincide, it is assumed that they do measure the same. This simplification seems justified, since the center of rotation in human motion typically lies around body's center of gravity, approximately in the low back area of the spinal column [28], and both tracking devices are worn on the upper back.

JOHAN to VICON The position and orientation are provided with respect to a manually defined reference coordinate frame (Ψ_V) that is chosen to be in plane with the earth's surface. The coordinate frame in which VICON denotes the object's orientation and position is defined by the user. The *magic wand*, a tool with known dimensions that the system uses for calibration, is positioned on the floor to define the VICON coordinate frame. The JOHAN world frame is defined based on the direction of the magnetic north. In order to be able to compare VICON measurements with JOHAN measurements, both coordinate systems have to be aligned. Since the x- and y-axes of both $\Psi_W$and Ψ_V lie in plane with the Earth's surface, the difference in orientation is expected to only concern a rotation over the z-axis, as is depicted in Figure 4-8.

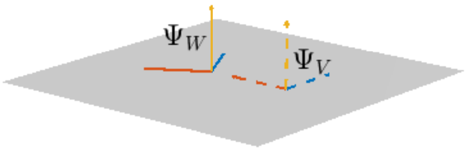
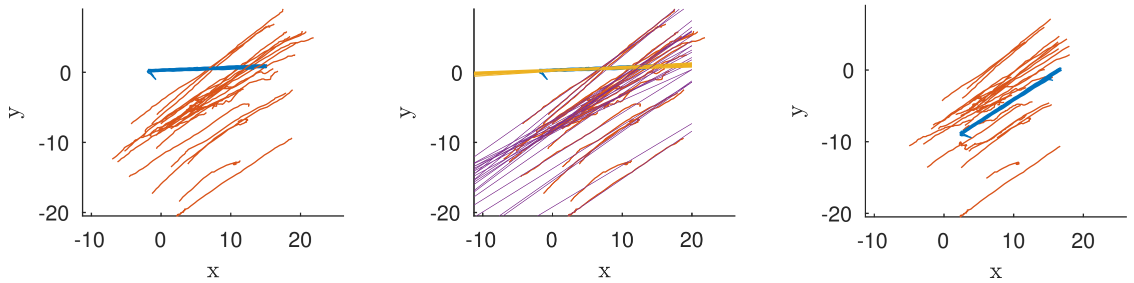


Figure 4-8: The VICON coordinate frame (Ψ_V) needs to be aligned with the world frame (Ψ_W) to enable comparison.

To align the coordinate frames, the positions measured by VICON are compared to the positions measured by JOHAN for several experiments. In the experiments where the subject is walking, jogging, running and sprinting, the path traveled is along an almost straight line, see Figure 4-9a. A line is fit through both the VICON measurements and the JOHAN measurements, to find the misalignment between the two coordinate frames, see Figure 4-9b. Per experiment, the angle between the fitted lines is determined. The mean difference angle is found to be 30.04 degrees.



(a) Position measurements in VICON frame (blue) and JOHAN world frame (red).

(b) Straight lines fitted through VICON (yellow) and JOHAN measurements (purple).

(c) Corrected VICON position measurements.

Figure 4-9: Aligning the Ψ_V and Ψ_W. A straight line is fitted througth positions measured in several experiments. The angle between the positions measured by VICON and the positions measured by JOHAN represents the coordinate frame misalignment and is found to be 30 degrees.

The difference in position of the origin of both coordinate frames is determined next. As is visible in Figure 4-9, sequences of GPS measurements per experiment do not coincide, although the subject did travel the same path, through a 'corridor' of cameras. The paths measured by the GPS lie up to 10 m apart. The reason that this difference is so large lies in the fact that the experiment was performed right next to a tall building.

The next step is to find the displacement between the VICON coordinate frame and the JOHAN world frame. This is done by comparing the positions at t_0 for each experiment. The difference in initial $(x,\ y)$-position is determined for each experiment. The mean displacement at t_0 is assumed to be the displacement between both coordinate frames. The corrected VICON position measurements are displayed in Figure 4-9c.

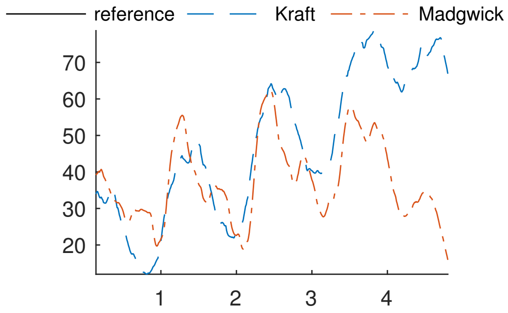
4-2-4 Results

The JOHAN measurements corresponding to the time-sequences measured by the VICON Motion Capture system are fed into the different filter configurations. The estimation accuracy for each different filter configuration is investigated.

In Figure 4-10, the angular errors of the Kraft filter and Madgwick filter are depicted. The errors are very large, up to 70 degrees. Given the aligning process of the coordinate frames, it is questionnable how accurate the reference orientation actually is. Assuming that the orientation errors are correct, poor translational motion estimates can be expected.

The angular RMSE's are given in Table 4-2. The Madgwick filter outperforms the Kraft filter, however, the estimation errors for both filters are unacceptably large.

The position estimates are presented in Figure 4-11. As was expected from the aligning process described in Section 4-2-3, the errors on the position estiamtes are large. This expectation is confirmed, the position errors grow to 4 meters.



(a) Angular error.

Figure 4-10: Angular estimation error on a sprinting experiment. The angular estimation errors of both the Kraft and Madgwick filter are large.

RMSE	θ [degrees]
Kraft	51.44
Madgwick	39.04

Table 4-2: Angular RMSE's in VICON experiment.

The velocity estimates are presented in Figure 4-12. As the figure shows, velocity estimates go crazy. The estimation accuracy could be improved by initializing the filter with the initial reference state $x_{0,\text{ref}}$. Furthermore, if the experiment would have lasted longer, it may be possible that the velocity estimates would still converge to the actual state. This does not seem likely however, given the frantically growing errors in the first 4 seconds.

The acceleration estimates are given in Figure 4-13. A glance at Figure 4-13b learns that the estimation errors are very large. The errors are as large as the range in which the state is expected to be. The poor acceleration estimates are explained by the poor orientation estimates. The

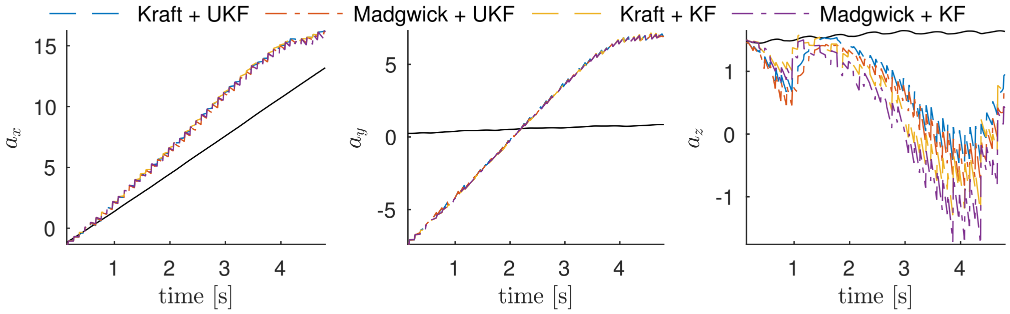
		Position [m]			Velocity [m/s]			Acceleration [m/s^2]		
		x	y	z	x	y	z	x	y	z
Kraft	UKF	2.819	4.479	0.989	6.097	5.010	3.528	6.204	8.371	12.422
	KF	2.672	4.456	1.154	8.635	4.661	4.139	6.234	8.608	12.674
Madgwick	UKF	2.875	4.511	1.306	5.299	4.048	3.532	3.927	4.374	6.735
	KF	2.691	4.493	1.566	9.024	4.084	4.231	4.479	4.386	6.847

Table 4-3: RMSE's of translational motion estimates in the VICON experiment.

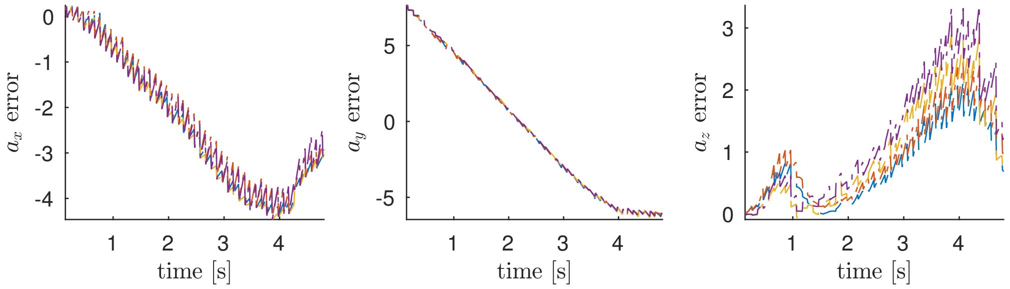
4-2-5 Remarks

The estimation errors in the VICON experiment are huge. The root cause for the large errors can lie in different areas.

The approach to obtaining reference states has its limitations. The reference velocities and accelerations were obtained from position data. This process introduces uncertainties in the reference state. Furthermore, the process of aligning the JOHAN world coordinate frame and



(a) Position estimate.



(b) Errors in the position estimate.

Figure 4-11: Position estimates and corresponding tracking errors on a sprinting motion experiment. As expected from the coordinate frmae aligning process, the position errors are large. This is largely caused by the fact that the experiment was performed next to a tall building.

VICON coordinate frame is quite crude, possibly too crude. It is therefore unsure how accurate the reference state is.

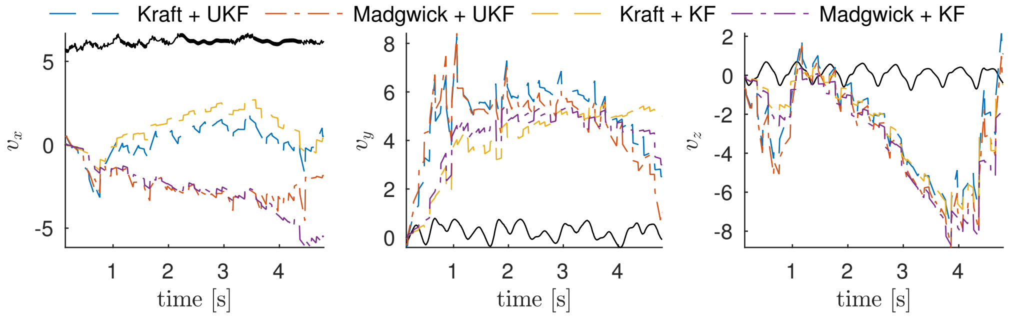
Other possible sources of errors are the sensors in the tracking device. The calibration process of the gyroscope and acceleromter is very simplistic. The biases on the accelerometer after calibration are still in the order of $0.1\ \text{m/s}^2$. Furthermore, the experiment was performed close to a building for reasons of power supply. This leads to disturbances in the GPS measurements larger than usual, as well as disturbances in the magnetometer measurements due to magnetic fields.

Another possible source of error lies in the experiment duration. The time-sequence in which the subject is in view of the camera's is short (±5 seconds), leading to short experiments. The initialization period therefore takes up a relatively large portion of the experiment.

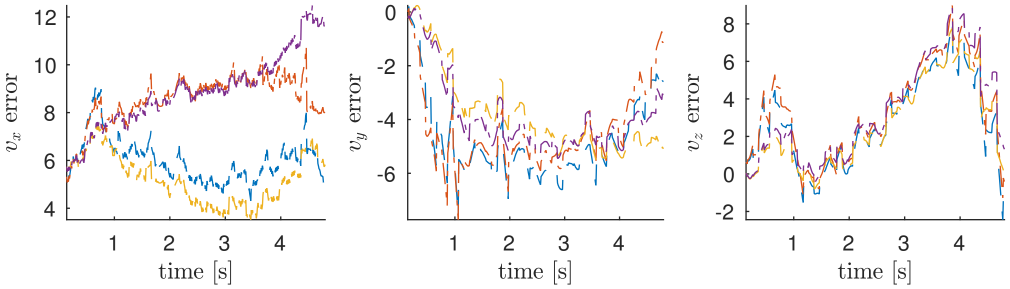
4-3 Summary

The *reference trajectory experiment* has shown that the filtering operations do not improve position estimates. This was expected, since the measurement noise on GPS measurements is not zero-mean and not white due to the on-chip filtering operation.

In the VICON experiment, estimation errors are large. It is however hard to determine where the source of these errors lies. The process in which the reference motion, with which the estimates are compared, is obtained has its limitations. This makes the accuracy of the



(a) Velocity estimate.

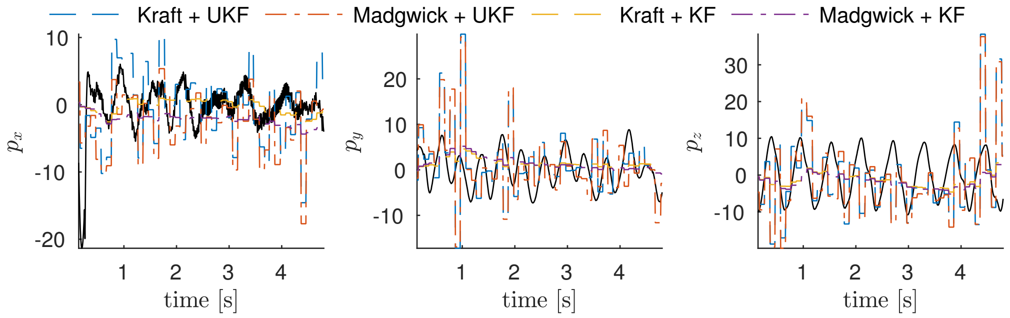


(b) Errors in the velocity estimate.

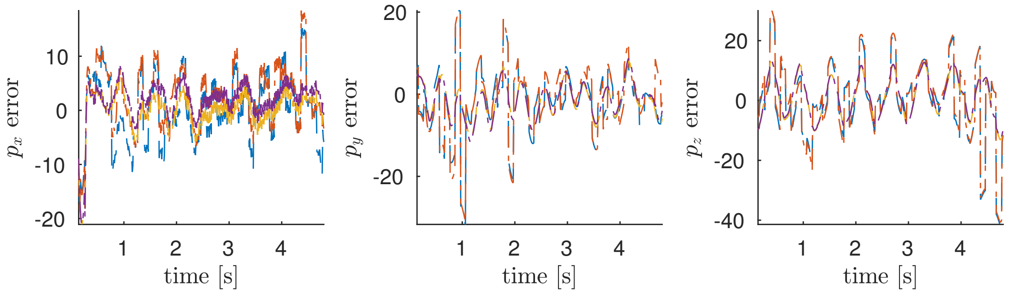
Figure 4-12: Velocity estimates and corresponding tracking errors on a sprinting motion experiment. The velocity estimates are poor. Note that the initial x-velocity lies far from the reference x-velocity. Better initialization could improve the estimation accuracy.

reference states uncertain. Another important possible source of errors is the location of the experiment: right next to a tall building. The presence of the building is likely to disturb the GPS measurements, as well as the magnetometer measurements.

Due to the large uncertainty on the source of errors, no conclusions regarding estimation accuracy are drawn based on the VICON experiment.



(a) Acceleration estimate.



(b) Errors in the acceleration estimate.

Figure 4-13: Acceleration estimates and corresponding tracking errors on a sprinting motion experiment. The estimation errors are large. The poor orientation estimates play a large role in this.

Chapter 5

Conclusions and recommendations

To be able to accurately track sports player motion, different sensor fusion algorithms are compared in this thesis. The Kraft filter and Madgwick filter, combined with either an unscented Kalman filter or a linear Kalman filter are put to the test in both simulations and real-world experiments. Which filter is able to most accurately estimate player motion, accelerations in particular? Which filter has the lowest computational load? Which filter is most suitable to be applied by JOHAN Sports?

In this chapter, the main conclusions from the research are summarized and recommendations for further research are posed.

5-1 Conclusions

To answer the research question, the results of the simulations and real-world experiments are interpretated. The main conclusions of the research are stated below.

- It is concluded that the Madgwick filter, combined with the linear Kalman filter are most suitable for motion tracking in field sports with the current version of the JOHAN Sports tracking device. This conclusion is drawn after interpreting the results of the simulation in which VICON-measured motion is simulated.
  - The Madgwick filter estimates orientation to higher accuracy than the Kraft filter.
  - Estimation accuracy of the unscented Kalman filter and the linear Kalman filter in estimating translational motion is similar.
  - The Madgwick filter has a lower computational load than the Kraft filter.
  - The linear Kalman filter has a lower computational load than the unscented Kalman filter.

- Errors in the orientation estimates are the main factor for deterioration of translational motion estimates. The simulation in Section 3-2 has shown that the lack of p_z measurements is not necessarily a problem. With accurate orientation estimates, accurate world-frame acceleration measurements can be obtained, with which the absence of p_z measurements is corrected for.
- Based on the JOHAN tracking device measurements, the filters that are developed in this thesis are unable to accurately estimate sports player motion.

5-2 Recommendations for further research

From the research performed for this thesis, a number of recommendations for further research follow. The recommendations are categorized per topic: *modeling*, *simulation*, *experiments* and *hardware*.

5-2-1 Modeling

- Estimate gyroscope- and accelerometer biases instead of angular velocities and accelerations. Biases are expected to remain constant over a measurement. This should simplify estimation.
- Research the improvements of feeding estimated accelerations back into the rotational filters. The Kraft filter and Madgwick filter use the accelerometer as a 'gravity' sensor in estimating the orientation. The accelerometer measurements however, contain measurements of translational accelerations as well. The accelerometer measurements can be rid of these translational accelerations by rotating the world-frame acceleration estimates into body-frame and subtracting them from the accelerometer measurements. For this thesis, this has been shortly investigated, but the topic is left to rest after a lack of quick results.

5-2-2 Simulation

The simulations that are performed for this thesis each have their limitations. The simple pure rotations and translations, but also the more complex random-walk simulations do not yield motions that are representative for player motion. Furthermore, the sensor measurements are simulated to only be contaminated with Zero-Mean White Noise. To simulate more realistic measurements, other types of errors could be added.

- Simulate motion as a *bounded random walk* [29]. By defining bounds in which each state has to remain, a motion can be simulated that is more representative for actual player motion.
- Simulate *biases, scaling errors and coupling errors* on the accelerometer and gyroscope measurements. Currently, the accelerometer and gyroscope measurements are simulated to be contaminated with measurement noise only, while in reality they suffer from biases, scale factors and coupling errors. By simulating these as well, a more realistic simulation is obtained.

- Simulate *piecewise constant offsets* for GPS position measurements. Due to the on-chip filtering operation of the GPS receiver, the measurement errors are not ZMWN.

- In the simulations in which motions measured with VICON are imitated, the subject only travels in x*-direction*. Manipulate the simulations in such a way that the direction in which the subject travels is normally distributed over all directions. The estimation errors on positions, velocities and accelerations in x- and y-direction should then be more similar than they are in this thesis.

5-2-3 Experiments

With respect to estimating motion with the real sensor data, it is recommended to further investigate the following:

- Apply a *more involved calibration* technique. In this thesis, a very simple calibration technique is applied. Results are expected to improve with more accurate calibration.

- Perform experiments with *higher grade sensors*. Especially the GPS receiver in the tracking device is problematic. The on-chip filtering operation causes much information to be lost.

- Perform the *pendulum experiment* with tracking devices that contain a magnetometer. For the research, the pendulum experiment was designed and performed. In this experiment, the tracking device is attached to the bob of a pendulum. The pendulum is then swung back and forth, while taking measurements with the JOHAN tracking device. The experiment was performed with outdated hardware, lacking a magnetometer. The measurements are therefore not suitable for the filters described in this thesis.

5-2-4 Hardware

- Perform research with a higher grade GPS receiver. The on-chip filtering operation of the GPS receiver disables the use of tightly coupled filtering algorithms. It is expected that by using a GPS receiver that outputs raw measurements, estimation accuracy can be improved.

- In the IMU, a so-called *Digital Motion Processor (DMP)* is available. This filter runs at 200 Hz on the internal CPU [30], allowing for more accurate orientation estimates. The Digital Motion Processor can also handle magnetometer measurements. Use the on-chip DMP to estimate the orientation.

- Investigate the possibility of *real-time filtering*. Currently, the raw data is stored on the tracking device. The raw data is processed by the filters post-mission, on computers. Ideally, the filters would run on the CPU of the tracking device. Challenges that have to be overcome in order to enable running the filters on the tracking device are:
    - Calibration of the magnetometer.

- Initialization procedure. The world-frame is defined by the first GPS measurement (origin) and the magnetometer measurement at that time instant (magnetic North, x-direction). For running the filter on-chip, a more robust method should be devised.

Appendix A

Attitude representations

The attitude of a 3D rigid body in space can be represented in different ways. A few commonly used representations will be addressed in this section.

A-1 Euler angles

A well-known orientation representation is called *Euler angles*. It expresses a rotation in three angles, often referred to as *pitch*, *roll* and *yaw*, denoted by ϕ, θ and ψ. These angles represent three consecutive rotations about the axes of the coordinate frame, for example z-x-z': First rotate ϕ radians about the z-axis, then rotate θ radians about the x-axis and finally rotate ψ radians about the new z-axis. The problem of this parametrization is that it has singularities at pitch values of $\theta = \frac{\pi}{2} + k\pi$, with $k \in \mathcal{Z}$. To overcome these singularities, other attitude representations have been developed. [31]

A-2 Direction Cosine Matrix

As is described in [32], the Direction Cosine Matrix (DCM) is a square matrix, whose columns are the orthogonal unit vectors of one frame expressed in the other frame.

$$ {}^{A}_{B}R = \begin{pmatrix} {}^{A}_{B}u_1 & {}^{A}_{B}u_2 & {}^{A}_{B}u_3 \end{pmatrix} \tag{A-1}$$

where ${}^{A}_{B}u_i$ is the unit vector along B frame axis i projected on coordinate frame A axis. From orthogonality it follows that ${}^{A}_{B}R^T = {}^{A}_{B}R^{-1}$. A vector v expressed in frame A equals the product of ${}^{A}_{B}R$ with the same vector expressed in frame B (and vice versa with ${}^{A}_{B}R^T$).

$$ {}^{A}v = {}^{A}_{B}R \; {}^{B}v, \qquad {}^{B}v = {}^{A}_{B}R^T \; {}^{A}v = {}^{B}_{A}R \; {}^{A}v \tag{A-2}$$

Using the chain rule, multiple rotations can be expressed in one direction cosine matrix by:

$$ {}^{A}_{C}R = {}^{A}_{B}R \, {}^{B}_{C}R \tag{A-3}$$

A-3 Quaternion

Another commonly used attitude representation is the quaternion [31]. This representation is particularly popular because of its lack of singularities. A quaternion can be represented as a vector $q \in \mathbb{H}$:

$$q = \begin{pmatrix} q_w \\ q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} q_w \\ q_{1:3} \end{pmatrix} \tag{A-4}$$

The adjoint ($\bar{q}$), norm ($||q||$) and inverse (q^{-1}) of quaternion q are given by:

$$\bar{q} = \begin{pmatrix} q_w \\ -q_{1:3} \end{pmatrix} \tag{A-5}$$

$$||q|| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \tag{A-6}$$

$$q^{-1} = \frac{\hat{q}}{||q||}. \tag{A-7}$$

Quaternions are multiplied in a special, non-commutative way. Let q and p be quaternions, they are multiplied as:

$$q \cdot p = \begin{pmatrix} q_w p_0 - q_{1:3}^T p_{1:3} \\ q_w p_{1:3} + p_0 q_{1:3} - q_{1:3} \times p_{1:3} \end{pmatrix} \tag{A-8}$$

$$= \begin{pmatrix} q_w & -q_{1:3}^T \\ q_{1:3} & q_w I_3 - C(q_{1:3}) \end{pmatrix} \begin{pmatrix} p_0 \\ p_{1:3} \end{pmatrix} = Q(q) \cdot p \tag{A-9}$$

$$= \begin{pmatrix} p_0 & -p_{1:3}^T \\ p_{1:3} & p_0 I_3 + C(p_{1:3}) \end{pmatrix} \begin{pmatrix} q_w \\ q_{1:3} \end{pmatrix} = \bar{Q}(p) \cdot q \tag{A-10}$$

where $Q(q)$ is the *quaternion matrix* function of q, $\bar{Q}(p)$ is the *conjugate quaternion matrix* function of p and C is defined as the skew-symmetric *cross product matrix* function $C : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ (similar to $(e\times)$ in Equation A-11):

$$\begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}, \tag{A-11}$$

Written in full, the quaternion matrix $Q(q)$ looks like Equation A-12.

$$\begin{pmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{pmatrix} \tag{A-12}$$

A quaternion can be written as a Direction Cosine Matrix as Equation A-13.

$$R_q(q) = \begin{pmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2q_xq_y + 2q_wq_z & 2q_xq_z - 2q_wq_y \\ 2q_xq_y - 2q_wq_z & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2q_yq_z + 2q_wq_x \\ 2q_xq_z + 2q_wq_y & 2q_yq_z - 2q_wq_x & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix} \tag{A-13}$$

Appendix B

Kalman filter

The reader is expected to be familiar with the traditional Kalman filter []. For a quick refresher, a short summary is given here.

Assume the initial state estimate to be x_0 and the initial covariance estimate P_{x_0}. Furthermore, assume a propagation model as given in Equation B-1 and an observation model as Equation B-2.

$$x_k = Fx_{k-1} + Bu_k + w_k \tag{B-1}$$

$$y_k = Hx_k + v_k, \tag{B-2}$$

with:

F	state transition matrix
B	control input model
H	observation model
w	process noise
v	measurement noise

In the prediction step, the *a priori* state estimate is obtained by applying the propagation model. The a priori covariance is calculated via Equation B-4.

$$x_k^- = Fx_{k-1} + Bu_k \tag{B-3}$$

$$P_k^- = FP_{k-1}F^T + Q, \tag{B-4}$$

where Q is the covariance matrix of the process noise.

Subsequently, the measurement update is performed. The innovation is calculated by subtracting the expected measurements from the actual measurements in Equation B-5. From the innovation, the innovation covariance S_k is calculated via Equation B-6. The Kalman gain is calculated by Equation B-7. Using the Kalman gain, the a posteriori state estimate and covariance are calculated via Equation B-8 and Equation B-9.

$$y_k = z_k - Hx_k^- \tag{B-5}$$

$$S_k = HP_k^- H^T + R \tag{B-6}$$

$$K = P_k^- H^T S^{-1} \tag{B-7}$$

$$x_k = x_k^- + Ky_k \tag{B-8}$$

$$P_k = (I - KH)P_k^- \tag{B-9}$$

with:

y_k	innovation vector
z_k	measurement vector
S_k	innovation covariance
R	measurement noise covariance
K	Kalman gain

Appendix C

Numerical example of the unscented transformation

To explain the working principle of the unscented transformation, assume the fictitious model in Equation C-1. One step of the unscented transformation is performed based on this model with initial state $x_0 = \begin{pmatrix} 2 & 1 & 1 \end{pmatrix}^T$ and initial covariance matrix $P = 1/4\ I_3$. The transformation is depicted in Figure C-1.

$$f(x) = \begin{pmatrix} x_1^2 \\ x_1 + x_2 \\ x_3 + \frac{1}{2} \end{pmatrix} \tag{C-1}$$

First, sigma points $\mathcal{X}$ are chosen around the initial state estimate, distributed in accordance with P_x:

$$\mathcal{X} = \left[x - \sqrt{P_x}, \quad x, \quad x + \sqrt{P_x} \right] \tag{C-2}$$

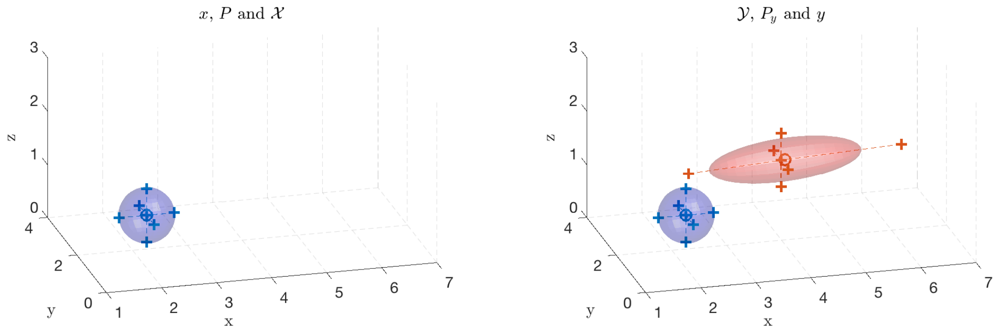
$$= \left[\begin{pmatrix} 1.5 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 0.5 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2.5 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1.5 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 1.5 \end{pmatrix} \right] \tag{C-3}$$

Each individual sigma point is then propagated through the prediction model:

$$\mathcal{Y} = f(\mathcal{X}) = \left[\begin{pmatrix} 2.25 \\ 2.5 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 4 \\ 2.5 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 6.25 \\ 3.5 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 4 \\ 3.5 \\ 1.5 \end{pmatrix}, \begin{pmatrix} 4 \\ 3 \\ 2 \end{pmatrix} \right] \tag{C-4}$$

The transformed state estimate is calculated by taking the mean of all transformed sigma points:

$$y = \frac{1}{7} \sum_{i=1}^{7} \mathcal{Y} = \begin{pmatrix} 4.07 \\ 3 \\ 1.5 \end{pmatrix} \tag{C-5}$$



(a) Sigma points

(b) Transformed state estimate

Figure C-1: The working principle of the unscented transformation. Assume an initial state x (blue 'o') and covariance P (blue ellipsoid). Based on the covariance, sigma points (blue plus signs) are chosen (Figure C-1a). Each individual sigma point is propagated through $f(\ )$ (red plus signs). From the transformed sigma points, the new state (red 'o') and covariance (red ellipsoid) are determined (Figure C-1b).

Appendix D

Tuning

Each of the filters is tuned using an evolutionary tuning technique described in Section D-1. The resulting tuning parameters are presented Sections D-2 to D-5.

D-1 Method

To optimize estimation accuracy, tuning is required. The filters are tuned on the simulated measurements and reference states derived from the VICON experiments (Section 3-3). To prevent favoring one type of motion over the other, three experiments of each motion type (walking, running, etc.) are used as reference data. To reduce complexity, the Kraft filter, Madgwick filter, UKF and KF are tuned independently. For tuning the translational filters (UKF and KF), the reference quaternions are used to rotate the accelerometer measurements into world frame, imitating a perfect orientation estimation.

For tuning of filters, an evolutionary approach is chosen. For each tuning parameter, an initial value is chosen and stored in the vector p_0. Mutations of the initial values p^* are added to the set of tuning parameters resulting in a set of *children* Ω, like Equation D-1.

$$\Omega = [p_0, p_1^*, \ldots, p_n^*] \tag{D-1}$$

$$p_i^* = p_0 \cdot (1 + \eta \cdot \Xi_i), \quad \text{for } i = 1, 2, \ldots n. \tag{D-2}$$

The mutations arise according to Equation D-2, where $\eta < 1$ is a tuning parameter and Ξ_i is a vector of random numbers between $[-1, 1]$.

One by one, the filter is run, using a different *child* as tuning parameters. The most successful child, the vector of tuning parameters that yields the smallest RMSE, is used for reproduction. It becomes the new p_0. The process is repeated until the RMSE stops decreasing, or the maximum number of iterations is reached. The resulting tuning parameters are a local optimum.

D-2 Tuning the Kraft filter

The tuning matrices for the Kraft filter are $P_{0,\text{base}}$, Q_{base} and R_{base}. Their initial values are chosen based on measured variances in the sensors. Then the tuning algorithm is run. The RMSE that is minimized in this case is the angular RMSE as defined by Equation 3-7.

After 6 iterations, the optimal angular RMSE does not decrease anymore. The decrease in angular RMSE is depicted in Figure D-1. The resulting parameters for P_0, Q and R are provided in Equations D-3 to D-5.

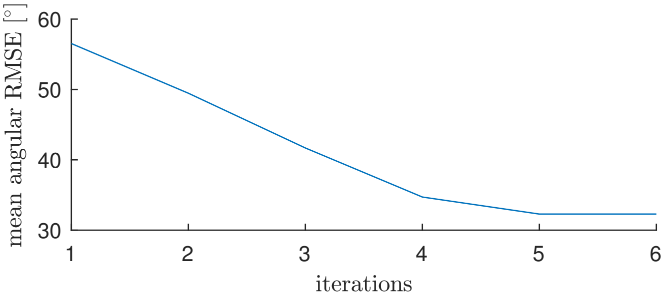


Figure D-1: Decrease in angular RMSE of the orientation estimation by the Kraft filter over subsequent iterations of the tuning algorithm.

Note that the mean angular RMSE with optimal tuning matrices is still as large as 30 degrees. The Kraft filter is now tuned for optimal performance on the VICON-based simulated motion. For use on real data, the filter needs retuning.

$$P_0 = \begin{pmatrix} 0.0012 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0004 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0007 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0071 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0051 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0647 \end{pmatrix} \tag{D-3}$$

$$Q = \begin{pmatrix} 0.0005 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0020 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0081 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0283 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0414 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7786 \end{pmatrix} \tag{D-4}$$

$$R = \begin{pmatrix} 0.0310 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0340 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2651 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8798 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.7391 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.0131 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0018 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0045 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix} \tag{D-5}$$

D-3 Tuning the Madgwick filter

The Madgwick filter is tuned in similar fashion as the Kraft filter. However, for the Madgwick filter only one tuning parameter has to be optimized. After 3 iterations, the tuning algorithm finds the locally optimal value of $\beta = 0.4357$. The decrease in RMSE per iteration is depicted in Figure D-2.

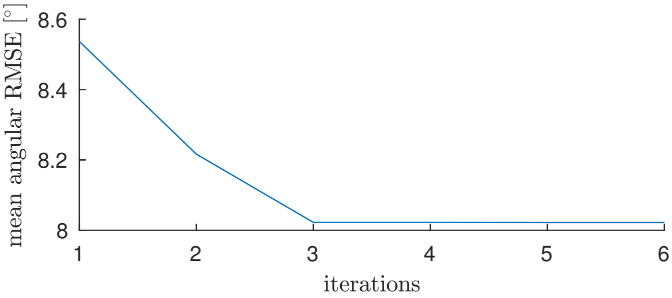


Figure D-2: Decrease in angular RMSE of the orientation estimation by the Madgwick filter over subsequent iterations of the tuning algorithm.

D-4 Tuning the Unscented Kalman filter

For the tuning of the UKF, the same evolutionary algorithm approach is used. Like in the Kraft filter, matrices P_0, Q and R contain the tuning parameters.

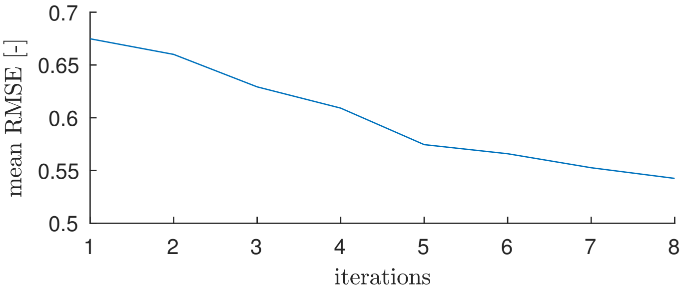


Figure D-3: Decrease in the mean RMSE in the translational states estimated by the UKF over subsequent iterations of the tuning algorithm.

The tuning parameters found by the tuning algorithm are presented in Equations D-6 to D-8.

$$P_0 = \begin{pmatrix} 0.0447 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1628 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0033 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0176 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0049 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0067 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0008 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0003 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0042 \end{pmatrix} \tag{D-6}$$

$$Q = \begin{pmatrix} 0.0188 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0190 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0200 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0010 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0006 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0014 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0002 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0007 \end{pmatrix} \tag{D-7}$$

$$R = \begin{pmatrix} 0.0714 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0197 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.1809 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0566 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0004 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0002 \end{pmatrix} \tag{D-8}$$

D-5 Tuning the linear Kalman filter

The linear Kalman filter is tuned in similar fashion as the Kraft filter and the UKF. The RMSE's per iteration are depicted in Figure D-4. The resulting tuning parameters are provided in Equations D-9 to D-11.

$$P_0 = \begin{pmatrix} 0.1133 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0886 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0065 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0127 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0023 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0019 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0002 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0005 \end{pmatrix} \tag{D-9}$$

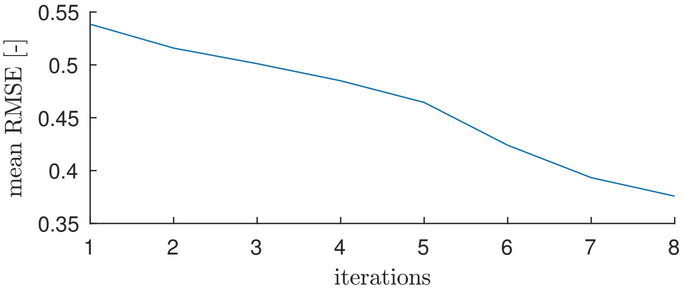


Figure D-4: Decrease in the mean RMSE in the translational states estimated by the linear Kalman filter over subsequent iterations of the tuning algorithm.

$$Q = \begin{pmatrix} 0.0025 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0136 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0035 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0018 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0005 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0031 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0003 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0002 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0010 \end{pmatrix} \tag{D-10}$$

$$R = \begin{pmatrix} 0.1454 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0981 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.6261 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0003 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0004 \end{pmatrix} \tag{D-11}$$

Appendix E

Recognizing active playing blocks

During measurements, the JOHAN Sports tracking device stores all raw data on a memory card. The filtering operation is performed afterwards, called post-mission processing. This enables perfroming a data pre-processing action on the complete dataset before filtering.

Only use data corresponding to playing blocks For generating a training advice for sports coaches it is desired to only take into account data of the time periods where players actually play. The motion of a player heading to the dressing room during the break or sitting on the bench after a substitution are not of interest. Selecting the active playing blocks is currently done manually by the user. With knowledge of the start- and endtime of a match and knowledge on substitutes, it is easy to select active playing blocks. However, it is time-consuming and not user friendly. Therefore it is desired to automatically recognize active playing blocks.

Drop erroneous GNSS measurements Another reason why it is desired to pre-process measurement data is the following. Measurements of the JOHAN tracking device are often performed in the proximity of buildings, for example large steel or concrete soccer stadiums. These buildings can cause temporary or complete loss of GNSS reception. When GNSS reception is lost, the chip outputs erroneous data. It is desired to drop erroneous datapoints, since they contaminate the filter output. Erroneous GNSS measurements can sometimes be easily recognized. Speed measurements can rise to values unattainable for human and positions can drift away far from the playing field. However, when the measured speed and position lie within normal bounds, it is hard to tell if a GNSS measurement is correct or erroneous.

It happens to be the case that erroneous GNSS points are often measured during the break or when a player is on the bench. Hence, being able to correctly recognize active vs. inactive playing blocks can kill two birds with one stone.

PlayerLoad$^{\text{TM}}$ A measure that indicates the intensity of a match or training session is called the *PlayerLoad* [33]. The PlayerLoad is calculated from 100Hz accelerometer measurements and is defined by Equation E-1.

$$\text{PlayerLoad}^{\text{TM}} = \frac{\sqrt{\Delta a_x^2 + \Delta a_y^2 + \Delta a_z^2}}{\Delta t} \tag{E-1}$$

It is expected that PlayerLoad is high only when a player is actively playing. If this is the case, it is should be possible to distinguish between active and inactive playing blocks by filtering the PlayerLoad.

E-1 Active play

When a player is actively playing, he is expected to have a high PlayerLoad. An example of the PlayerLoad during a real hockey match is depicted in Figure E-1. The red signal indicates where the PlayerLoad reaches above a threshold t_1.

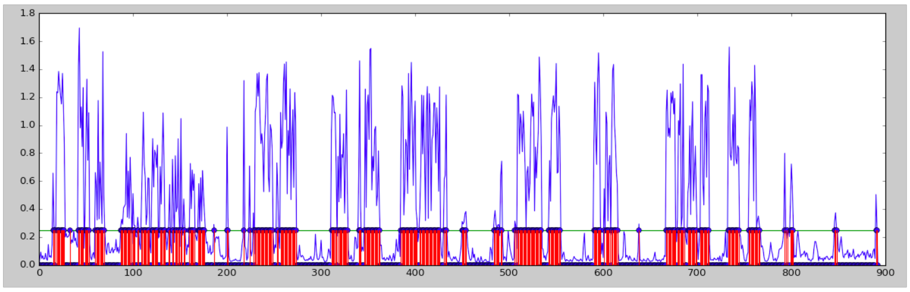


Figure E-1: PlayerLoad during a hockey match. The red signal indicates the time instances where the PlayerLoad is above threshold t_1.

A player on the field is not necessarily moving all the time. For example during set plays like free kicks or corners, players stop moving for an instance. The PlayerLoad may then drop under the threshold.

E-2 Playing blocks

For the data analysis, it is desired to take into account the complete period of time a player is on the field. Matches with a lot of set plays are expected to be less intense than matches with little set plays, influencing the total match intensity.

With knowledge on the start- and entime of each half and knowledge on substitutes, a user can define the playing blocks manually. The manually defined playing blocks corresponding to the depicted PlayerLoad are depicted in Figure E-2.

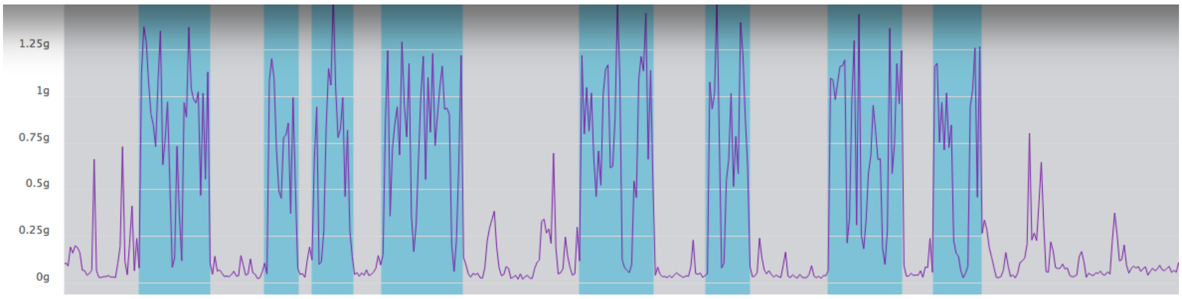


Figure E-2: User-defined playing blocks. The purple line indicates the PlayerLoad, the blue blocks correspond to the active playing blocks. Note that in field hockey the number of substitutions is unlimited.

E-3 Rolling mean filter

A simple threshold as proposed in section E-1 will not suffice to recognize playing blocks. However, following from the assumption that PlayerLoad is generally high inside playing blocks and low outside, it is expected that it is also possible to recognize playing blocks from the PlayerLoad data. The filter design is described here.

First, a binary signal indicating active play is derived from the PlayerLoad. The binary signal returns a 1 for a PlayerLoad above threshold t_1. Next, a rolling mean filter with windowsize ws is applied to the active play signal. The window mean is placed at the middle position of the window. Finally, a new binary signal is defined to be 1 where the rolling mean is above threshold t_2 and 0 where it is below that threshold. This signal indicates playing blocks.

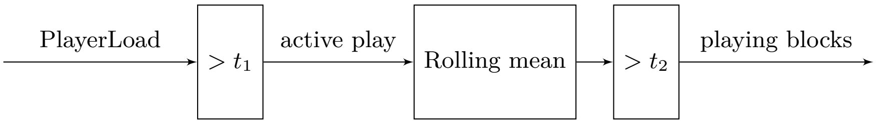


Figure E-3: Recognizing playing blocks from PlayerLoad.

E-4 Tuning

The three tuning parameters in the filter are thresholds t_1 and t_2 and the window size ws. The above-described filter was tuned on one single dataset, corresponding to a field hockey match. It is possible that for soccer or for other field hockey matches, different tuning parameters are required. The approach to tuning these parameters is described below.

A large number of datasets exist for both soccer and field hockey matches. In each of these datasets, the active playing blocks are manually defined. Each datapoint is labeled either as active (1) or as inactive (0). The datasets are split into sets corresponding to soccer and sets corresponding to field hockey. Then for each sport, the sets are split into a training set (with 70% of all matches) and a test set (30% of all matches). With different values for tuning parameters t_1, t_2 and ws, the playing blocks are computed for the training set. The classification error is defined as

$$e = \frac{1}{n} \sum_{i=1}^{n} |y_{user,i} - y_{filter,i}| \tag{E-2}$$

With the tuning parameters yielding the smallest classification error, the matches belonging to the test set are classified. The classification error of the test set is the reported classification error.

The algorithm will be implemented such that it presents *suggested playing blocks* to users. Users will still be able to edit the playing blocks to their liking. In this way, more user-labeled data will accumulate over time. The tuning parameters can then be fine-tuned periodically to increase classification accuracy.

Bibliography

[1] J. H. Stubbe, A. M. M. C. Van Beijsterveldt, S. Van Der Knaap, J. Stege, E. A. Verhagen, W. Van Mechelen, and F. J. G. Backx, "Injuries in professional male soccer players in the Netherlands: A prospective cohort study," *Journal of Athletic Training*, vol. 50, no. 2, pp. 211–216, 2015.

[2] R. Bahr and T. Krosshaug, "Understanding injury mechanisms: a key component of preventing injuries in sport.," *British journal of sports medicine*, vol. 39, no. 6, pp. 324–9, 2005.

[3] G. Howatson and A. Milak, "Exercise-induced muscle damage following a bout of sport specific repeated sprints," *Journal of Strength and Conditioning Research*, vol. 23, no. 8, pp. 2419–2424, 2009.

[4] H. W. Sorenson, *Kalman filtering: theory and application.* IEEE, 1985.

[5] R. van der Merwe, E. Wan, and S. J. Julier, "Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, no. August, pp. 1–30, 2004.

[6] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *Int. symp. aerospace/defense sensing, simul. and controls*, 1997.

[7] J. Wendel, A. Maier, J. Metzger, and G. F. Trommer, "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS / INS Integration Improved Vehicle Navigation Using Aiding with Tightly Coupled Integration," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, no. August, pp. 1–15, 2005.

[8] N. El-Sheimy, E.-H. Shin, and X. Niu, "Kalman Filter Face-Off: Extended vs Unscented Kalman Filters for Integrated GPS and MEMS Inertial," *Inside GNSS*, pp. 48–54, 2006.

[9] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *IEEE International Conference on Rehabilitation Robotics*, pp. 179–185, 2011.

[10] J. Craig, *Introduction to Robotics Mechanics and Control.* Pearson Education International, 2005.

[11] W. Smart, *Textbook on Spherical Astronomy.* Cambridge University Press, 6 ed., 1977.

[12] N. Yazdi, F. Ayazi, and K. Najafi, "Micromachined inertial sensors," *Proceedings of the IEEE*, vol. 86, no. 8, pp. 1640–1658, 1998.

[13] D. Tedaldi, A. Pretto, and E. Menegatti, "A robust and easy to implement method for IMU calibration without external equipments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3042–3049, 2014.

[14] I. Skog and H. Peter, "Calibration of a MEMS inertial measurement unit," *International Measurement Confederation*, pp. 1–6, 2006.

[15] J. Lenz and S. Edelstein, "Magnetic sensors and their applications," *IEEE Sensors Journal*, vol. 6, no. 3, pp. 631–649, 2006.

[16] G. Dudek and M. Jenkin, "Inertial Sensors, GPS, and Odometry," *Springer Handbook of Robotics*, pp. 477–490, 2008.

[17] L. Gauthier, P. Michel, J. Ventura-Traveset, and J. Benedicto, "EGNOS: the first step in Europe's contribution to the global navigation satellite system," *ESA bulletin*, vol. 105, no. June, pp. 35–42, 2001.

[18] ESA, "Monthly Performance Report November 2016," no. November, pp. 1–47, 2016.

[19] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems 1," vol. 82, no. Series D, pp. 35–45, 1960.

[20] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems.* CRC press, 2011.

[21] E. Kraft, "A quaternion-based unscented Kalman filter for orientation tracking," *Proceedings of the 6th International Conference on Information Fusion, FUSION 2003*, vol. 1, pp. 47–54, 2003.

[22] M. Vernacchia, "Unscented Kalman Filter for 3D Attitude Estimation 16 . 322 Final Project,"

[23] N. Delleman, C. Haslegrave, and D. Chaffin, *Working Postures and Movements.* Ergonomics and human factors, Taylor & Francis, 2004.

[24] K. Deb, K. Sindhya, and J. Hakanen, "Multi-objective optimization," in *Decision Sciences: Theory and Practice*, pp. 145–184, CRC Press, 2016.

[25] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures.," *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[26] www.VICON.com, "Vicon official website," aug 2016.

[27] M. Windolf, N. Götzen, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems-exemplified on the Vicon-460 system," *Journal of Biomechanics*, vol. 41, no. 12, pp. 2776–2780, 2008.

[28] R. Behnke, *Kinetic Anatomy.* Human Kinetics, 2012.

[29] J. Nicolau, “Stationary Processes That Look like Random Walks: The Bounded Random Walk Process in Discrete and Continuous Time,” *Econometric Theory*, vol. 18, no. 01, pp. 99–118, 2002.

[30] I. Inc., “MPU-6000 and MPU-6050 Product Specification,” *InvenSense Inc.*, vol. 1, no. 408, pp. 1–57, 2013.

[31] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, pp. 1–35, 2006.

[32] P. G. Savage, “Algorithm Design Part 1 : Attitude Algorithms,” *Design*, vol. 21, no. 1, pp. 19–28, 1998.

[33] L. J. Boyd, K. Ball, and R. J. Aughey, “The Reliability of MinimaxX Accelerometers for Measuring Physical Activity in Australian Football,” pp. 311–321, 2011.

Glossary

List of Acronyms

3mE	Mechanical, Maritime and Materials Engineering
AHRS	Attitude and Heading Reference System
DCM	Direction Cosine Matrix
DoF	Degrees of Freedom
DCSC	Delft Center for Systems and Control
ECEF	Earth-Centered Earth-Fixed
EGNOS	European Geostationary Navigation Overlay Service
EKF	Extended Kalman filter
ESA	European Space Agency
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
MEMS	Microelectromechanical system
MTJ	Magnetic Tunnel Junction
SBAS	Satellite-Based Augmentation System
UKF	Unscented Kalman filter
ZMWN	Zero-Mean White Noise

List of Symbols

J	Jacobian matrix
L	Number of sigma points
P	State covariance matrix
Q	Process noise covariance matrix
R	Measurement noise covariance matrix
R	Rotation matrix
T	Sample time
Φ	Magnetic flux
α	Tuning parameter
κ	Tuning parameter
λ	Longitude
$\mathcal{X}$	Set of sigma points around x
$\mathcal{Y}$	Set of sigma points around y
μ	Step size in gradient descent
∇f	Gradient of f
ω	Angular velocity
ω_{bias}	Gyroscope bias
ϕ	Latitude
θ	Bearing
a	Acceleration
a_{bias}	Accelerometer bias
b	Earth's magnetic field
f	Propagation model
g	Gravity
h	Observation model
k	Time-step
m	Magnetometer measurement
p	Position
q	Quaternion
$q^\star$	Conjugate quaternion
u	External input vector
v	Measurement noise vector
v	Velocity
w	Process noise vector
x	State vector
y	Expected observation vector
Ψ_B	Body frame
Ψ_E	Earth frame
Ψ_F	Field frame

Ψ_W	World frame
d	Great circle distance
r	Radius of a sphere