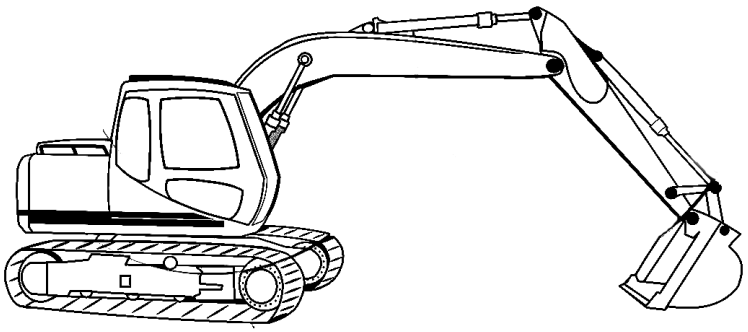
CHALMERS



Orientation estimation of a rigid multi body system using accelerometers, gyroscopes and the geometry

Master's Thesis in System, control and mechatronics

JOHAN GUSTAFSSON

Department of Signal and Systems
Division of signal processing
Chalmers University of Technology
Gothenburg, Sweden 2014
Master's Thesis EX035/2014

ACKNOWLEDGEMENTS

I would like to extend my greatest gratitude to my supervisor, Peter Forsberg at CPAC Systems AB, for giving me his attention and guidance in times of need.

I would also like to address many thanks to my supervisor at Chalmers, Lennart Svensson, for spending time, and discussing the use of Bayesian statistics in estimation theory.

Thank you!

Johan Gustafsson

Gothenburg, June 11, 2014

ABSTRACT

The orientation and movement of an excavator arm needs to be estimated by the use of accelerometers and gyroscopes positioned along the arm. Knowing this information allows the development of multiple new applications for heavy construction equipment, such as load indication of the mass in the bucket and map integration using GPS. Further, the arm may be used to measure the construction site and this time consuming task could therefore be simplified. The performance of two different distributed filter implementation, extended Kalman filter and unscented Kalman filter are evaluated in terms of roll and pitch, angular velocity and acceleration estimates. The implementations represent orientations using quaternions and make use of the geometry of the excavator to achieve a better estimate of the system.

The applicability of the proposed algorithm has been thoroughly demonstrated on a representative model. Similar performance are seen in both implementations, and both filter estimate the system parameters within the range of usage. For the last inertial measurement unit in line, mounted on the bucket of the excavator with a total arm length of more than 10 m, the root mean square error is as low as 0.63°and 0.05°angles for pitch and roll respectively during simulated working conditions.

However, the implementations differs in computational load as the motion model of the extended Kalman filter make use of a small angle approximation and because of the different methods used in propagating the mean and covariance in the two filter implementations. For implementation on embedded systems the extended Kalman filter using small angle approximation is recommended for its ease of use and low computational cost.

Contents

List of Tables

List of Figures

Acronyms

AHRS Attitude and Heading Reference System. 2, 5

CAN controller area network. ix, 3, 9, 12, 13, 31, 32
CF complementary filter. xiii, 6, 7
CPU central processing unit. 12

DH Denavit Hartenberg. 16, 17, 26

ECEF earth-centered earth-fixed. 13
EKF extended Kalman filter. iii, xi, 3, 5–7, 21, 22, 33, 38, 39, 44–47, 49, 53
ENU east north up. 13, 27–29, 34, 38, 41, 49

IMU inertial measurement unit. iii, xi, 1–3, 5, 7, 9, 31, 33–35, 39, 46–49, 52, 53
INS inertial navigation system. 10, 11

KF Kalman filter. iii, xi, xiii, 3, 18–22, 24, 33, 36, 53

LTP local tangent plane. 13, 27

MEMS microelectromechanical systems. 5, 6, 9–12, 28

NED north east down. 13

PAE peak absolute error. 46, 47, 49
PDF probability density function. 2, 18

RMSE root mean square error. 5–7, 46, 47, 49

TCF time varying complementary filter. 6

UKF unscented Kalman filter. iii, xi, 3, 7, 21–23, 33, 36, 44–47, 49, 51, 53

1

Introduction

In the process of automating heavy machine equipment, there is a need to achieve a better description of the system in real time. This thesis evaluates different distributed filtering implementations on an excavator model, based a system of inertial measurement units mounted on the machine and along the arm.

1.1 Background and motivation

Heavy construction equipment becomes more integrated with automation control, allowing easier operation of the machine and even fully controlled equipment. This requires more information about the state of the machine, such as the accelerations, velocities and positions of an excavator arm. With the knowledge of the position and orientation of the machine, the information can be incorporated with maps to simplify the operation and give information about critical areas in the surrounding such as pipelines and electrical lines.

Machine control projects aiming in automating certain tasks of the machine may be developed using the information about the state of the machine. Functionality such as weight estimation is possible by knowing the pressure in the cylinders and the acceleration, velocity and position of the excavator arm.

A system of inertial measurement units (IMUs) mounted on an excavator are used to estimate the position, velocity and acceleration of the excavator arm. The absolute position may be measured with GPS to a fixed point on the machine, and the relative position of the specific part is calculated by translating over its own geometry. This method required knowledge about the orientation and position of interconnecting parts, that may be measured by IMUs.

An IMU consist of an accelerometer that measures the experienced acceleration and a gyroscope which gives output about the observed rotational velocity. Together these two sensors measure 6 degrees of freedom, which can fully express the orientation of the unit.

The accelerometer and gyroscope have different uncertainties, which makes it difficult to trust any of the sensors. All types of accelerations are detected by the accelerometer, therefore difficulties arises in estimating the orientation relative gravity during motion. The rate gyroscope have some zero-offset that changes over time, expressing rotations even at rest. Other factors such as temperature may also affect the sensors. Estimation theory seeks to fuse the sensors to reduce the uncertainties and get a better estimate of the orientation. On a single machine, there may be between 5 to 7 IMUs connected in a network. These can work independently, or in a network to achieve the satisfied results.

1.2 Problem statement

The current Attitude and Heading Reference System (AHRS) does not deliver estimates of the state of the system accurate to the degree that is needed. Estimation of the orientation, angular velocity and angular acceleration needs to be improved to reach the aspired precision of the developed applications.

1.3 Aims and objectives

The objective is to develop an algorithm for sensor fusion that will increase the orientation estimate of each node of an array of IMUs. Accurate estimates of the angular velocity and angular acceleration of each IMU needs to be delivered. The concept is to be implemented on the existing hardware, and benchmarked against current Attitude and Heading Reference System (AHRS) algorithm.

The aim of the fusion algorithm is to estimate the orientation, velocity and acceleration of each node accurately, and the position of the end node using the knowledge of the system of linkage better than the current AHRS.

1.4 Concept overview

The filter implementation are based on Bayesian statistics, that aim to construct the probability density function (PDF) of the state estimates by using all available information. The system of IMUs communicate with each other to make the system adaptive to changes in acceleration and velocity, and give a more accurate estimate of the orientation of the system. By mounting the system of IMUs in the plane of an excavator arm and using the information of linkage system, the IMUs use the information of other IMUs to estimate their orientation, velocity and acceleration. Mounting the IMUs in this way results in the same roll of all IMUs. Further, using the estimates of the angular acceleration and angular velocity allows the IMUs to predict the experienced tangential and centrifugal acceleration of other IMUs using the length of link in the rigid body system.

A figure of a part of the system is shown in Figure 1.1. An IMU is mounted on the beginning of the stick, in such a way that it will have the same roll as the other IMUs mounted on the excavator arm and be able to predict the acceleration of the IMU

mounted on the bucket. Using the knowledge of the length of the stick l_i, the IMU mounted on the stick may predict the experienced acceleration of the IMU mounted on the bucket.

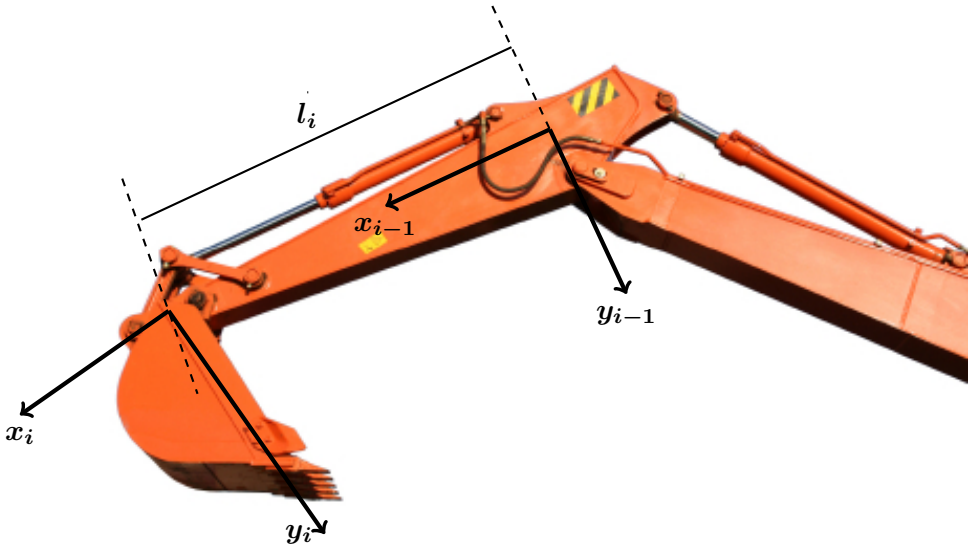


Figure 1.1: An excavator arm with an IMU mounted on the stick and one on the bucket

An extended Kalman filter (EKF) and an unscented Kalman filter (UKF) implementation are evaluated that use the predicted measurements of other sensors. Shown in Figure 1.2 is how the received predicted acceleration, and the estimated roll of the excavator arm, are incorporated in the filter implementations. The filter implementations are initialized with an initial state with corresponding state variance matrix, followed by the Kalman prediction step of the internal state vector. The measurement step uses the information of the measurements from the sensors, and the information from the system of IMUs to incorporate the information about the roll of the excavator arm and the predicted acceleration of the IMU. Using the *a posteriori* estimate of the state, the IMU predicts the acceleration of its neighbour IMU with the corresponding certainties and broadcasts all the information on the CAN bus. Only the first IMU estimate the roll angle of the excavator arm, and will therefore not receive any information about the roll of the system. The distributed system of IMUs, estimating the experienced acceleration and roll of other sensors is shown to be both very responsive and adaptive.

1.5 Thesis outline

This thesis begins with a short introduction in Chapter 1, where the problem is formulated and an overview of the concept is shown. Study of previous literature and work is described in Chapter 2. Chapter 3 introduces the necessary theoretical knowledge used

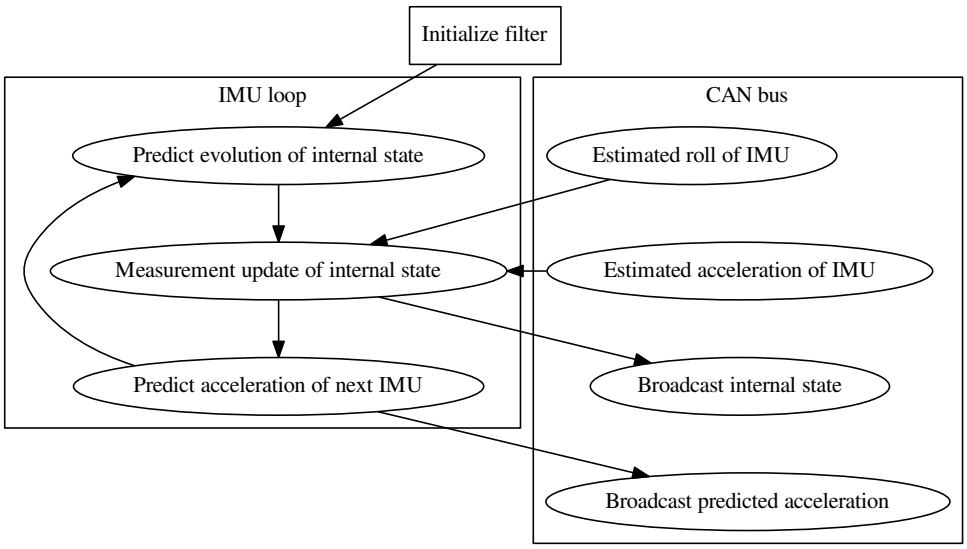


Figure 1.2: An illustration of the different steps in the filter implementations

in this report. The theory is applied on the specific system in Chapter 4 and Chapter 5. In Chapter 4 is the system modelled, and information regarding the specific sensors, microcontrollers and communication is described. The filters are described in Chapter 5, where the filtering technique and the models are linked. Following are the results presented in Chapter 6, that are discussed in Chapter 7. The ending Chapter consists of the concluding ideas in Chapter 8.

2

Existing systems and Related work

Following are the related work and publications that have been performed in the field of study. Different filtering techniques and different representation of angles are analyzed, together with the corresponding motion and sensor models.

There have been tremendous amount of research on IMU and AHRS over the last couple of years. Mainly due to the low price of microelectromechanical systems (MEMS), and the integration of MEMS in mobile devices allowing everyone to have their own AHRS and inclination unit in their cell phone. AHRS have for a long time been used in airplanes and naval applications, where tracking the orientation and position is of high importance at all times. Most recently, there have been a great interest among academics and hobbyist to construct Quadracopters for which knowing the orientation is of high importance.

Madgwick et al. have constructed an orientation filter for IMUs that incorporates tri-axis accelerometer, tri-axis gyroscope and optionally a tri-axis magnetometer. A quaternion representation is used in the filter implementation, effectively avoiding any risk of singularity and allowing the gyroscope to be represented as a quaternion derivative and integrated in the sensor frame. The accelerometer and magnetometer data is used in a gradient-descent algorithm to compute the direction of the gyroscope measurement error as a quaternion derivative. The combination of the gradient descent algorithm and the raw output from the gyroscope is integrated by the Runge Kutta integration method for quaternions, and converted into the Euler angles to represent roll, pitch and yaw in the world frame. The filter has gained much of its popularity by its easy implementation and low cost of computation power making it exceptional for microcontrollers and embedded systems. In the publications [18, 19] the authors present results matching the ones of EKF with root mean square error (RMSE) errors below 0.8° for static measurements and below 1.7° for dynamic RMSE. The filter assumes the acceleration in the world

frame to be the true gravity, which in practice is seldom true. Once the sensor is in motion, there will be other accelerations detected by the accelerometer causing a flawed estimation of the gravity, yielding incorrect data to the filter. The lightweight filter does not estimate other state parameters than the quaternion.

Sabatini estimates the orientation of a rigid body by the use of a quaternion based EKF. A tri-axis accelerometer, gyroscopes and magnetometer are used to provide the necessary data. The sensors are modelled with individual scaling factor, bias and noise. The bias and scale factor are known to be functions of environmental conditions, which is especially true for low cost MEMS gyroscopes. Bias of the gyroscope tend to change with temperature, which is even seen during the warm-up period [27], and may continue to do so dependent on the surrounding temperature. The gyroscope measurements are used as inputs to the motion model, which is integrated to present a quaternion of the orientation. The measurement model is constructed by the accelerometer and the magnetometer, for which the orientation of the gravity and the orientation of north are rotated to the sensor frame by the estimated quaternion. Running the filter at 25 Hz yields RMSE errors in the magnitude of 1.01° and 1.19° in roll and pitch respectively [27]. Although the applications are for studies of the human arm motion, no care is taken to model the movement of the human motion making the filter unable to detect external accelerations.

In robotic orientation estimation, Roan et al. investigate the use of accelerometers and gyroscopes instead of the rotational encoders traditionally used on robot arms. The research compares the complementary filter (CF), the time varying complementary filter (TCF) and an EKF expressed in Euler angles with the encoder value given by the robot. The CF and TCF are implemented by passing the calculated angle from the accelerometer through a low pass filter and the corresponding integrated angle from the gyroscope through a high pass filter. The constant for the cut off frequency can either be fixed for the CF or varying when other motions are detected on the accelerometer as implemented in the TCF. The motion model for the EKF makes use of the geometry of the robot arm. Since the robot is limited to only rotational joints, the angles can easily be solved by rotation matrices in Euler angles to form a model of the accelerometer and the gyroscope.

Results show that the TCF and CF exceeds the result of the EKF in the links closer to the body, whereas the EKF shows better results in the links further away. Although the authors have used the geometry when modelling the motion model, no model for the motion accelerations is proposed. It is however, proposed to make use of the coupled kinematic to find the motion accelerations to better estimate the joint angles.

Any rigid body may make use of the geometry to find a more suitable motion model in order to find a better estimate. Trimpe & D'Andrea positioned several accelerometers and gyroscopes in a cube, that was to balance on one of its corners. The relationship between the placement of the accelerometer was used to find the introduced motion accelerations, and better estimate the acceleration due to gravity. An optimal linear estimator is proposed in the least square sense, if the knowledge of the system dynamics is not used. All the accelerometers broadcast their information, and the filtering is

performed individually in each sensor based on the information from the entire system. Whether the technique is ready to be extended to moving bodies is unknown, although it has shown satisfactory results in the study [31]. The estimation of the gravity vector may be used as a basis for further filtering and fusion technique, for moving dynamic bodies when incorporated with gyroscopes.

The research performed by Vihonen & Honkakorpi shows excellent results incorporating the rate gyroscopes, accelerometers and a dynamic motion model that integrate the introduced motion accelerations. The setup consists of a multi body linkage assembly with only rotational joints, and two IMUs positioned at each end-node of each rigid link. Using Euler angles and rotational matrices, the relationship between each IMU relative the previous IMUs is modelled using the angular velocity, angular acceleration and the linear acceleration in each previous point. For this, the angle, the angular velocity and the angular acceleration is estimated in each joint and fused by the principles of the complementary filter. The author uses a PI-type CF which yields RMSE as low as 0.06° , 0.09° ,0.14° for the first consecutive three joints using link lengths of 0.47 m.

The fact that Euler angles suffers from singularities renders them useless in some applications. Further, using a linear filter such as the CF is not optimal when estimating angles and the nonlinearity the rotation causes. In [15], Kraft solves the estimation problem of the orientation of a rigid body with accelerometer, rate gyroscope and magnetometer measurements with quaternions and an UKF to handle the nonlinear filtering. The UKF is more accurate and less costly than the EKF, as no Jacobian needs to be calculated [15]. Although in this application, there are several extensions to the original UKF to deal with the unit quaternions and the difficulties of finding the mean rotation of several quaternions which the author addresses and proposes solutions for. Results are shown, but no numeric data as to how well the filter estimates the given orientation is supplied.

The study of related work has analysed the sensor models, the motion models and different filtering techniques. The different ways of expressing the angles have a big impact on both the filter and the model, which may results in different accuracy of the estimates. Following are the proposed filters, motion models and sensor models.

3

Background Theory

The following Section describes general theory about the sensors, the communication between the sensors and the filtering techniques used. Each IMU is constructed by sensors connected to a microcontroller, which in term communicate with other microcontrollers and other systems through a CAN bus. A rigid multi body system can be represented in different ways to find the model linking the different bodies to each other. Further, the dynamic system of an excavator is explained and related to the measurements of the sensors. However, a model can never fully describe a system, therefore there are uncertainties relating to both the measurements from the sensors and the dynamics of the excavator. This requires mathematical estimators to find the best estimation of describing the system, which are discussed in the section of estimation theory.

3.1 Sensors

Microelectromechanical systems (MEMS) have enabled the reduction in size of mechanical sensors, and have increased their functionality drastically. The combination of cost, size and performance allow the sensors to be used in a wide range of different products. The automotive field have exerted the strongest push on micromechanical inertial sensor technology, which have been used for passenger safety systems including airbags, vehicle dynamics control, and navigation systems [14]. This has introduced a large demand in volume, allowing the evolution of MEMS to happen. Today's application of inertial sensors covers a much broader spectrum where their small size and low cost have an even larger impact [36].

Inertial sensors measure physical quantities of motion of a solid object. Acceleration of one point of a body and its rate of rotation around three orthogonal axes provide a full description of the body's motion [14]. Other sensors, such as GPS or magnetometer, can be incorporated to give an even better estimate of the orientation of the object [12].

3.1.1 Accelerometer

An accelerometer measures the proper acceleration. It measures the acceleration it experiences relative to free fall. These types of accelerations are popular measures in terms of the acceleration of gravity, where 1 g $\approx$ 9.82 m/s^2. When held fixed in the gravitational field the accelerometer will indicate 1 g upwards relative to a free falling reference frame. Any acceleration introduced by motion will be indistinguishable from acceleration caused by gravity. For the practical purpose of finding the acceleration of objects with respect to the Earth, such as for use in an inertial navigation system (INS), a knowledge of local gravity is required.

The proper acceleration measured is not necessarily the equivalence to the rate of change of velocity in the reference frame. Instead, the accelerometer measures the accelerations experienced by a proof mass attached in the sensor frame with respect to the reference frame. An accelerometer at rest will measure the acceleration of gravity, and an accelerometer in a turning car will measure the acceleration of the vehicle, the centrifugal acceleration and the gravity.

There are a range of different types of accelerations to meet the multiple applications they are used in. The highly sensitive accelerometers are components of INS for aircraft and military systems. There are accelerometers used to measure and detect vibrations and shocks, and accelerometers used in tablet computers and digital cameras to detect the orientation of the product and rotate the screen thereafter. MEMS accelerometers are available in a wide variety of measuring ranges, reaching up to thousands of g's. The designer must make a compromise between sensitivity and the maximum acceleration that can be measured.

A MEMS accelerometer consist of a damped proof mass anchored to a fixed frame. The anchoring can be modelled as cantilever beam with surrounding residual gas sealing the device, having an effective spring constant K and a damping factor D. Any external acceleration on the system will displace the proof mass, and the displace is measured to give an indication of the external acceleration. A figure of a MEMS accelerometer is shown in Figure 3.1a.

The mechanical equation for the system can be written shown in Equation (3.1), where a is the acceleration in the world frame and $\ddot{x}$ is the acceleration relative the fixed frame in the sensor [17]. $D\dot{x}$ and Kx are the forces introduced by the damping and the spring. Taking the Laplace transformation of the mechanical systems yields the result in Equation (3.2).

$$F = ma = m\ddot{x} + D\dot{x} + Kx \tag{3.1}$$

$$H(s) = \frac{x(s)}{a(s)} = \frac{1}{s^2 + \frac{D}{m}s + \frac{K}{m}} = \frac{1}{s^2 + \frac{\omega_r}{Q}s + \omega_r^2}, \tag{3.2}$$

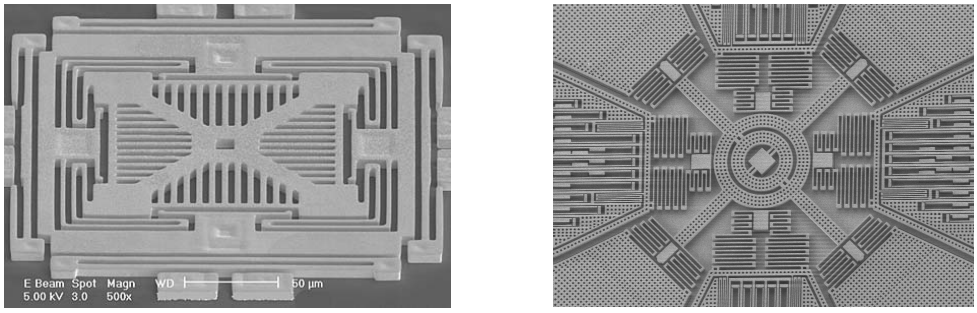
where ω_r is the resonance frequency and Q the quality factor. For most applications, the applied acceleration frequency is much smaller than ω_r, making the mechanical sensitivity of the device $1/\omega_r^2$. The quality factor is $Q = \omega_r m/D$, for which [13] shows the importance of a larger mass.

There are single and multi-axis models of accelerometers, to detect the orientation and motion in different directions. Most MEMS accelerometers operate in only one direction and, by integrating several accelerometers perpendicularly a sensor measuring in two or three axis may be achieved. Manufacturing error may cause misalignment errors causing the respective axis not to be perpendicular. This is commonly referred to as cross axis sensitivity in the sensors.

3.1.2 Gyroscope

The accelerometer is incapable of finding rotational rate of itself, therefore the gyroscope is used. Together with the accelerometer, the rate gyroscope is widely used in INS, robotics and entertainment electronics. Traditional rotating wheels, and precision fiber optic and ring laser gyroscopes are all too expensive and too large to be used in most modern applications. MEMS can reduce the fabrication costs and shrink the sensor size, and allow the electronics to be integrated on the same silicon chip [36].

Modern MEMS rate gyroscopes use at least one vibrating mechanical element that responds to the Coriolis force if its motion is disturbed by a forced rotation around a sensitive axis. This way, there is no need of rotating parts that would require bearings. The size of the Coriolis force generated represents a direct measure for the applied rate of rotation [14, 17, 36, 35]. A figure illustrating a MEMS gyroscope is shown in Figure 3.1b.



(a) Accelerometer [9] **(b)** Gyroscope [37]

Figure 3.1: Mems sensors

Construction of MEMS rate gyroscope may cause geometrical imperfections in the vibrating mechanical structure, and the sense electrodes causing an output signal even in the absence of rotation [38]. The architecture of the rate gyroscopes are fabricated with silicon, which is a temperature sensitive material and its physical characteristics vary greatly with ambient temperature [38]. With an increased ambient temperature, the Young's module of the silicon material changes [35], causing inaccuracies in the output. There are also thermoelectric effects where the difference in temperature introduces voltage differences, causing the sensor to give a bias output. The effect of temperature and humidity may result in a permanent offset [25]. An important performance parameter for vibratory gyroscope is its zero rate output [36].

Inexpensive vibrating rate gyroscopes can be incorporated with accelerometer to give an output that has six degrees of freedom, and can fully describe the motion of the sensor. These can be produced with the MEMS technology to keep the costs low and the size small, allowing the sensors to be mobile. Together with a microcontroller, they can act as standalone embedded systems.

3.2 Microcontroller

A microcontroller is a single integrated circuit, similar to a computer, with a processor core, memory and programmable input and output connectors. These units, as compared with computers, are design to run in real time, in embedded applications.

The use of microcontroller simplifies communication with different sensors, as a microcontroller can communicate over inter-integrated communication (I2C), Serial Peripheral Interface bus (SPI bus). Between each other, the microcontroller may communicate over the CAN bus. Microcontroller are widely used as smaller units in larger systems, where the computational power differs greatly between the different applications.

3.3 CAN communication

Controller area network (CAN) is a communication standard among vehicles designed to allow microcontrollers, and standalone devices to communicate over a network, without the need of a host computer and intermediate connections. It was designed specifically for automotive applications, but is now also used in other areas such as aerospace, maritime and construction equipment. The protocol was officially released in 1986 and have been revised several times since. Today, there are different standards used to meet the specific criteria and demands. The CAN standard mainly used in vehicle components is CAN 2.0B[1] which allows an extended frame with a 29 bit identification address as compared to CAN 2.0A which allows an 11 bit address. The baud rate can be set to either 250 kbit/s or 500 kbit/s.

For the communication to function proper, each node requires a central processing unit (CPU) that can decipher the messages and transmit itself. Several sensors can be connected to the same CPU and share the resource to broadcast their information on the network. Further, a CAN controller with a synchronous clock is necessary that acts as a link between the CPU and the bus. The controller can store the serially received bits, and forward these as an entire message once it is received. To protect the controller, there is a transceiver that can convert the signal received on the bus to levels the controller expects. There is usually some protective circuitry on the transceiver, to protect the CAN controller and inner circuitry.

The messages are transmitted serially onto the bus and is sensed by all nodes. Every node can send messages on the bus, but not simultaneously. The addresses are constructed by identification bits, which can have different levels of priority. A logical 1

[1] SAE J1939

is recessive to the logical 0, making lower addresses (addresses starting with 0) more dominant.

CAN devices send data across the network in packets called frames that consists of an address, data and control bits. The frames begin with a dominant bit, the start of frame bit, followed by the identification bits. The full layout of a frame can be seen in Table B.1.

3.4 Coordinate frames

Coordinate frames are used to express the position and orientation of a point relative a reference. In this report the earth-centered earth-fixed (ECEF), east north up (ENU) and body frames are studied. The ECEF frame rotates along the Earth and has its origin at the centre of the Earth (hence Earth-fixed). The z-axis passes through the north pole, but does not exactly coincide with the rotational axis of the Earth, and the x-axis passes through the intersection of the equatorial plane and the reference meridian [22]. The y-axis completes the right hand coordinate system in the equatorial plane. Shown in Figure 3.2 are the coordinate system.

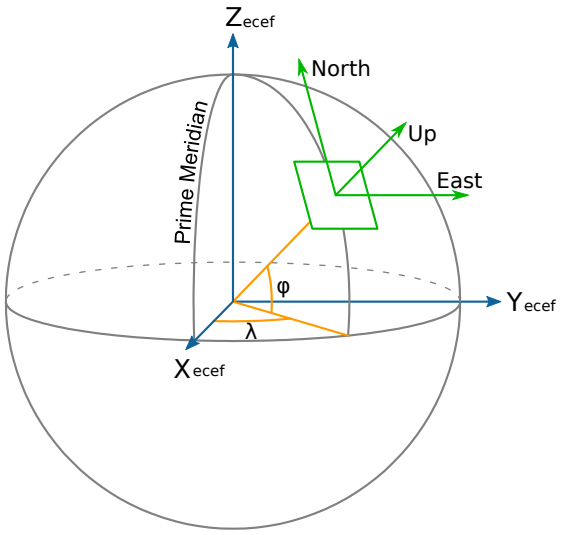
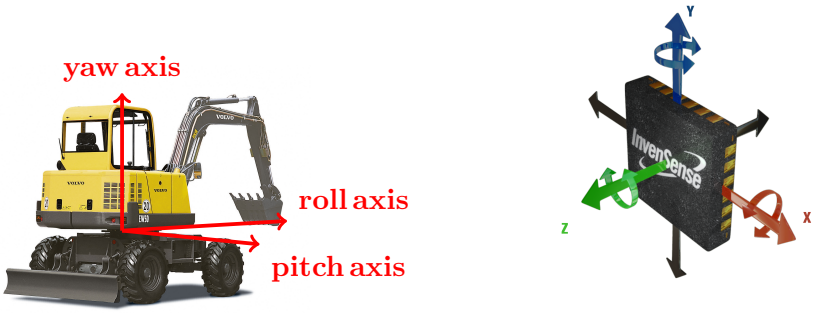


Figure 3.2: The coordinate systems [33]

The ENU frame, is a local tangent plane (LTP), and used to represent attitudes and velocities of objects near the surface of the Earth. This coordinate frame has its x-axis pointing east, y-axis pointing north and the z-axis completes the right hand system by pointing up [22]. Another common system is the north east down (NED) system, where the x, y, and z axis lies along north, east and down to comply with the right-hand system.

The body frame is a coordinate system usually located in the center of gravity of an object. The rotation around the axis pointing in the forward direction is called the

roll-axis, and the rotation around the transverse direction is known as the pitch-axis. Rotations around the last axis that completes the right-hand system and points upwards, are referred to as the yaw or heading of the system. The system is seen in Figure 3.3a. The sensor centered coordinate system is used. It is a coordinate system aligned with the measurement axis of the sensor, seen in Figure 3.3b.



(a) Excavator frame **(b)** Sensor frame

Figure 3.3: The body and sensor frames

3.5 Expressing rotations

Rotations between different frames and coordinate systems can be represented in different ways. Following are the rotation matrices that are mainly used when describing Euler and Tait-Bryan angles, following with a section about quaternion representation of rotations.

3.5.1 Rotations

A rotation matrix is used to change a vectors' orientation, not its magnitude. In 3-space, the rotation around the x, y and z axis can be achieved using the rotation matrices seen in Equation (3.3)[34].

$$R_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \qquad R_y(\phi) = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad R = R_x R_y R_z \tag{3.3}$$

It is important to notice that the order of rotation makes a difference. Hence, letting the rotation of $\phi = 90°$, results in a R_y that aligns the rotational axis of R_x and R_z. As a result, one degree of freedom is lost.

Euler angles are constructed by multiplying three consecutive rotation matrices, but letting the first and last matrix be a rotation around the same axis. Tait-Bryan angles describing commonly describing roll, pitch and yaw uses three different axis to describe the orientation of one frame relative another.

3.5.2 Quaternions

Quaternions can be used to describe rotations. The main advantage over rotation matrices is that the quaternions do not suffer from Gimbal lock. The convention is therefore popular in describing the orientation of satellites and airplanes. Quaternions are defined with a scalar (q_w), with three imaginary vector components ($q_x i$, $q_y j$, $q_z k$) [6]. The imaginary part has the properties seen in Equation (3.4). Following the multiplication of imaginary numbers, the quaternion multiplication, $\otimes$ can be expressed as in Equation (3.5).

$$\begin{aligned} i^2 = j^2 = k^2 = -1 \qquad & ij = -ji = k \\ jk = -kj = i \qquad & ki = -ik = j \end{aligned} \tag{3.4}$$

$$\mathbf{q^1} \otimes \mathbf{q^2} = \begin{pmatrix} q_w^1 & -q_x^1 & -q_y^1 & -q_z^1 \\ q_x^1 & q_w^1 & -q_z^1 & q_y^1 \\ q_y^1 & q_z^1 & q_w^1 & -q_x^1 \\ q_z^1 & -q_y^1 & q_x^1 & q_z^1 \end{pmatrix} \begin{pmatrix} q_w^2 \\ q_x^2 \\ q_y^2 \\ q_z^2 \end{pmatrix} \tag{3.5}$$

A quaternion rotation around an axis $\mathbf{e} = [e_x, e_y, e_z]^T$ with the angle ϕ is represented in the following quaternion seen in Equation (3.6).

$$q_0 = \cos(\phi/2) \qquad \mathbf{q_{xyz}} = \frac{\mathbf{e}}{|\mathbf{e}|}\sin(\phi/2) \tag{3.6}$$

Rotating a vector $\mathbf{p} = [0, p_x, p_y, p_z]^T$ is shown in Equation (3.7). The quaternion inverse, $\mathbf{q^{-1}}$, is shown in Equation (3.8).

$$\mathbf{p'} = \mathbf{q^{-1}} \otimes \mathbf{p} \otimes \mathbf{q} \tag{3.7}$$

$$\mathbf{q^{-1}} = \begin{pmatrix} q_w \\ -\mathbf{q_{xyz}} \end{pmatrix} \tag{3.8}$$

The derivative of a quaternion is shown in Equation (3.9) as expressed in [1, 16]. The rate of rotation is expressed in a quaternion using the three axes such that $\boldsymbol{\omega} = [0, \omega_x, \omega_y, \omega_z]^T$.

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega} \tag{3.9}$$

3.6 Kinematics model

In the following section, the kinematic model describing how the state of the system evolve in time is described. The Denavit Hartenberg (DH) convention is presented which is commonly used in robotics, followed by a relative method of finding accelerations in a rigid multi body system.

3.6.1 Forward position kinematics

The Denavit Hartenberg (DH) parameters are commonly used when representing kinematic models of robots. A robot with n joints, will have $n+1$ links [11], where the parameters are used to represent the position, and orientation relative two frames [4]. The Denavit Hartenberg (DH) parameters are regarded as the products of two rotations and two translations, as seen in Equation (3.10). Forward kinematic can be described from the base frame to each frame in the system, by going through each link up to the specific frame.

Coordinate frames are attached to the joints between two links, such that the first transformation is associated with the joint and the other associated with the link [30]. Joints can be modelled as either revolute or prismatic joints, to represent the relative movement between the links in the system.

The DH representation is seen in Equation (3.10). The constants α and a are specific for each joint in the system and are called link parameters [11]. a is the distance between the axes z_{i-1} and z_i, and α is the twist angle that screws the z_{i-1} axis into the z_i axis. Since each joint is defined with only one degree of freedom, i.e. can be only be either prismatic or revolute, only one of the variables θ and d is variable and the other is fixed. For a revolute joint θ, and for a prismatic joint d is used, where d is the distance between the x_{i-1} axis and the x_i axis. θ is the angle that screws the x_{i-1} axis into the x_i axis.

$$
\begin{aligned}
{}^{i-1}A_i &= Trans_{z_{i-1},d_i} Rot_{z_{i-1},\theta_i} Trans_{x_{i-1},a_i} Rot_{x_{i-1},\alpha_i} \\
&= \left(\begin{array}{ccc|c}
\cos(\theta_\mathrm{i}) & -\cos(\alpha_\mathrm{i})\,\sin(\theta_\mathrm{i}) & \sin(\alpha_\mathrm{i})\,\sin(\theta_\mathrm{i}) & a_i\,\cos(\theta_\mathrm{i}) \\
\sin(\theta_\mathrm{i}) & \cos(\alpha_\mathrm{i})\,\cos(\theta_\mathrm{i}) & -\sin(\alpha_\mathrm{i})\,\cos(\theta_\mathrm{i}) & a_i\,\sin(\theta_\mathrm{i}) \\
0 & \sin(\alpha_\mathrm{i}) & \cos(\alpha_\mathrm{i}) & d_i \\
\hline
0 & 0 & 0 & 1
\end{array}\right)
\end{aligned}
\tag{3.10}
$$

The transformation between two adjacent frames can be expanded to go, link by link, to any frame in the system. To find the orientation and position of the end frame relative the base frame, multiplying consecutive DH parameters from the base to the end node gives the satisfying result. This is seen in Equation (3.11) using only revolute joints [24].

$$
{}^0A_n = {}^0\,A_1(\theta_1)\,{}^1A_2(\theta_2)\,...\,{}^{n-1}A_n(\theta_n) \tag{3.11}
$$

3.6.2 Relative motion

The relative motion between two moving frames can be calculated by deriving the DH parameters with respect to time. Using this method, the forward angular and linear motion of any link in the system can be found[11]. The relative motion can also be described in the local coordinate system [3]. Considering a point p, moving in coordinate system xyz, can be expressed in the global coordinate system XYZ and the local coordinate system as in Equation (3.12). The velocity expressed in the global coordinate system then becomes as in Equation (3.13) when deriving Equation (3.12) with respect to time.

$$r_{p/o} = r_{o'/o} + r_{p/o'} \tag{3.12}$$

$$v_p = v_{o'} + (vp)_{xyz} + \omega \times r_{p/o'} \tag{3.13}$$

Angular acceleration is expressed in α, and θ is the angular velocity in the global coordinate system. The acceleration of the point p in the global coordinate system can be achieved by deriving Equation (3.13) with respect to time as in Equations (3.14) to (3.17).

$$a_p = \frac{d}{dt}v_p = \frac{d}{dt}v_{o'} + \frac{d}{dt}(v_p)_{xyz} + \frac{d}{dt}(\omega \times r_{p/o'}) \tag{3.14}$$

$$\frac{d}{dt}(v_p)_{xyz} = (a_p)_{xyz} + \omega \times (v_p)_{xyz} \tag{3.15}$$

$$(a_p)_{xyz} = \frac{\partial}{\partial t}(v_p)_{xyz} \tag{3.16}$$

$$\frac{d}{dt}(\omega \times r_{p/o'}) = \alpha \times r_{p/o'} + \omega \times [(v_p)_{xyz} + \omega \times r_{p/o'}] \tag{3.17}$$

The acceleration of the point p can now be expressed as in Equation (3.18). For a fixed rigid body system, with only revolute joints, the acceleration of a point p can be expressed as in Equation (3.19) since $(a_p)_{xyz} = (v_p)_{xyz} = 0$, [3]. The acceleration of the point p consists of the acceleration of the moving frame $a_{o'}$ expressed in the global coordinate system, the tangential acceleration of the point $\alpha \times r_{p/o'}$ and the centrifugal acceleration $\omega \times (\omega \times r_{p/o'})$.

$$a_p = a_{o'} + (a_p)_{xyz} + \alpha \times r_{p/o'} + \omega \times (\omega \times r_{p/o'}) + 2\omega \times (v_p)_{xyz} \tag{3.18}$$

$$a_p = a_{o'} + \alpha \times r_{p/o'} + \omega \times (\omega \times r_{p/o'}) \tag{3.19}$$

3.7 Estimation theory

There are different techniques to estimate the state of a time varying system from which noisy measurements are observed. The state of the system may refer to angles, accelerations and positions that can be observed by measurements. The noise in the measurements means that there is a certain degree of uncertainty. A function of the dynamic

system may be formed, connecting how the state evolve as a function of time. Knowing that the dynamic system can not entirely be modelled deterministically the process uncertainty is required[29]. The model of the state transition is therefore modelled with some degree of uncertainty.

3.7.1 Bayesian statistics

Bayesian estimator aim to construct the posterior probability density function (PDF) of the required state vector, given all the available information. This gives a complete description state describing the system [28], and it allows tracking of object over time. The recursive Bayesian filters provide a formal mechanism for propagating and updating the posterior probability density function (PDF) as new measurements are received.

Dynamic estimation problem assumes two fundamental mathematical models; (i) the state dynamics and (ii) the measurement equation. The dynamic model, seen in Equation (3.20), describes how the state vector evolves with time. The x_{k-1} vector is the previous state, u_{k-1} some input to the function f_{k-1} and v_{k-1} is the process noise.

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, v_{k-1}) \tag{3.20}$$

The formal Bayesian recursive filter consists of a prediction and an update operation. The PDF of v_{k-1} is assumed to be known. Equation (3.20) defines a dynamic model, where a known previous state $p(x_{k-1}|Z_{k-1})$ is propagated from the time $k-1$ to the time k through the dynamics function. Z_{k-1} denotes all the measurements received up to z_{k-1} [28]. An equivalent probabilistic description of the state evolution is $p(x_k|x_{k-1})$ which may be called the transition density, or the prior PDF. When f is linear, and v is Gaussian so is the transition density [28].

Using a measurement equation seen in Equation (3.21), an equivalent conditional PDF $p(z_k|x_k)$ may be achieved.

$$z_k = h_k(x_k, w_k) \tag{3.21}$$

The posterior PDF, $p(x_k|Z_k)$, may be achieved as Equation (3.22) [28].

$$\underbrace{p(x_k|Z_k)}_{\text{Posterior}} = \underbrace{p(z_k|x_x)}_{\text{Likelihood}} \underbrace{p(x_k|Z_{k-1})}_{\text{Prior}} / \overbrace{p(z_k|Z_{k-1})}^{\text{Normalising denominator}} \tag{3.22}$$

$$\text{where } \underbrace{p(x_k|Z_{k-1})}_{\text{Prior at k}} = \int \underbrace{p(x_k|x_{k-1})}_{\text{Dynamics}} \underbrace{p(x_{k-1}|Z_{k-1})}_{\text{Posterior from k-1}} dx_{k-1} \tag{3.23}$$

3.7.2 Kalman filter

The Kalman filter (KF) is one of the most important data fusion algorithm used today. It was invented over 50 year ago by Rudolf E. Kálmán, and became popular due to its

small computational requirement and status as the optimal estimator for one dimensional linear systems with Gaussian noise [7].

Typical use of the Kalman filter (KF) include smoothing noisy data and providing estimates and observations of the state of a system. Applications include everything from financial data, to global positioning systems and smoothing the output from laptop trackpads.

The KF is an efficient recursive filter that estimates the internal state of a linear dynamic system, given a series of measurements. Measurements are usually noisy observations that are related to the estimated state through some dynamics. The estimated state vector can be much larger than the given measurements.

The recursive algorithm work in a two step process; (i) a state transition prediction and (ii) a measurement update. The state transition step predicts the estimates of the next state, for which both the estimated state its estimated covariance are predicted. The estimated state vector and its covariance are then corrected using the Kalman gain, which is constructed using measurements from the system. This two step recursive algorithm can run in real time using only the present input measurements, the previous calculated state and its uncertainties.

The Kalman filter assumes that the state from the previous time $k-1$ evolve

$$x_k = F_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}, \tag{3.24}$$

where x_{k-1} is the state vector at time $k-1$, u_{k-1} is the input vector. F_{k-1} is the state transition matrix, and B_{k-1} is the control input matrix at time $k-1$. w_{k-1} is the vector containing the process noise term for each parameter in the state vector, assumed to be zero mean with covariance described by Q_{k-1}. Measurements of the system can be modelled

$$z_k = H_{k-1}x_{k-1} + v_{k-1}, \tag{3.25}$$

where H_{k-1} is the state matrix that maps the state vectors into the measurements domain and v_{k-1} is the measurement noise term for each measurement. The v_{k-1} is assumed to be Gaussian white noise with zero mean and variance R_{k-1}.

$$p(\mathbf{w_{k-1}}) \sim \mathcal{N}(0, Q_{k-1}) \qquad p(\mathbf{v_{k-1}}) \sim \mathcal{N}(0, R_{k-1}) \tag{3.26}$$

The KF work in a two step process with a state transition model and an measurement update sequence. The state vector is assumed to follow the state transition matrix F_{k-1} shown in Equation (3.27). From the state transition function the predicted *a priori* state $\hat{x}_{k|k-1}$ can be estimated, shown in Equation (3.28)

$$x_k = F_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} \tag{3.27}$$

$$\begin{aligned}\hat{x}_{k|k-1} =& E[F_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}] = F_{k-1}E[x_{k-1}] + B_{k-1}u_{k-1} \\ =& F_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}\end{aligned} \tag{3.28}$$

The corresponding covariance matrix, the *a priori* covariance matrix $P_{k|k-1}$ is estimated using the following derivation

$$\begin{aligned} e_{k|k-1} =& x_{k-1} - \hat{x}_{k|k-1} = F_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1}) + w_{k-1} && (3.29)\\ P_{k|k-1} =& E[e_{k|k-1}\, e_{k|k-1}^T] \\ =& E[(F_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1}) + w_{k-1})(F_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1}) + w_{k-1})^T] \\ =& F_{k-1}\, E[(x_{k-1} - \hat{x}_{k-1|k-1})(x_{k-1} - \hat{x}_{k-1|k-1})^T]\, F_{k-1}^T \\ &+ F_{k-1}\, E[(x_{k-1} - \hat{x}_{k-1|k-1})w_{k-1}^T] + E[w_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1})^T]F_{k-1}^T \\ &+ E[w_{k-1}w_{k-1}^T] && (3.30) \end{aligned}$$

The state error and process error are assumed to be uncorrelated, which gives the results shown in Equation (3.31). This yields the result shown in Equation (3.32), where $P_{k-1|k-1}$ is the *a posteriori* covariance matrix at time $k-1$.

$$F_{k-1}E[(x_{k-1} - \hat{x}_{k-1|k-1})w_{k-1}^T] = E[w_{k-1}(x_{k-1} - \hat{x}_{k-1|k-1})^T]F_{k-1}^T = 0 \quad (3.31)$$

$$\begin{aligned} P_{k|k-1} &= F_{k-1}\, E[(x_{k-1} - \hat{x}_{k-1|k-1})(x_{k-1} - \hat{x}_{k-1|k-1})^T]\, F_{k-1}^T + E[w_{k-1}w_{k-1}^T] \\ &= F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1} \end{aligned} \quad (3.32)$$

Filter cycle

The filter is initialised with some state estimate $\hat{x}_{0|0}$ with a covariance matrix $P_{0|0}$. The Kalman filter estimates the *a priori* state vector $\hat{x}_{k|k-1}$ and its covariance matrix $P_{k|k-1}$ using the state transition matrix F_{k-1}.

$$\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1} \quad (3.33)$$

$$P_{k|k-1} = F_{k-1}P_{k|k}F_{k-1}^T + Q_{k-1} \quad (3.34)$$

The update step handles the new information available at time k, where the *a priori* estimate must be corrected with the new information which is handled by the Kalman gain. The measurement model, together with the innovation matrix S_k and the *a priori* state error, the covariance matrix $P_{k|k-1}$ is used to find the Kalman gain K_k, seen in Equation (3.36). The Kalman gain is constructed to minimize the variance of the state vector [8].

$$S_k = H_{k-1}P_{k|k-1}H_{k-1}^T + R_{k-1} \quad (3.35)$$

$$K_k = P_{k|k-1}H_{k-1}^T(S_k)^{-1} \quad (3.36)$$

The Kalman gain is used to update the *a priori* prediction and its covariance matrix. The gain is constructed to balance the noise levels of the sensor data, and the uncertainty of the *a priori* estimate, by weighting their respectively uncertainties.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_{k-1}\hat{x}_{k|k-1}) \quad (3.37)$$

$$P_{k|k} = (I - K_kH_{k-1})P_{k|k-1} \quad (3.38)$$

The *a posteriori* state vector estimate and the covariance matrix $P_{k|k}$ are the output of the filter which are used in the next cycle. There are several applications and extensions of the KF, such as the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) which are both better suited for nonlinear systems.

3.7.3 Extended Kalman filter

The extended Kalman filter (EKF) is a nonlinear version of the Kalman filter. Using the first order Taylor expansion of the nonlinear system equations around the previous state, allows the retrieval of the distribution a random variable with Gaussian distribution through nonlinear functions. The approximation is acceptable if the signal to noise ratio is low, or the nonlinearity is almost linear [2].

Similar to the KF, there is a process function f and a measurement function h that propagates the state $\hat{x}_{k-1|k-1}$ one time step through the function f. Likewise, the predicted measurement $\hat{z}_k$ is found by propagating the predicted state $\hat{x}_{k|k-1}$ through the function h.

Propagation of uncertainty by linearisation

Since the filter uses a state vector of estimates with corresponding covariances, the covariance matrices needs to be update to the new time step. In order to propagate an estimation $\hat{\mathbf{x}} \in \mathbb{R}^n$, with mean $\mu_{\mathbf{x}}$ and covariance matrix $\mathbf{\Sigma_x}$, the function may be linearised [23].

$$\mathbf{y} = f(\mathbf{x}) \approx f(\mu_{\mathbf{x}}) + \mathcal{F}(\mathbf{x} - \mu_{\mathbf{x}}) \tag{3.39}$$

Where $\mathcal{F}$ is the Jacobian of the function f expressed as $\mathcal{F} = \frac{\partial f}{\partial \mathbf{x}}\big|_{\mu_x}$. The estimated vector $\hat{\mathbf{y}}$ will now have mean and a covariance matrix

$$\mu_y = f(\mu_{\mathbf{x}}) \qquad\qquad \mathbf{\Sigma_y} = \mathcal{F}\mathbf{\Sigma_x}\mathcal{F}^T \tag{3.40}$$

Filter cycle

The recursive prediction-measurement cycle is shown in Equation (3.43) to Equation (3.51). Similar to the KF, the EKF is split into one measurement update providing $\hat{x}_{k-1|k-1}$, $P_{k-1|k-1}$ and a time update yielding $\hat{x}_{k|k-1}$, $P_{k|k-1}$.

A system described by the state transition function f with process noise w_{k-1}, and a measurement model h with measurement noise vector v_{k-1}. w_{k-1} and v_{k-1} are assumed to be uncorrelated white Gaussian noise with zero mean and covariances given by Q_{k-1} and R_{k-1} respectively.

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \tag{3.41}$$
$$y_k = h(x_k) + v_k \tag{3.42}$$

The filter is initialized with an estimate $\hat{x}_{0|0}$ with corresponding covariance matrix $P_{0|0}$. The predicted state vector, $\hat{x}_{k|k-1}$, with its corresponding covariance matrix,

$P_{k|k-1}$, is found by linearising the state transition function around the estimated point, shown in Equations (3.43) to (3.45).

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \tag{3.43}$$
$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \tag{3.44}$$
$$F_{k-1} = \left.\frac{\partial f(x, u_{k-1})}{\partial x}\right|_{x=\hat{x}_{k-1|k-1}, u_{k-1}} \tag{3.45}$$

The predicted measurement is estimated using the predicted state vector, seen in Equation (3.46). Linearising the measurement model around the predicted state, allows the calculation of the Kalman gain seen in Equation (3.48). The predicted state vector is updated using the Kalman gain, Equation (3.49) and its covariance matrix in Equation (3.50).

$$\hat{y}_k = h(\hat{x}_{k|k-1}, u_k) \tag{3.46}$$
$$S_k = H_{k-1} P_{k|k-1} H_{k-1}^T + R_{k-1} \tag{3.47}$$
$$K_k = P_{k|k-1} H_k^T (S_k)^{-1} \tag{3.48}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_k) \tag{3.49}$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{3.50}$$
$$H_{k-1} = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}_{k|k-1}} \tag{3.51}$$

3.7.4 Unscented Kalman filter

The unscented Kalman filter (UKF) is based on the regular Kalman filter, but able to handle nonlinear state transition and measurement models. To propagate the covariance matrices, the filter implementation make use of unscented transform. This, as compared with the EKF, does not require derivation of the models and thus the Jacobian does not need to be calculated. Instead, it makes use of the method unscented transform, which deterministically selects different sigma points each with a corresponding weight that are propagated through the functions.

A system may have the following form, where x_k is the state vector at time k, f is the state transition function and h is the measurement model. The uncertainties w_k and v_k are assumed to be zero mean and normally distributed.

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \tag{3.52}$$
$$y_k = h(x_k) + v_k \tag{3.53}$$

Unscented transform

A minimal set of sample points are deterministically chosen and propagated through the nonlinear function to capture the posterior mean and covariance to the 2nd order of Taylor approximation [10].

$$\chi_{0,k-1} = \hat{x}_{k-1|k-1} \qquad i = 0 \tag{3.54}$$

$$\chi_{i,k-1} = \hat{x}_{k-1|k-1} + \sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}_i \qquad \forall i = 1,..,n \tag{3.55}$$

$$\chi_{i,k-1} = \hat{x}_{k-1|k-1} - \sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}_{i-n} \qquad \forall i = n+1,..,2n \tag{3.56}$$

$$\lambda = \alpha^2(n+\kappa) - n \tag{3.57}$$

The scaling factor α determines the spread of the sigma points around $\hat{x}_{k-1}$ and is usually set to a small positive value, and κ is a secondary scaling factor. $\sqrt{(n+\lambda)\mathbf{P}_{k-1|k-1}}_i$ is the ith row or column of the matrix square root of $(n+\lambda)\mathbf{P}_{k-1|k-1}$. The square root can calculated using lower triangular Cholesky factorization, and the state vector $x_{k-1} \in R^{n\times 1}$ and $y_{k-1} \in R^{m\times 1}$. The corresponding weight to the sigma points are calculated in the following manner [5].

$$W_0^m = \frac{\lambda}{n+\lambda} \tag{3.58}$$

$$W_0^c = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{3.59}$$

$$W_i^m = W_0^c = \frac{\lambda}{2(n+\lambda)} \qquad \forall i = 1,...,2n \tag{3.60}$$

The β parameter adjust the weighting of the zeroth sigma point of the calculation of the covariance, where $\beta = 2$ is optimal for Gaussian distribution [5]. The sigma points are propagated through the nonlinear function as

$$\chi_{i,k} = \mathbf{f}(\chi_{i,k-1}) \qquad \forall i = 0,...,2n \tag{3.61}$$

The predicted state vector $\hat{x}_{k|k-1}$ of the transformed variable is given by the weighted sum of the transformed sigma points shown in Equation (3.62). The predicted covariance is calculated using the corresponding weight to each sigma point, shown in Equation (3.63).

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} W_i^m \chi_{i,k} \tag{3.62}$$

$$P_{k|k-1} = \sum_{i=0}^{2n} W_i^c [\chi_{i,k} - \hat{x}_{k|k-1}][\chi_{i,k} - \hat{x}_{k|k-1}]^T \tag{3.63}$$

Filter cycle

The UKF makes use of the unscented transform when estimating the mean and the covariance, and as such no derivation of the nonlinear functions are needed to propagate the respective variance. By deterministically selecting sigma points $\chi_{i,k-1}$ using the

previous state covariance $P^{xx}_{k-1|k-1}$, the prediction of the state vector $\hat{x}_{k-1|k}$ and its covariance matrix $P^{xx}_{k-1|k}$, is made using the chosen sigma points with its corresponding weight for mean and covariance. The sigma points are further used to find the estimated measurement $\hat{y}_k$ with its corresponding covariance matrix P^{yy}_k.

$$\chi_{i,k} = \mathbf{f}(\chi_{i,k-1}, u_{k-1}) \qquad \hat{x}_{k|k-1} = \sum_{i=0}^{2n} W_i^m \chi_{i,k} \tag{3.64}$$

$$\mathcal{Y}_{i,k} = \mathbf{h}(\chi_{i,k}) \qquad \hat{y}_k = \sum_{i=0}^{2n} W_i^m \mathcal{Y}_k \tag{3.65}$$

The corresponding covariance matrices $P^{xx}_{k|k-1}$, P^{yy}_k, and the innovation matrix are given by the equations shown in Equations (3.66) and (3.67). The covariance matrix of the measurement error is calculated in a similar manner, using the covariance weight and the sigma points of $\mathcal{Y}$.

$$P^{xx}_{k|k-1} = \sum_{i=0}^{2n} W_i^c [\chi_{i,k} - \hat{x}_{k|k-1}][\chi_{i,k} - \hat{x}_{k|k-1}]^T \tag{3.66}$$

$$P^{vv}_k = P^{yy}_k + R_k \tag{3.67}$$

The cross correlation matrix between the observations and the state prediction is given by Equation (3.68). Knowing the cross correlation matrix, allows the calculation of the Kalman gain $\mathcal{K}_k$ shown in Equation (3.69).

$$P^{xy}_k = \sum_{i=0}^{2n} W_i^c [\chi_{i,k} - \hat{x}_{k|k-1}][\mathcal{Y}_{i,k} - \hat{y}_k]^T \tag{3.68}$$

$$\mathcal{K}_k = P^{xy}_k (P^{vv}_k)^{-1} \tag{3.69}$$

The estimated state vector $\hat{x}_{k|k}$ and the updated covariance matrix $P^{xx}_{k|k}$ are given by Equations (3.70) and (3.71).

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \mathcal{K}_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right) \tag{3.70}$$

$$P^{xx}_{k|k} = P^{xx}_{k|k-1} - \mathcal{K}_k P^{vv}_k \mathcal{K}_k^T \tag{3.71}$$

Similar to the other recursive two step Kalman filter implementations, the filter continues to estimate the state at time $k+1$, and once the measurements are achieved the predicted state and its covariance is corrected.

4

Modelling of the system

A model is seldom complete, and will therefore consists of flaws. However, some models are more accurate than others, and they are therefore crucial for the entire solution to work well. In the following Chapter, different models are described in how the system evolves in time and how the sensors can be represented.

4.1 State vector

The state of a mechanical system completely describes the system at a particular time. This is to be modelled, using a state vector describing only the variables needed. There is a need to describe the orientation, the angular velocity and the angular acceleration. Further, the bias of the gyroscope needs to be estimated to complete the possibly flawed measurements of the gyroscope.

The orientation is represented by a quaternion $\mathbf{q} = \begin{pmatrix} q^w & q^x & q^y & q^z \end{pmatrix}$. This avoids the problem when two axes align, known as the Gimbal lock. The angular velocity is estimated in the body frame of the sensor, and is described by $\omega = \begin{pmatrix} \omega^x & \omega^y & \omega^z \end{pmatrix}$. The angular acceleration and the bias of the gyroscope are both estimated in the body frame and represented by $\alpha = \begin{pmatrix} \alpha^x & \alpha^y & \alpha^z \end{pmatrix}^T$ and $\mathbf{b} = \begin{pmatrix} b^x & b^y & b^z \end{pmatrix}$. Thus, the state vector $\mathbf{x} \in R^{13x1}$ is shown in Equation (4.1).

$$\boldsymbol{x} = \begin{pmatrix} \mathbf{q} & \omega & \alpha & \mathbf{b} \end{pmatrix}^T \tag{4.1}$$

4.2 Excavator model

The model of the excavator describes how all the links are connected, and how the inclinations, velocities and accelerations in each link affect each part of the system. The

excavator is modelled using the Denavit Hartenberg (DH) parametrization as seen in Table 4.1.

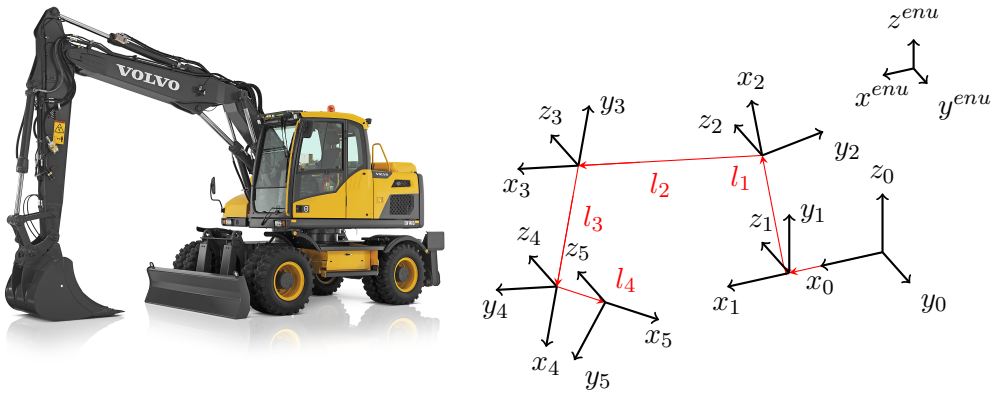
Table 4.1: Parameters for Denavit Hartenberg

#	Joint	a_i	α_i	d_i	θ_i
1	Rotation between platform and house	p_x	$\pi/2$	0	θ_1
2	Rotation between the house and first boom	l_1	0	p_z	θ_2
3	Rotation between the first boom and the second boom	l_2	0	0	θ_3
4	Rotation between the second boom and stick	l_3	0	0	θ_4
5	Rotation between the stick and the bucket	l_4	0	0	θ_5

The table shows the parameters used for the DH parametrization of an excavator. There are solely rotational joints, going from the platform of the excavator to the end node of the bucket.

An excavator can be constructed in several ways. In this report, an excavator is modelled on wheels, with a rotational axis between the platform and the house. The boom is in two parts, followed by a stick and a bucket. As such, there are five revolute joints with the joint angles $\theta_{i=1..5}$. The parameters $l_{i=1..4}$ corresponds to the length of the first and second boom, stick and the bucket respectively. The parameter p_x and p_z corresponds to the translation from the revolute joint between the house and the platform, and the position of the joint between the first boom and the house.

A Volvo excavator can be seen in Figure 4.1a. The specific excavator is on wheels, and have a boom in two parts, a stick and a bucket. A model of an excavator is seen in Figure 4.1b.



(a) Volvo EW140D **(b)** Excavator model

Figure 4.1: A real excavator with the model of an excavator

4.2.1 Prediction of acceleration of the next sensor in line

The acceleration experienced in a rigid multi body system can be calculated. Knowing the acceleration and rotations in one point, the accelerations in a fixed body can be calculated in the local coordinate system by the method shown in Section 3.6.2. This acceleration can be rotated to the ENU coordinate system using the quaternion math shown in Section 3.5.2.

Imagine two sensors mounted on an excavator arm in the points i and $i+1$ as seen in Figure 4.2. Between the two sensors is a revolute joint, in a point $\vec{\boldsymbol{r}}_i$ from the first sensor and $\vec{\boldsymbol{r}}_{i+1}$ from the second, expressed in their respective coordinate systems. The acceleration is the same in point $\boldsymbol{R}$ no matter if its expressed in the coordinate system of i or in the one of $i+1$.

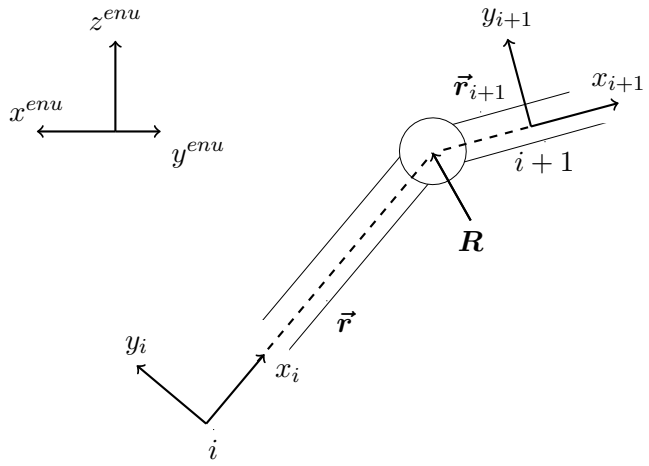


Figure 4.2: The acceleration in a fixed body system can be calculated using the known state

Knowing the linear acceleration $\boldsymbol{a}_i^i$, the angular acceleration $\boldsymbol{\alpha}_i$ and angular velocity $\boldsymbol{\omega}_i$ in point i express in the coordinate system of i, the linear acceleration in point $\boldsymbol{R}$ can be calculated using the equation shown in Equation (4.2). Knowing the quaternion $\boldsymbol{q}_i$ that rotates the coordinate system i into the ENU coordinate system, the acceleration can be expressed in the local tangent plane (LTP).

$$\boldsymbol{a}_R^i = \boldsymbol{a}_i^i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \vec{\boldsymbol{r}}_i) + \boldsymbol{\alpha}_i \times \vec{\boldsymbol{r}}_i \tag{4.2}$$

$$\begin{pmatrix} 0 \\ \boldsymbol{a}_R^{enu} \end{pmatrix} = \boldsymbol{q}_i \otimes \begin{pmatrix} 0 \\ \boldsymbol{a}_R^i \end{pmatrix} \otimes \boldsymbol{q}_i^{-1} \tag{4.3}$$

To find the acceleration in the point of $i+1$ expressed in the coordinate system of $i+1$, the same equations may be used. Knowing the quaternion $\boldsymbol{q}_{i+1}$ that rotates the coordinate system $i+1$ into the ENU coordinate system, the conjugate of the quaternion

can be used to find how the ENU coordinate system rotates into $i+1$.

$$\begin{pmatrix} 0 \\ \boldsymbol{a}_R^{i+1} \end{pmatrix} = \boldsymbol{q}_{i+1}^{-1} \otimes \begin{pmatrix} 0 \\ \boldsymbol{a}_R^{enu} \end{pmatrix} \otimes \boldsymbol{q}_{i+1} \tag{4.4}$$

$$\boldsymbol{a}_{i+1}^{i+1} = \boldsymbol{a}_R^{i+1} + \boldsymbol{\omega}_{i+1} \times (\boldsymbol{\omega}_{i+1} \times \vec{\boldsymbol{r}}_{i+1}) + \boldsymbol{\alpha}_{i+1} \times \vec{\boldsymbol{r}}_{i+1} \tag{4.5}$$

In the case when there are no rotations and the system is standing still, there should only be the gravity present in the point.

4.2.2 Common roll in the excavator arm

The arm of the excavator moves in a plane, hence positioning the sensors along the arm aligns them in the plane of the arm. This yields the same yaw and roll of all sensors. A quaternion can not directly describe the roll, but needs to be converted to the Tait-Bryan angles. The Tait-Bryan rotations can be express as the rotations of quaternion (Section 3.5.2) in the following way

$$\begin{pmatrix} 0 \\ \mathbf{p}' \end{pmatrix} = \mathbf{q}^{-1} \otimes \begin{pmatrix} 0 \\ \mathbf{p} \end{pmatrix} \otimes \mathbf{q} = \begin{pmatrix} \mathbf{0}_{1x3} \\ \mathbf{Q} \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{p}' \end{pmatrix} \tag{4.6}$$

$$\mathbf{p}' = \mathbf{R_z R_y R_x p} \tag{4.7}$$

$$\mathbf{Q} = \mathbf{R_z R_y R_x}, \tag{4.8}$$

where both $\mathbf{Q} \in R^{3x3}$ and $\mathbf{R_z R_y R_x} \in R^{3x3}$, and all the elements in $\mathbf{Q}$ needs to equal the elements in $\mathbf{R_z R_y R_x}$. This shows that $\sin(\phi) = 2(q_y q_z + q_w q_x)$. Assuming that the roll will be within +/- 15 °, the approximation $\sin(x) \approx x$ can be made for small angles.

$$\sin(\phi) = 2(q_y q_z + q_w q_x) \longrightarrow \phi \approx 2(q_y q_z + q_w q_x) \tag{4.9}$$

A similar relationship can be found by investigating the yaw of the excavator arm.

4.3 Physical sensor

The system is equipped with two types of sensors that measures physical (derived) quantities. The sensor used in the application is the MPU-6050 from Invensense. It is a microelectromechanical systems (MEMS) sensor, combining 3 axes accelerometer and 3 axes rate gyroscope in one unit.

4.3.1 Accelerometer

The accelerometer is constructed by separate MEMS sensors, each consisting of a separate proof mass, and their axes are not perfectly perpendicular. For this reason, the

measurements in the body frame may be a combination of the perpendicular axes. Accelerometer is modelled in the body frame as seen in Equation (4.10).

$$\mathbf{a_b^m} = \mathbf{S_a a_b} + \mathbf{b_a} + \mathbf{v_{acc}} \tag{4.10}$$

The $\mathbf{S_a}$ may be used to indicate the scaling of the sensors and the cross-axis correlation. These are ideally identity matrices $\mathbf{I_{3\times 3}}$ but may contain both correlation between the axes and different scaling for the different axes. The $\mathbf{a_b}$ is the true acceleration in the body frame, $\mathbf{b_a}$ is an offset. The $\mathbf{v_{acc}}$ is the sensor noise.

The full scale range of the digital output is adjustable between ±2,4,8,16 g, where the digital output is 16 bits. The scale of the accelerometer is chosen to ±2 g to be within the working range for the application, yet output a good resolution.

Thermal changes affects the accelerometer, and make may alter the scaling and zero offset. Subjecting different accelerometer units to an increase in temperature yields the results seen in Figure A.1.

The distribution of the sensor measurement, for a fixed sensor measuring solely gravity, can be seen in Figure 4.3. The standard deviations of the distribution on the three axis is approximated to be $\sigma_x = 0.020$ m/s^2, $\sigma_y = 0.020$ m/s^2 and $\sigma_z = 0.028$ m/s^2.

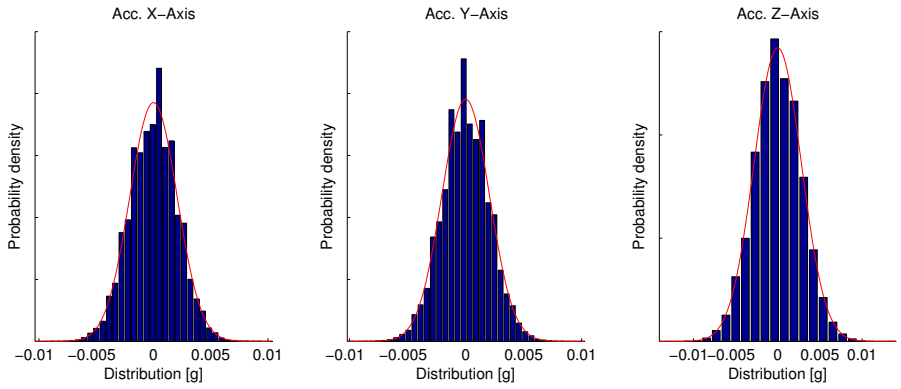


Figure 4.3: Distribution of noise

Observation model for the accelerometer measurements

Acceleration in the local plane, the ENU frame, is measured in the body orientation of the accelerometer. The measurement of the acceleration is compared with the actual acceleration in the same coordinate frame, the body coordinate frame. Therefore, the real acceleration in the point of measurement needs to be rotated to the body frame using the quaternion state.

$$\begin{pmatrix} 0 \\ \mathbf{a_b^m} \end{pmatrix} = \mathbf{q}^{-1} \otimes \begin{pmatrix} 0 \\ \mathbf{a^w} \end{pmatrix} \otimes \mathbf{q} + \mathbf{v_{acc}} \tag{4.11}$$

$$\mathbf{v_{acc}} \sim \mathcal{N}(0, \sigma_{acc}^2) \tag{4.12}$$

The accelerometer is assumed to be calibrated and $\mathbf{b_a}$ and $\mathbf{S_a}$ are therefore neglected.

4.3.2 Gyroscope

The rate gyroscope measures the angular velocity in the body frame of the sensor. When the gyroscope is rotated, the Coriolis effect causes a vibration that is detected by capacitive pickoff. The resulting signal is amplified and filtered to produce a voltage proportional to the angular rate. The voltage is further digitized using the chips 16-bit analog to digital converter. The full scale of the MPU-6050 can be adjusted to ±250,500,1000,2000 degrees per second. The scale of the gyroscope is chosen to ±500 deg/sec. This gives a relatively good resolution given the digital 16-bit output, and a range that is within the application. Gyroscope are modelled in the body frame as seen in Equation (4.13).

$$\omega_{\mathbf{b}}^{\mathbf{m}} = \mathbf{S_g}\omega_{\mathbf{b}} + \mathbf{b_g} + \mathbf{n_g} \tag{4.13}$$

The $\mathbf{S_g}$ may be used to indicate the scaling of the sensors and the cross-axis correlation. Ideally, $\mathbf{S_g}$ would be an identity matrices $\mathbf{I_{3\times 3}}$. The gyroscope is modelled as $\omega_{\mathbf{b}}$ which is the angular velocity in the body frame, $\mathbf{b_g}$ a slow varying offset and $\mathbf{v_{gyro}}$ the sensor noise.

The sensors are subjected to thermal drift and have a zero offset between ±20 degrees per second. This can be calibrated for, but problem arises when the ambient temperature changes. Subjecting gyroscope units to changes in temperature shifts the zero offset seen in Figure A.2.

The distribution of the sensor measurements with no rotation is seen in Figure 4.4. Approximating the distribution using a normal distribution indicates a standard deviation of $\sigma_x = 0.040$ deg/s $\sigma_y = 0.038$ deg/s $\sigma_z = 0.036$ deg/s.

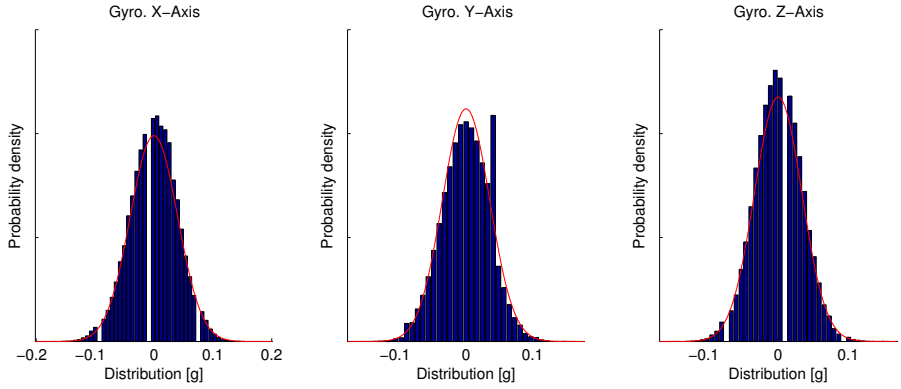


Figure 4.4: Distribution of noise

Observation model for the gyroscope measurements

The gyroscope are modelled using a slow varying bias, b_g and a normally distributed noise v_{gyro}. The slowly varying bias changes over time, and is modelled using the normally

distributed change rate of n_b with standard deviation σ_{bias}.

$$\boldsymbol{\omega}^{\boldsymbol{m}} = \boldsymbol{\omega} + \mathbf{b_g} + \mathbf{v_{gyro}} \tag{4.14}$$

$$\dot{\mathbf{b}_\mathbf{g}} = \boldsymbol{v_b} \tag{4.15}$$

$$\mathbf{v_{gyro}} = \mathcal{N}(0, \sigma^2_{gyro}) \tag{4.16}$$

$$\boldsymbol{v_b} = \mathcal{N}(0, \sigma^2_{bias}) \tag{4.17}$$

4.4 State transition model

The state evolve in time, and are expected to follow the following model seen in Equations (4.18) to (4.22). The quaternion derivative is expressed in Section 3.5.2, and the derivative of the angular velocity is the angular acceleration. The process noise $\boldsymbol{w_q}$, $\boldsymbol{w_\omega}$, $\boldsymbol{w_\alpha}$ and $\boldsymbol{w_b}$ are expected to be normally distributed.

$$\boldsymbol{q_r} = \begin{pmatrix} 0 & \omega^x & \omega^y & \omega^z \end{pmatrix}^T \tag{4.18}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{q_r} + \boldsymbol{w_q} \tag{4.19}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha} + \boldsymbol{w_\omega} \tag{4.20}$$

$$\dot{\boldsymbol{\alpha}} = \boldsymbol{w_\alpha} \tag{4.21}$$

$$\dot{\boldsymbol{b}} = \boldsymbol{w_b} \tag{4.22}$$

where

$$\boldsymbol{w_q} \sim \mathcal{N}(0, \boldsymbol{Q_q}) \qquad \boldsymbol{w_\omega} \sim \mathcal{N}(0, \boldsymbol{Q_\omega}) \tag{4.23}$$

$$\boldsymbol{w_\alpha} \sim \mathcal{N}(0, \boldsymbol{Q_\alpha}) \qquad \boldsymbol{w_b} \sim \mathcal{N}(0, \boldsymbol{Q_b}) \tag{4.24}$$

4.5 Communication

The IMUs communicate with each other over a controller area network (CAN). This comes with limitations since there are restrictions in how much can be transmitted between the IMUs.

Knowing that the CAN is running on 500 kbps, and each frame being maximum 135 bits a maximum of 500 kbps/135 bits/frame = 3703 frames/second can be transmitted. The filter to runs at 100 Hz, mean it has a cycle time of 10 ms. The maximum if 3703 frames/ second gives 37 frames/cycle. With five sensors in the network, the possibility of 37 frames/5 sensors = 7 frames/(sensor and cycle).

In order to achieve the required resolution, each float needs to be represented by a minimum of 2 bytes. This results in 7 frames/(sensor and cycle) × 8 bytes/frame × 1 float/ 2 bytes = 28 floats/(sensor and cycle).

Information needs to be communicated at different times, such as settings during the start up and different debug frames at other times. On a cycle basis, the orientation, the

angular velocity and the angular acceleration of the sensor needs to be communicated. The orientation can be transmitted as a Tait-Bryan angles or as a quaternion. Finally, the predicted acceleration at a point of the next sensor with corresponding uncertainty needs to be communicated on the bus. A suggestion for what data are to sent is seen in Table 4.2.

Table 4.2: The data to be sent over CAN

Description	Number of floats
Quaternion	4
Angular acceleration	3
Angular velocity	3
Predicted acceleration	3
Covariance matrix of p. acceleration	9
Sum	22

5

Proposed filters

In the following chapter, the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) are explained further and related to the models of the system.

5.1 Extended Kalman filter

The extended Kalman filter (EKF) is implemented as described in Section 3.7.3 with an additional term to the innovation matrix. The transition model propagates the state vector from one timestep to the next. The quaternion evolves by constructing a quaternion derivative of the angular velocity, which is integrated in discrete time.

The EKF is implemented with the state vector shown in Equation (5.1). The state transition model describes how the state vector evolve in time. An observation model, using both the state vector and the predicted acceleration received from another IMU, maps the state vector into the observations of the system. The roll of the first IMU is used as a measurement of the roll of the entire system IMUs.

$$\boldsymbol{x} = \begin{pmatrix} \mathbf{q} & \omega & \alpha & \mathbf{b} \end{pmatrix}^T \tag{5.1}$$

5.1.1 State transition model

The state transition model propagates the previous estimated state, $\hat{x}_{k-1|k-1}$, one timestep forward to predict $\hat{x}_{k|k-1}$. A function describing how the quaternion, angular velocity, angular acceleration and the bias of the gyroscope evolve in time needs to be formed. The function f that takes the previous state vector as an input, and propagates the state to form the predicted state one timestep forward needs to be constructed, shown

in Equation (5.2).

$$\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}} = f(\hat{\boldsymbol{x}}_{\boldsymbol{k-1|k-1}}) = \begin{pmatrix} \hat{\boldsymbol{q}}_{\boldsymbol{k|k-1}} \\ \hat{\boldsymbol{\omega}}_{\boldsymbol{k|k-1}} \\ \hat{\boldsymbol{\alpha}}_{\boldsymbol{k|k-1}} \\ \hat{\boldsymbol{b}}_{\boldsymbol{k|k-1}} \end{pmatrix} \tag{5.2}$$

The quaternion derivative is estimated, and integrated to form the predicted quaternion at time k given the estimation at time $k-1$ [1]. The addition of the derivative may cause problems, as the quaternion is not subjected to addition and subtraction, and will not stay a unit quaternion. As such, the quaternion needs to be normalized after each iteration not to diverge.

$$\hat{\boldsymbol{q}}_{\boldsymbol{r}} = \begin{pmatrix} 0 & \hat{\omega}^x_{k-1|k-1} & \hat{\omega}^y_{k-1|k-1} & \hat{\omega}^z_{k-1|k-1} \end{pmatrix}^T \tag{5.3}$$

$$\dot{\hat{\boldsymbol{q}}}_{\boldsymbol{k|k-1}} = \frac{1}{2}\hat{\boldsymbol{q}}_{\boldsymbol{k-1|k-1}} \otimes \hat{\boldsymbol{q}}_{\boldsymbol{r}} \tag{5.4}$$

$$\hat{\boldsymbol{q}}_{\boldsymbol{k|k-1}} = \hat{\boldsymbol{q}}_{\boldsymbol{k-1|k-1}} + \dot{\hat{\boldsymbol{q}}}_{\boldsymbol{k-1|k-1}}\Delta t \tag{5.5}$$

Angular velocity, acceleration and drift of the gyroscope are propagated using the models in Section 4.4 expressed in discrete time.

$$\hat{\boldsymbol{\omega}}_{\boldsymbol{k|k-1}} = \hat{\boldsymbol{\omega}}_{\boldsymbol{k-1|k-1}} + \hat{\boldsymbol{\alpha}}_{\boldsymbol{k-1|k-1}}\Delta t \tag{5.6}$$

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{k|k-1}} = \hat{\boldsymbol{\alpha}}_{\boldsymbol{k-1|k-1}} \tag{5.7}$$

$$\hat{\boldsymbol{b}}_{\boldsymbol{k|k-1}} = \hat{\boldsymbol{b}}_{\boldsymbol{k-1|k-1}} \tag{5.8}$$

5.1.2 Observation model

The observation model maps the predicted state vector into the observations of the system. This way, the distribution of the predicted state vector and the observation of the system can be combined to form an estimate with a lower variance of the system.

The accelerometers sense all types of accelerations, meaning that they sense gravity as well as centrifugal and tangential acceleration. During motion the observed accelerations will not be limited to solely gravity. As such, the system of sensors predict the distribution of likely acceleration in a point near the location of neighbouring IMUs to allow the filter to have better understanding of the external accelerations. The covariance matrix of the predicted acceleration is propagated through the linearised prediction function to find the probability density in the body frame. Since the distribution of the predicted acceleration is not additive, an extension to the innovation matrix needs to be formed.

From a sensor closer to the house, a sensor receives the predicted acceleration distribution $\hat{\boldsymbol{a}}^{\boldsymbol{enu}} \sim \mathcal{N}(\boldsymbol{\mu}_{\hat{\boldsymbol{a}}}, \sigma_a^2)$. Further, the predicted acceleration in the ENU coordinate system is rotated to the body coordinate system, as described in Section 4.3.1. The

measurement model of the accelerometer is seen in Equation (5.9), where $\boldsymbol{a_r^b}$ consists of the centrifugal and tangential acceleration experienced due to the lever arm between the joint and the sensor. $\boldsymbol{a_r^b} = \hat{\omega}_{k-1} \times (\hat{\omega}_{k-1} \times r) + \hat{\alpha}_{k-1} \times r$, where r is the distance between the joint and the sensor and $\hat{\omega}_{k-1}$ and $\hat{\alpha}_{k-1}$ are the *a posteriori* estimated angular velocity and angular acceleration respectively of the sensor.

$$\begin{pmatrix} 0 \\ \hat{\boldsymbol{a}}^{\boldsymbol{b}} \end{pmatrix} = \hat{\boldsymbol{q}}_{\boldsymbol{k|k-1}}{}^{-1} \otimes \begin{pmatrix} 0 \\ \boldsymbol{\mu_{\hat{a}}} \end{pmatrix} \otimes \hat{\boldsymbol{q}}_{\boldsymbol{k|k-1}} + \begin{pmatrix} 0 \\ \boldsymbol{a_r^b} \end{pmatrix} \tag{5.9}$$

$$\hat{\boldsymbol{\omega}} = \hat{\boldsymbol{\omega}}_{\boldsymbol{k|k-1}} + \hat{\boldsymbol{b}}_{\boldsymbol{k|k-1}} \tag{5.10}$$

$$\hat{e}_r = 2(\hat{q}^y_{k|k-1}\hat{q}^z_{k|k-1} + \hat{q}^w_{k|k-1}\hat{q}^x_{k|k-1}) \tag{5.11}$$

The measurement model of the gyroscope shown in Equation (5.10), is modelled similar to that of the system in Section 4.3.2. The roll is estimated using the estimation of the quaternion, seen in Equation (5.11), for all IMU but the one mounted closest to the house.

$$\hat{\boldsymbol{z}}_{\boldsymbol{k}} = h(\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}) = \begin{pmatrix} \hat{\boldsymbol{a}}^{\boldsymbol{b}} \\ \hat{\boldsymbol{\omega}} \\ \hat{e}_r \end{pmatrix} \tag{5.12}$$

5.1.3 Improving the innovation matrix

The innovation matrix S_k, is the covariance matrix of the measurement error. The measurement function h is linearised

$$\boldsymbol{z_k} = h_a(\boldsymbol{x_k}, \boldsymbol{a^{enu}}) + \boldsymbol{v_k} \tag{5.13}$$

$$\approx h_a(\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}, \hat{\boldsymbol{a}}^{\boldsymbol{enu}}) + \frac{\partial h_a(\boldsymbol{x}, \boldsymbol{a})}{\partial \boldsymbol{x}} \left(\boldsymbol{x_k} - \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}\right) + \frac{\partial h_a(\boldsymbol{x}, \boldsymbol{a})}{\partial \boldsymbol{a}} \left(\boldsymbol{a^{enu}} - \hat{\boldsymbol{a}}^{\boldsymbol{enu}}\right) + \boldsymbol{v_k} \tag{5.14}$$

By letting

$$H_x = \frac{\partial h_a(\boldsymbol{x}, \boldsymbol{a})}{\partial \boldsymbol{x}} \qquad H_a = \frac{\partial h_a(\boldsymbol{x}, \boldsymbol{a})}{\partial \boldsymbol{a}} \tag{5.15}$$

$$\hat{\boldsymbol{z}}_{\boldsymbol{k}} = h(\hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}, \boldsymbol{a^{\hat{e}nu}}) \tag{5.16}$$

$$\boldsymbol{z_k} - \hat{\boldsymbol{z}}_{\boldsymbol{k}} = H_x \left(\boldsymbol{x_k} - \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}\right) + H_a \left(\boldsymbol{a^{enu}} - \hat{\boldsymbol{a}}^{\boldsymbol{enu}}\right) + \boldsymbol{v_k} \tag{5.17}$$

The innovation matrix S_k is found by

$$S_k = E[(\boldsymbol{z_k} - \hat{\boldsymbol{z}}_{\boldsymbol{k}}) (\boldsymbol{z_k} - \hat{\boldsymbol{z}}_{\boldsymbol{k}})^T] \approx \tag{5.18}$$

$$E[H_x \left(\boldsymbol{x_k} - \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}\right) \left(\boldsymbol{x_k} - \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}\right)^T H_x^T] + \tag{5.19}$$

$$E[H_a \left(\boldsymbol{a^{enu}} - \hat{\boldsymbol{a}}^{\boldsymbol{enu}}\right) \left(\boldsymbol{a^{enu}} - \hat{\boldsymbol{a}}^{\boldsymbol{enu}}\right)^T H_a^T] + E[\boldsymbol{v_k} \boldsymbol{v_k}^T] \tag{5.20}$$

Since the state error and the acceleration error is assumed to be uncorrelated, as well as the measurement noise with the predicted acceleration and the state vector.

$$E[\left(\boldsymbol{x_k} - \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}\right)(\boldsymbol{a^{enu}} - \hat{\boldsymbol{a}}^{\boldsymbol{enu}})^T] = \tag{5.21}$$

$$E[\left(\boldsymbol{x_k} - \hat{\boldsymbol{x}}_{\boldsymbol{k|k-1}}\right)\boldsymbol{v_k}^T] = E[(\boldsymbol{a^{enu}} - \hat{\boldsymbol{a}}^{\boldsymbol{enu}})\,\boldsymbol{v_k}^T] = 0 \tag{5.22}$$

The covariance of the predicted state is $P_{k|k-1}$, and the estimated covariance matrix of the predicted acceleration A. The measurement error is assumed to be known and expressed in R.

$$S_k = H_x P_{k|k-1} H_x^T + H_a A H_a^T + R \tag{5.23}$$

Given the measurement model $h_a(\boldsymbol{q}, \boldsymbol{a}) = \boldsymbol{q}^{-1} \otimes \begin{pmatrix} 0 \\ \boldsymbol{a} \end{pmatrix} \otimes \boldsymbol{q}$, H_a is found to be

$$H_a = \begin{pmatrix} {q_w}^2 + {q_x}^2 - {q_y}^2 - {q_z}^2 & 2\,q_w\,q_z + 2\,q_x\,q_y & 2\,q_x\,q_z - 2\,q_w\,q_y \\ 2\,q_x\,q_y - 2\,q_w\,q_z & {q_w}^2 - {q_x}^2 + {q_y}^2 - {q_z}^2 & 2\,q_w\,q_x + 2\,q_y\,q_z \\ 2\,q_w\,q_y + 2\,q_x\,q_z & 2\,q_y\,q_z - 2\,q_w\,q_x & {q_w}^2 - {q_x}^2 - {q_y}^2 + {q_z}^2 \end{pmatrix} \tag{5.24}$$

5.2 Unscented Kalman filter

The implementation of the unscented Kalman filter (UKF) make use of the fact that a unit quaternion is used, therefore one element is always dependent on the other such that $q_w = \sqrt{1 - ||q_{xyz}||}$.The state vector that describe the system is

$$\boldsymbol{x} = \begin{pmatrix} q^x & q^y & q^z & \omega & \alpha & \mathbf{b} \end{pmatrix}^T \tag{5.25}$$

5.2.1 Sigma points

The process noise is added before the sigma points are propagated through the dynamic function, described in [5, 15]. Rather than using additive noise, noise quaternions are formed are multiplied with the estimate, making the quaternion remain unit quaternion throughout the entire filtering. By letting $\boldsymbol{\zeta}$ describe the distribution of the sigma points, noise quaternions can be formed.

$$\boldsymbol{\zeta} = [\ \gamma\sqrt{P_k + Q_k}\ ] = \begin{bmatrix} \boldsymbol{\zeta_{qxyz}} & \boldsymbol{\zeta_\omega} & \boldsymbol{\zeta_\alpha} & \boldsymbol{\zeta_b} \end{bmatrix} \tag{5.26}$$

The matrix square root $\sqrt{P_k + Q_k}$ can be found by using lower triangular Cholesky factorization. The multiplicative quaternion noise is found by letting $\zeta_{qw} = \sqrt{1 - ||\boldsymbol{\zeta_{qxyz}}||}$.

The sigma points can now be found in the following manner, where $\boldsymbol{\zeta}^i$ is the ith row of the matrix square.

$$\mathcal{X}^0_{k-1|k-1} = \begin{bmatrix} \hat{q}_{k-1|k-1} \\ \hat{\omega}_{k-1|k-1} \\ \hat{\alpha}_{k-1|k-1} \\ \hat{b}_{k-1|k-1} \end{bmatrix} \tag{5.27}$$

$$\mathcal{X}^i_{k-1|k-1} = \begin{bmatrix} \boldsymbol{\zeta}^{\boldsymbol{i}}_{\boldsymbol{q}\,k-1|k-1} \otimes \hat{q}_{k-1|k-1} \\ \hat{\omega}_{k-1|k-1} + \zeta^n_\omega \\ \hat{\alpha}_{k-1|k-1} + \zeta^n_\alpha \\ \hat{b}_{k-1|k-1} + \zeta^n_b \end{bmatrix} \qquad \forall i \in 1..n \tag{5.28}$$

$$\mathcal{X}^i_{k-1|k-1} = \begin{bmatrix} \boldsymbol{\zeta}^{(\boldsymbol{i-n})^{-1}}_{\boldsymbol{q}\,k-1|k-1} \otimes \hat{q}_{k-1|k-1} \\ \hat{\omega}_{k-1|k-1} - \zeta^n_\omega \\ \hat{\alpha}_{k-1|k-1} - \zeta^n_\alpha \\ \hat{b}_{k-1|k-1} - \zeta^n_b \end{bmatrix} \qquad \forall i \in n+1..2n \tag{5.29}$$

5.2.2 State transition

The state transition function describes how the state evolve in time. The estimated angular velocities can form quaternion to describe this motion, and multiplied with the estimated quaternion using quaternion multiplication (Section 3.5.2) to find the predicted quaternion.

$$\mathcal{X}_{q,k|k-1} = \Omega(\hat{\omega}_{k-1|k-1}) \otimes \mathcal{X}_{q,k-1|k-1} \tag{5.30}$$

$$\mathcal{X}_{\omega,k|k-1} = \mathcal{X}_{\omega,k-1|k-1} + \mathcal{X}_{\alpha,k-1|k-1}\Delta t \tag{5.31}$$

$$\mathcal{X}_{\alpha,k|k-1} = \mathcal{X}_{\alpha,k-1|k-1} \tag{5.32}$$

$$\mathcal{X}_{b,k|k-1} = \mathcal{X}_{b,k-1|k-1} \tag{5.33}$$

The quaternion Ω is formed by letting the total angular rotation be the norm of the estimated angular measurements, and the axis of rotation the corresponding vector ω_x, ω_y and ω_z.

$$\Omega(\hat{\omega}_{k-1|k-1}) = \begin{bmatrix} \cos(|\mathcal{X}_{\omega,\mathbf{k-1|k-1}}|\Delta t/2) \\ \frac{\mathcal{X}_{\omega,\mathbf{k-1|k-1}}}{|\mathcal{X}_{\omega,\mathbf{k-1|k-1}}|}\sin(|\mathcal{X}_{\omega,\mathbf{k-1|k-1}}|\Delta t/2) \end{bmatrix} \tag{5.34}$$

There are different ways of finding the weighted sum of the quaternion. Using the weighted sum of the quaternion does not give the exact mean as quaternions are not closed for addition and subtraction. However, assuming small distances between the

rotations the weighted sum of the propagated quaternions yields an approximate solution, although the quaternion needs to be normalised to remain of unit size. The other parameters in the state vector are summed in the normal method using the weight corresponding to each sigma point.

$$\hat{\mathbf{q}}_{k|k-1} = \frac{\sum W_i \mathcal{X}_{q,k|k-1}}{|\sum W_i \mathcal{X}_{q,k|k-1}|} \tag{5.35}$$

The predicted covariance of the predicted state vector is also calculated using quaternion multiplication as the quaternion is not closed for addition and subtraction. To find the predicted covariance, quaternion multiplication is used in the following manner.

$$\Lambda_q = \mathcal{X}_{q,k+1|k} \otimes (\hat{\mathbf{q}}_{k+1|k})^{-1} \tag{5.36}$$
$$\Lambda_\omega = \mathcal{X}_{\omega,k+1|k} - \hat{\omega}_{\mathbf{k+1|k}} \tag{5.37}$$
$$\Lambda_\alpha = \mathcal{X}_{\alpha,k+1|k} - \hat{\alpha}_{\mathbf{k+1|k}} \tag{5.38}$$
$$\Lambda_b = \mathcal{X}_{b,k+1|k} - \hat{\mathbf{b}}_{\mathbf{k+1|k}} \tag{5.39}$$
$$\tag{5.40}$$

Using only the vector part ($\boldsymbol{q_{xyz}}$) of Λ_q to form λ_q, the predicted covariance is found using

$$P_{k|k-1} = \sum_{i=0}^{2n} W_i \begin{bmatrix} \lambda_q & \Lambda_\omega & \Lambda_\alpha & \Lambda_b \end{bmatrix}_i \begin{bmatrix} \lambda_q & \Lambda_\omega & \Lambda_\alpha & \Lambda_b \end{bmatrix}_i^T \tag{5.41}$$

5.2.3 Observation model

The observation model maps the predicted sigma points to the observations. Acceleration at the sensor position in the east north up (ENU) frame, will consist of mainly gravity but also centrifugal and tangential acceleration. The sensors communicate the estimated acceleration at the point of the sensor in the ENU frame $\hat{\mathbf{a}}^{\mathbf{enu}}$, where $\hat{\boldsymbol{a}}^{\boldsymbol{enu}} \sim \mathcal{N}(\boldsymbol{\mu_{\hat{a}}}, \sigma_a^2)$. The accelerometer is used in the measurement equation, where the acceleration measurement is given by

$$\begin{pmatrix} 0 \\ \mathcal{Y}_k^{i,a} \end{pmatrix} = \mathcal{X}_{q,k|k-1}^{(i)-1} \otimes \begin{pmatrix} 0 \\ \mu_{\hat{\mathbf{a}}} \end{pmatrix} \otimes \mathcal{X}_{q,k|k-1}^{(i)}, \tag{5.42}$$

where $\mathcal{X}_{q,k+1|k}^{(i)-1}$ is the quaternion conjugate of the ith sigma quaternion and $\mathcal{Y}_{k+1}^{i,a}$ is the corresponding measurement. Given that there are uncertainties in the predicted acceleration $\hat{\mathbf{a}}^{\mathbf{enu}}$, these needs to be incorporated in the model. This could be accomplished by incorporating the distribution of the predicted acceleration when choosing sigma points, or by a linearising the observation model of the accelerometer as in the EKF implementation. Since the implementation is dependent on the computational cost to be able to

run in real time on embedded systems and estimates of the experienced acceleration are achieved at a later time, the latter is chosen for this implementation.

The mean of the estimated measurements are found by the summation of the propagated sigma point with their corresponding weight, as seen in Equation (5.43). Further, the estimated acceleration is improved by the estimated tangential and centrifugal acceleration between the joint and the IMU, shown in Equation (5.44).

$$\hat{y}_k = \sum_{i=0}^{2n} W_i^m \mathcal{Y}_k \tag{5.43}$$

$$\hat{y}_k^a = \hat{y}_k^a + \boldsymbol{a_r^b} \tag{5.44}$$

Similar to the EKF implementation, $\boldsymbol{a_r^b} = \hat{\omega}_{k-1} \times (\hat{\omega}_{k-1} \times r) + \hat{\alpha}_{k-1} \times r$. r is the distance between the joint and the sensor and $\hat{\omega}_{k-1}$ and $\hat{\alpha}_{k-1}$ are the *a posteriori* estimated angular velocity and angular acceleration respectively of the sensor.

As such, the quaternion estimate $\hat{\mathbf{q}}_{k|k-1}$ is used together with the covariance matrix of the predicted acceleration, with the linearisation of the measurement equation H_a as expressed in Section 5.1.3. Monte Carlo simulations indicates very small differences between the simulated variance, estimated variance from the the linearized model and the unscented transform (Figure C.1) [20]. The observation model describing the gyroscope, and the roll is seen in Equations (5.45) and (5.46).

$$\mathcal{Y}_k^{\omega} = \mathcal{X}_{\omega,k|k-1}^{(i)} + \mathcal{X}_{\alpha,k|k-1}^{(i)} \Delta t \tag{5.45}$$

$$\mathcal{Y}_k^{r} = 2\left(\mathcal{X}_{qy,k|k-1}^{(i)} \mathcal{X}_{qz,k|k-1}^{(i)} + \mathcal{X}_{qw,k|k-1}^{(i)} \mathcal{X}_{qx,k|k-1}^{(i)}\right) \tag{5.46}$$

Calculation of the covariance matrix for the measurement equation is seen in Equation (5.47), and the cross correlation matrix of the state transition model and the measurement model is seen in Equation (5.48).

$$P_{k|k-1}^{yy} = \sum_{i=0}^{2n} W_i \left[\mathcal{Y}_k^a - \hat{\mathbf{y}}_k^-\right]_i \left[\mathcal{Y}_k^a - \hat{\mathbf{y}}_k^-\right]_i^T \tag{5.47}$$

$$P_{k|k-1}^{xy} = \sum_{i=0}^{2n} W_i \begin{bmatrix} \lambda_q & \Lambda_\omega & \Lambda_\alpha & \Lambda_b \end{bmatrix}_i \left[\mathcal{Y}_k^a - \hat{\mathbf{y}}_k^-\right]_i^T \tag{5.48}$$

The innovation matrix P_k^{vv} is seen in Equation (5.49), and the Kalman gain is computed as in Equation (5.50).

$$P_k^{vv} = P_k^{yy} + H_a A H_a^T + R \tag{5.49}$$

$$\mathcal{K}_k = P_k^{xy} (P_k^{vv})^{-1} \tag{5.50}$$

The estimated quaternion is given by

$$\hat{\boldsymbol{q}}_{k|k} = \delta \boldsymbol{q}_k \otimes \hat{\boldsymbol{q}}_{k|k-1}, \tag{5.51}$$

where the correction step of the quaternion is found by using the vector part of the quaternion to the form scalar part shown in Equation (5.53).

$$\begin{pmatrix} \delta\mathbf{q}_{xyz} \\ \boldsymbol{\delta\omega} \\ \boldsymbol{\delta\alpha} \\ \boldsymbol{\delta b} \end{pmatrix} = \mathcal{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{5.52}$$

$$\delta q_w = \sqrt{1 - ||\delta\mathbf{q}_{xyz}||} \tag{5.53}$$

$$\delta\mathbf{q}_k = \begin{bmatrix} q_w & \delta\mathbf{q}_{xyz} \end{bmatrix} \tag{5.54}$$

The vector $\mathbf{y}_k$ is composed of the measurement of the accelerometer, the gyroscope and the estimated roll of the arm. Empirical studies shows that the covariance of the roll does not change significantly in the region of interest, and is estimated to a fixed value.

6

Results

The results are based on the simulated data of an representative excavator model, and evaluated on the state estimate of orientation, angular velocity and angular acceleration as compared with the true reference.

6.1 Evaluation platform

The performance of the different filters are evaluated in Matlab®based on generated data from Simulink®. The evaluation platform is implemented in Simulink using SimMechanics. Each link of the excavator is assign a geometry, and movement is simulated resulting in data on acceleration, velocities and orientations.

The excavator is mounted on wheels, which are modelled as a parallel spring and a damper system. Hence, when the excavator arm is extended a resulting motion is seen throughout the entire machine. Between the wheels and the house of the excavator is a rotating platform, allowing the house and arm to rotate relative the platform. The arm of the excavator consists of a boom in two parts, a stick and a bucket. All the joints in the arm are one degree revolute joints. An illustration of the model is shown in Figure 6.1.

Five sensors are positioned on the machine. The first sensors is mounted on the house, the second on the first boom, the third sensor is mounted on the second boom. Following in the same plane is the fourth sensor mounted on the stick and the fifth sensor mounted on the bucket. To reduce introduced accelerations due to rotations, the sensors are positioned as close to the joints as possible. Because of technical reasons, it is believed that the distance between a sensor and a joint can not be smaller than 0.5 m.

The system is modelled in the local east north up (ENU) frame, with gravity of $g = 9.82 m/s^2$. The accelerometers sense the gravity as well as the tangential and centrifugal acceleration due to movement. The gyroscopes sense the angular velocity in the body frame, and are modelled with a moving zero-offset.

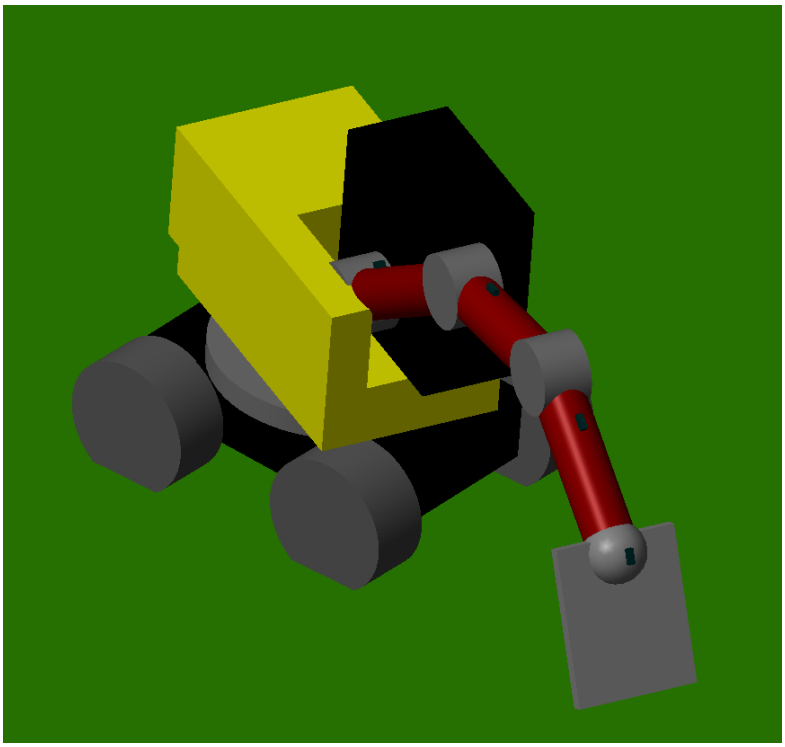


Figure 6.1: Excavator model in Simulink with sensors positioned along the arm

The accelerometer measurement $\mathbf{a}^m$ is modelled by the centrifugal acceleration $\mathbf{a_c}$ and the tangential acceleration $\mathbf{a_t}$ and the gravitation $\mathbf{g}$. The measurement noise n_a are comes from samples drawn at random from the normal distribution $\mathcal{N}(0, \sigma_a^2)$ with $\sigma_a = 0.02g$ m/s^2.

$$\mathbf{a}^m = \mathbf{a_c} + \mathbf{a_t} + \mathbf{g} + n_a \tag{6.1}$$

$$n_a \sim \mathcal{N}(0, \sigma_a^2) \tag{6.2}$$

The modelled gyroscope measures the angular velocity ω, which is bias by the angular random walk b and measurement noise n_ω. The measurement noise is drawn at random from the normal distribution $\mathcal{N}(0, \sigma_g^2)$ and the rate of change of the angular random walk disturbance n_b is drawn at random from $\mathcal{N}(0, \sigma_b^2)$. The standard deviation of the normal distributions are $\sigma_g = 0.2s$ deg/s and $\sigma_b = 1e-6$ deg/s^2. The initial value of the drift b_0, the zero offset of the gyroscope, is drawn at random from the normal distribution $\mathcal{N}(0,1)$.

$$\omega^m = \omega + b + n_\omega \tag{6.3}$$

$$b_{k+1} = b_k + n_b \Delta t \tag{6.4}$$

$$n_\omega \sim \mathcal{N}(0, \sigma_g^2) \tag{6.5}$$

$$n_b \sim \mathcal{N}(0, \sigma_b^2) \tag{6.6}$$

The references of the true state is given by the simulation platform, where the orientation of the link, the angular velocity and the angular acceleration are provided by Simulink. The simulated excavator performs movement in all joints, and moves in a digging pattern where angular velocities of 60 °and angular accelerations of more than 100 °are recorded. The excavator is modelled as in Section 4.2, where the geometry parameters set to $p_x = 1.50$ m, $p_z = 0.5$ m, and the lengths of the excavator arm to $l_1 = 4.00$ m, $l_2 = 3.00$ m, $l_3 = 3.00$ m and $l_4 = 1.00$ m.

6.2 Comparison of the filters

The function of the filter implementation is to provide accurate orientation estimates, with accurate estimates of the angular velocity and acceleration of linkage. With background of this, the filter implementations are evaluated based on their accuracy to estimate the state vector. The bias of the gyroscope is a measure of how well the implementation would handle drift in the gyroscope, and is considered necessary to estimate accurate angular velocities and acceleration and is therefore included in the evaluation.

Different quaternion representation can describe the same orientation in roll, pitch and yaw (e.g. $\boldsymbol{q}$ describes the same rotation as $\boldsymbol{q}$), therefore the quaternion is converted to the Tait-Bryan representation before comparison. The most important angle to evaluate is the pitch of each link. The yaw angle is not considered, as the integration of solely gyroscope is not considered accurate. Other sensors, such as magnetometer and GPS, can estimate the heading of the machine more accurately.

Evaluated parameters

- Angular velocity
- Angular acceleration
- Zero offset of gyroscope
- Pitch and roll

Further, the computational load will be considered since the system is to be implemented on microcontrollers. The two filter implementations have been evaluated in Matlab®and there are therefore no information on how long the computation time is for one iteration of the filter implementations. However, the computation time in Matlab®will be considered when comparing which implementation requires more processing power.

6.2.1 Evaluation of angular velocity, acceleration and bias compensation estimation

The two filters are implemented in different manners, requiring different estimates of the process noise, see Chapter 5. Therefore, not only are the filter implemented using different quaternion representation but their respective process noise are different. The

different noise levels combined with the different models result in different accuracy when estimating the state parameters of the system.

EKF estimates

The EKF estimates the angular velocity accurately, with a lower noise level than that of the measurements. Seen in Figure 6.2, the angular acceleration is observed well, but delay by 0.11 seconds. However, neglecting the delay of the estimate, the observation of angular acceleration is very accurate with a low noise level. In the short simulation, the uncertainty of the zero offset of the gyroscope is increased to allow the algorithm to estimate the zero offset within the time span. The filter estimates the drift quickly for all three axis of the gyroscope, and can from the time the drift is estimated accurately estimate the angular velocities.

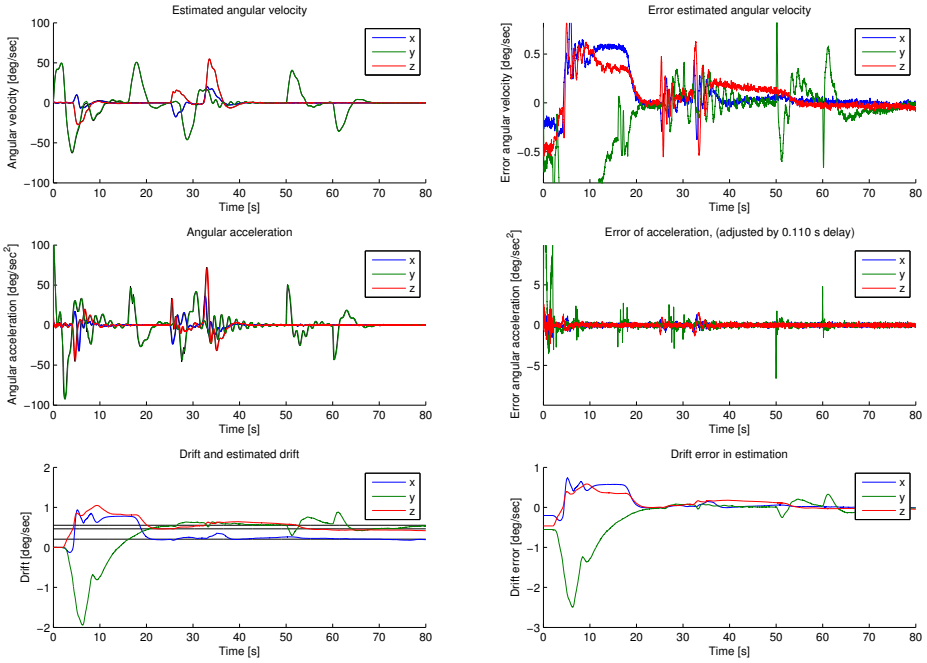


Figure 6.2: Angular estimates of EKF implementation

UKF estimates

Little difference is seen in comparing the estimates of the EKF with the ones of the UKF implementation. Similar to the EKF implementation, the uncertainty regarding the zero offset of the gyroscope are increased to model larger changes in the zero offset estimate. This results in an estimate with larger variance, which allows the filter implementation

to accurately estimate the zero offset in a short time span. A figure of the angular estimations are seen in Figure 6.3. Once the drift of the gyroscope is accurately estimated, the estimated angular velocities are accurate and delivered with less noise than that of the measurements. The observed angular accelerations are accurate, but delayed by 0.08 seconds. The lower delay in the observations could be explained by the slightly higher variance of the angular acceleration estimates, making the system brisk for changes.

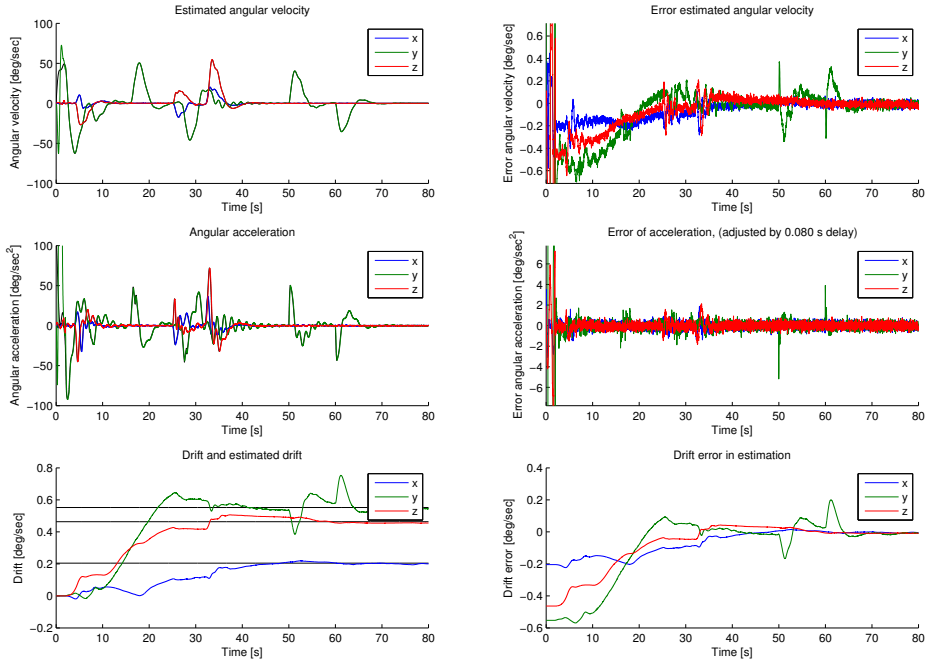


Figure 6.3: Angular estimates of UKF implementation

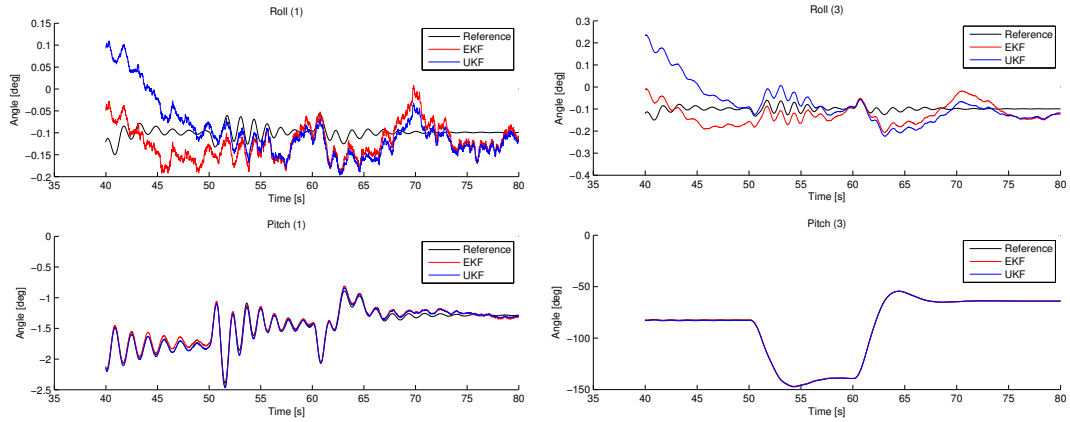
6.2.2 Evaluation of quaternion estimate

The quaternion estimates are converted to Tait-Bryan angles using the equivalent rotation matrices described in Section 3.5.2. Seen in Figure 6.4 are the angles of roll and pitch corresponding to sensors 1,3,4 and 5. The first sensor is mounted on the house and experiences mainly gravity. Other accelerations are not evident thanks to the relative short lever arm. The roll and pitch for both the EKF and UKF implementation corresponding to the first sensor are shown in Figure 6.4a.

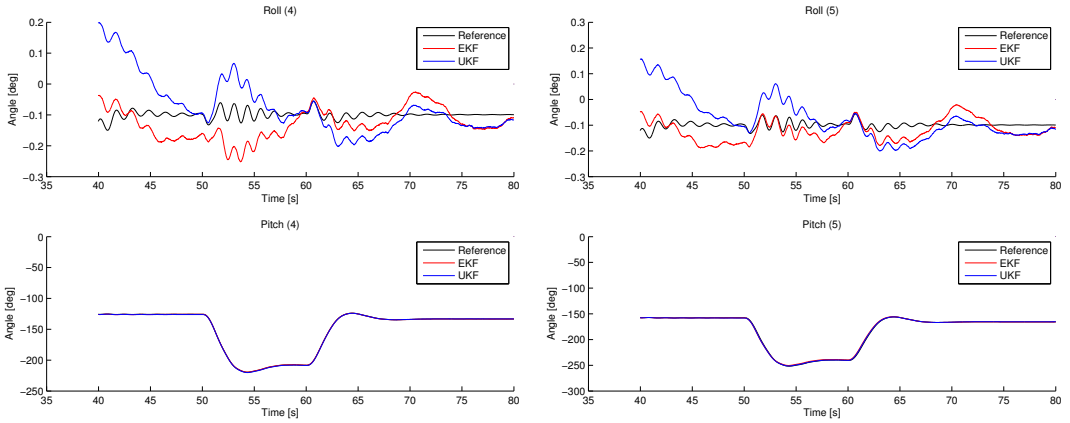
The first sensors propagate the expected acceleration to the joint of the next link, where the second sensor is attached. Further, expecting to have the same roll throughout the entire arm, the roll of the first sensor is communicated to the entire system of sensors as it is the most accurate roll estimate.

The second sensor propagate the expected acceleration at the joint connecting the

links where sensors 2 and 3 are mounted. In this excavator model, it is the joint between the first and the second boom of the excavator arm. The third IMU uses the information about the expected acceleration and estimated roll of the system, to estimate its state vector. The estimated angles for the EKF and UKF implementation are shown in Figure 6.4b for the third IMU in line.



(a) Roll, pitch and reference angle,IMU #1 **(b)** Roll, pitch and reference angle,IMU #3



(c) Roll, pitch and reference angle,IMU #4 **(d)** Roll, pitch and reference angle,IMU #5

Figure 6.4: Results for sensor

Similar to the previous sensors, the fourth IMU receives information about the roll from the first IMU, and the expected acceleration from the previous IMU. The roll and pitch for the two filter implementations are shown in Figure 6.4c for IMU 4 and in Figure 6.4d for the fifth IMU mounted on the bucket.

Error in the estimates

The error of the estimates are evaluated based on the RMSE and peak absolute error (PAE) shown in Equations (6.7) and (6.8). Since both implementations starts from a

poor initial state estimate[1], both implementations are allowed 40 seconds to estimate the orientation, angular velocity and acceleration, and the bias of the gyroscope. As such, the error measurements are calculated from $t = 40$ to $t = 80$.

$$E_{rms} = \sqrt{\frac{\sum_{i=0}^{n}(\theta - \hat{\theta})^2}{n}} \tag{6.7}$$

$$E_{pa} = \max_{i \in 0,..n} \left(||\theta_i - \hat{\theta}_i|| \right) \tag{6.8}$$

The error measurements of the roll for each IMU are shown in Table 6.1. The error measurements are very low, indicating a very good estimate for both EKF and the UKF implementation. However, the PAE is higher even from the first unit for the UKF implementation, indicating an error that have been propagated throughout the arm. As a result, the UKF shows poorer estimate for every unit and the complete system.

Table 6.1: Error measures for the roll

IMU #	RMSE EKF	RMSE UKF	PAE EKF	PAE UKF
1	0.044575	0.065868	0.105450	0.224818
2	0.049456	0.093597	0.093965	0.323034
3	0.051201	0.094496	0.107997	0.353893
4	0.058504	0.093837	0.140772	0.318587
5	0.045031	0.083805	0.091595	0.276536

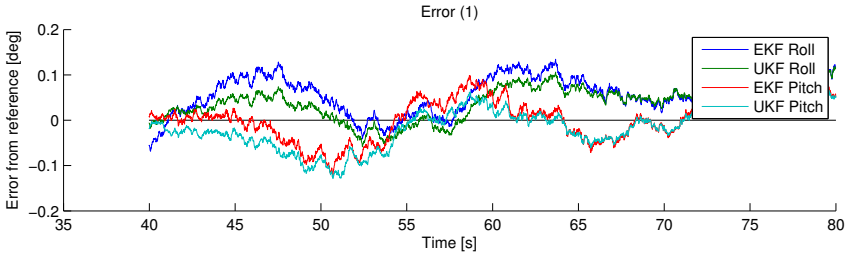
Shown in Table 6.2 are the error measurements for the pitch of each IMU. The first IMU implemented with the UKF algorithm indicates a more accurate estimate than the EKF implementation, when observing the RMSE even if the PAE is higher.

Unit two, three and four indicates a better pitch estimate for the EKF implementation both in term of RMSE and PAE. Surprisingly, the last IMU implemented with UKF estimates the pitch better in terms of the error measurements. Visual figures of the error over 40 seconds are shown in Figures 6.5a to 6.5e.

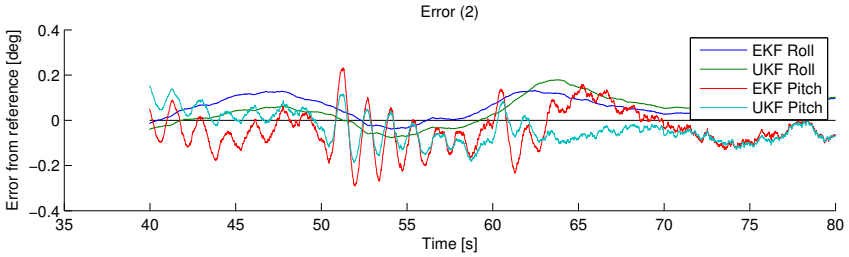
6.2.3 Computation time

The matrix inverse of a 7×7 matrix needs to be computed in every iteration of the EKF implementation, whereas the UKF implementation uses 27 sigma points that are propagated through both the state transition model and the observation model. Further, the Jacobian of both the state transition model and the observation model needs to be evaluated on every iteration of the EKF implementation. Given the simplified state transition model, it is found that the EKF implementation is evaluated in roughly 1/3 of the time compared with the UKF implementation when both filters are evaluated in Matlab®.

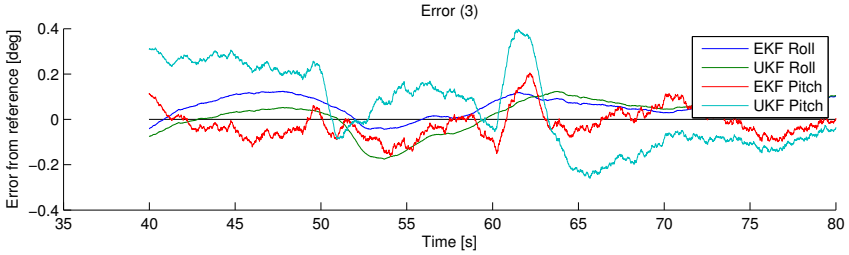
[1] All IMUs are initiated with the state $\boldsymbol{q} = [1\,0\,0\,0]$, $\boldsymbol{\omega} = [0\,0\,0]$,$\boldsymbol{\alpha} = [0\,0\,0]$ and $\boldsymbol{b} = [0\,0\,0]$



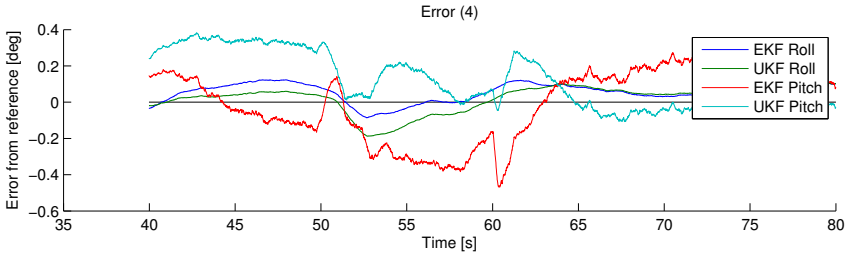
(a) Error in roll and pitch for IMU #1



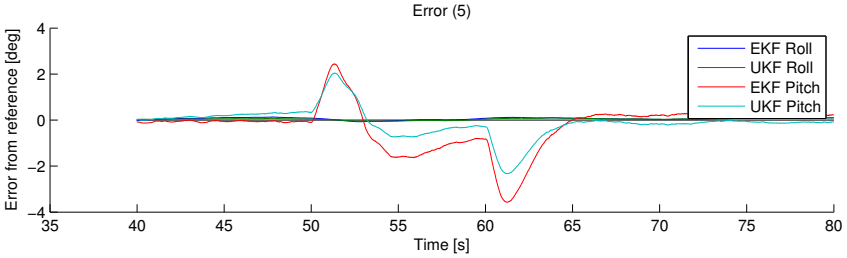
(b) Error in roll and pitch for IMU #2



(c) Error in roll and pitch for IMU #3



(d) Error in roll and pitch for IMU #4



(e) Error in roll and pitch for IMU #5

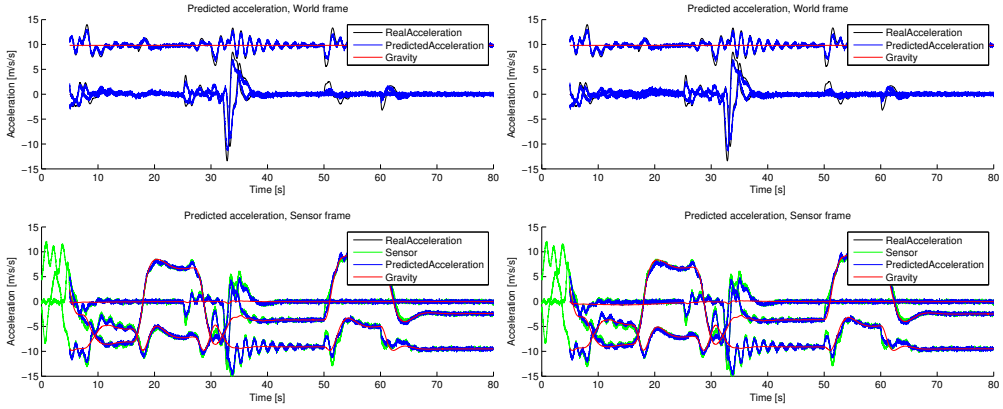
Figure 6.5: Error in roll and pitch for the IMUs in the system

Table 6.2: Error measures for the pitch

IMU #	RMSE EKF	RMSE UKF	PAE EKF	PAE UKF
1	0.043370	0.037026	0.098173	0.107770
2	0.099701	0.122752	0.355002	0.342053
3	0.142804	0.340670	0.301684	0.631387
4	0.188284	0.312224	0.479953	0.519715
5	1.054196	0.627550	3.665617	2.231611

6.3 Prediction of acceleration

Prediction of the acceleration in the position of the next sensor works well. Shown in the Figures 6.6a and 6.6b are examples of how the predicted acceleration behaves in the EKF and UKF implementation respectively. From the figures it can be interpret that using only the gravitation as a known source of acceleration may not always be sufficient when there is a strict requirement on orientation estimates. The acceleration in the ENU frame is shown in the top row, where it can be seen that the accelerations in the local plane are around zero and the largest acceleration is gravity, perpendicular to the local plane. Accelerations, other than gravity, are large enough to disturb the estimation of orientation relative gravity and can therefore not be neglected. Seen in the figures on the lower row are the accelerations in the body frame of the accelerometers. The system



(a) Prediction of acceleration EKF **(b)** Prediction of acceleration UKF

Figure 6.6: Results for sensor

of IMUs that aid the estimation of the state vector in a distributed system is not solely improved by the aid in predicted acceleration. Since the uncertainty corresponding to the predicted acceleration is communicated, the filter implementations becomes adaptive in involving the measurement from the accelerometer.

7

Discussion

The two implementations uses different state vectors, and state transition functions considering the quaternion representation. Following discussion addresses advantages and the issues related to the implementation and possible improvements for future applications.

7.1 Discussion

The two filters implementations behaves different in different regions, and it is difficult to name one of them better than the other. It is clear that the result differs, and that there are many more configurations and parameters that can be fine-tuned to possibly improve the estimates. The filter needs to be adjusted for the application, and depending on the adjustments different parameters can be estimated more accurately.

The two implementations uses different state transition models, therefore different state transition uncertainties are used. For this reason, it can be seen that the UKF is more brisk in detecting angular acceleration but at the same time estimates an angular acceleration with higher variance. Since the UKF implementation does not estimate the covariance of the scalar part of the quaternion, it may lose some degree of accuracy when estimate the full internal system. Further, some implementation such as assuming that all the sensors are positioned in a plane may be too naive. Given the huge mass an excavator may hold, at a distance of 10 m from the body of the excavator, it is likely that the roll is different throughout the arm.

There are no caution, or change detection implementation, that would notice if something in the system breaks. Fault are propagated from one sensor to the next, resulting in a worse estimations than may otherwise have been estimated. The same goes for mounting errors, or if one sensor is subjected to a collision (which is likely given the environment).

Other sensors can be incorporated in the system. Signals to the hydraulics can be used as inputs to the model and thus achieve an more accurate prediction of how the

system evolves in time. GPS data or magnetometers may be used to receive an estimation of the yaw of the machine, to align all the IMUs in the same plane.

7.2 Future recommendations

The distributed system, as it is now, can accurately predict the gravity when there are no external acceleration in the system. However, it is believed that using solely the gravity vector rather than using predicted gravity vector would improve the system further, as there would actually be no uncertainty about its direction or magnitude. For this, it is recommended to implement some change detection that may detect when the system is subjected to external accelerations or solely gravity and can switch between the using the predicted acceleration when the system is subjected to external accelerations and solely the gravity vector when no other accelerations are detected.

Since the system of IMUs will be mounted on excavators and other heavy construction equipments, the IMUs are subjected to an environment where material may hit the IMUs. If any of the IMUs is misaligned, it may predict the wrong acceleration of other IMUs and thus propagating the error along the arm. For this, error detection is of highly recommended to detect error such as this.

8

Conclusion

The two filter implementations show an overall very similar behaviour in estimating state vector. Since both implementations fulfil criteria of accuracy, the extended Kalman filter is suggested for its ease of use and low computational cost.

The performance of a distributed extended Kalman filter (EKF) implementations and a distributed unscented Kalman filter (UKF) are evaluated, for the use on estimating the orientation, velocities and accelerations of the linkage of heavy construction equipment. A simulation platform consisting of inertial measurement units (IMUs) positioned on an excavator is constructed, on which the two implementations are evaluated and compared to the true reference of the platform, where five IMUs are positioned on different strategic positions on an excavator.

The system of IMUs work in a system, so that each IMU predicts the experienced acceleration of another sensor in the system, together with the covariance matrix of the prediction. To make the system consistent with the filter algorithms, the covariance related to the predicted acceleration are propagated using linearization for the EKF implementation and unscented transform for the UKF implementation. Both implementations indicate an adaptive system, that can accurately estimate the orientation, angular velocities and angular acceleration of an excavator arm using IMUs.

However, given that both implementations fulfils the criteria of estimating the orientation within a certain degree, the EKF is suggested for use for this application. Its ease of use and low computational cost makes it advantageous over the UKF implementation.

Bibliography

[1] Andrle, M. & Crassidis, J. (2012) Geometric Integration of Quaternions. AIAA/AAS Astrodynamics Specialist Conference pp. 1–10.

[2] Axelsson, P. (2011) On Sensor Fusion Applied to Industrial Manipulators. Tech. Rep. 1511, Linköpings Universitet.

[3] Boström, A. & Jansson, P. A. k. (2012) Rigid body dynamics, lecture notes Chalmers University.

[4] Bruyninckx, H. (2010) Robot Kinematics and Dynamics.

[5] Cheon, Y. & Kim, J. (2007) Unscented filtering in a unit quaternion space for spacecraft attitude estimation. IEEE International Symposium on Industrial Electronics, 2007. ISIE 2007 pp. 66–71.

[6] Eberly, D. (2002) Rotation representations and performance issues. Magic Software pp. 1–13.

[7] Faragher, R. (2012) Understanding the basis of the Kalman filter via a simple and intuitive derivation. IEEE Signal Processing Magazine 29(September):128–132.

[8] Fletcher, T. (2009) The Kalman Filter Explained .

[9] Guillou, D. F. (2003) Packaging mems: New manufacturing methodology substantially reduces smart mems costs, <`http://archives.sensorsmag.com/articles/1203/20/main.shtml`>, retrieved 2014-05-27.

[10] Hauberg, S., Lauze, F. & Pedersen, K. (2013) Unscented kalman filtering on riemannian manifolds. Journal of mathematical imaging and vision 46:103–120.

[11] Jazar, R. N. (2007) Theory of Applied Robotics: Kinematics, Dynamics, and Control. Springer Publishing Company, Incorporated.

[12] Jiang, C., Xue, L., Chang, H., Yuan, G. & Yuan, W. (2012) Signal processing of MEMS gyroscope arrays to improve accuracy using a 1st order Markov for rate signal modeling. Sensors (Basel, Switzerland) 12(2):1720–37.

[13] Kaajakari, V. (2009) Practical MEMS: Design of Microsystems, Accelerometers, Gyroscopes, RF MEMS, Optical MEMS, and Microfluidic Systems. Las Vegas, NV: Small Gear Publishing pp. 1–5.

[14] Korvink, J. G. & Paul, O. (2006) MEMs: a Practical Guide to Design, Analysis and Applications. William Andrew, Inc pp. 523–565.

[15] Kraft, E. (2003) A quaternion-based unscented Kalman filter for orientation tracking. Proceedings of the Sixth International Conference of Information Fusion 1:47–54.

[16] Laviola, J. (2003) A comparison of unscented and extended Kalman filtering for estimating quaternion motion. American Control Conference, 2003. Proceedings of the 2003 .

[17] Luo, H. (2002) Integrated Multiple Device CMOS-MEMS IMU Systems and. Tech. rep., Carnegie Mellon University.

[18] Madgwick, S. (2010) An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK) .

[19] Madgwick, S. O. H., Harrison, A. J. L. & Vaidyanathan, A. (2011) Estimation of IMU and MARG orientation using a gradient descent algorithm. 2011 IEEE International Conference on Rehabilitation Robotics (ICORR) 2011:5975346.

[20] Maxson, B., Guba, V. & Savin, A. (2013) Covariance Based Uncertainty Analysis with Unscented Transformation. tair.tomsk.ru IEEE:8–11.

[21] Natale, M. D. (2012) Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice. Springer Publishing Company, Incorporated.

[22] Noureldin, A., Karamat, T. B. & Georgy, J. (2013) Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration. Springer Berlin Heidelberg, Berlin, Heidelberg.

[23] Ochoa, B. & Belongie, S. (2006) Covariance propagation for guided matching. Proceedings of the Workshop on Statistical Methods in Multi-Image and Video Processing (SMVP) pp. 1–12.

[24] Patel, B. P. & Prajapati, J. (2013) Kinematics of mini hydraulic backhoe excavator - part II. International Journal of Mechanisms and Robotic Systems 1(4):261.

[25] Patel, C. & McCluskey, P. (2012) A Characterization of the Performance of MEMS Vibratory Gyroscope in Temperature and Humidity Environments. Transaction on Control and Mechanical Systems pp. 239–244.

[26] Roan, P., Deshpande, N., Wang, Y. & Pitzer, B. (2012) Manipulator state estimation with low cost accelerometers and gyroscopes. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems pp. 4822–4827.

[27] Sabatini, A. M. (2006) Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. IEEE transactions on bio-medical engineering 53(7):1346–56.

[28] Salmond, D. & Gordon, N. (2005) An introduction to particle filters.

[29] Särkkä, S. (2013) Bayesian filtering and smoothing. Cambridge University Press.

[30] Shah, S. V., Saha, S. K. & Dutt, J. K. (2012) Denavit-Hartenberg Parameterization of Euler Angles. Journal of Computational and Nonlinear Dynamics 7(2):021006.

[31] Trimpe, S. & D'Andrea, R. (2010) Accelerometer-based tilt estimation of a rigid body with only rotational degrees of freedom. 2010 IEEE International Conference on Robotics and Automation pp. 2630–2636.

[32] Vihonen, J. & Honkakorpi, J. (2013) Geometry-aided angular acceleration sensing of rigid multi-body manipulator using MEMS rate gyros and linear accelerometers. 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) pp. 2514–2520.

[33] Wikipedia (2014) Geodetic system, <http://en.wikipedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_relationships.svg>, retrieved 2014-05-23.

[34] Woodman, O. (2007) An introduction to inertial navigation. University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696 (696).

[35] Xia, D., Chen, S., Wang, S. & Li, H. (2009) Temperature effects and compensation-control methods. Sensors (Basel, Switzerland) 9(10):8349–76.

[36] Yazdi, N., Ayazi, F. & Najafi, K. (1998) Micromachined inertial sensors. Proceedings of the IEEE 86(8).

[37] Yole Developpement (2011) Next generation of mems gyroscopes and inertial combo sensors from sensordynamics, <http://www.i-micronews.com/news/Generation-MEMS-gyroscopes-inertial-combo-sensors-SensorDyn,6375.html>, retrieved 2014-05-27.

[38] Zhao, D., Lu, M., Zhang, X. & Jiang, Z. (2013) Robot Visual Tracking via Incremental Self-Updating of Appearance Model. International Journal of Advanced Robotic Systems p. 1.

A

Thermal dynamics of sensors

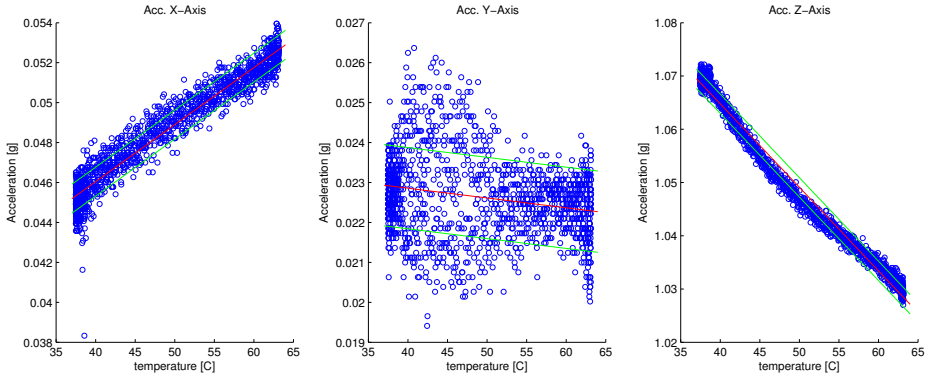


Figure A.1: Thermal drift of the accelerometer. The rate of change over temperature can be described by $(a_x\, a_y\, a_z)^T = (0.0003\ -0.00\ -0.0016)^T \times T + \boldsymbol{b}$ where $\boldsymbol{b}$ is some offset.

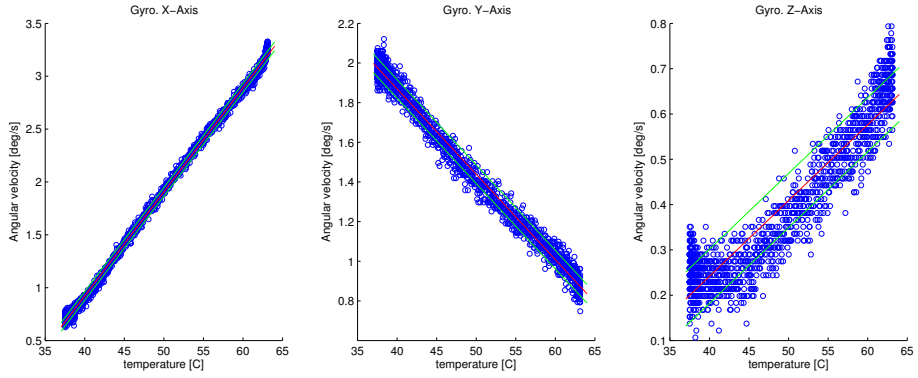


Figure A.2: Thermal drift of the gyroscope. The rate of change over temperature can be described by $(\omega_x\, \omega_y\, \omega_z)^T = (0.0984\ -0.0431\ -0.0167)^T \times T + \boldsymbol{b}$ where $\boldsymbol{b}$ is some offset.

B

Can communication

Table B.1: Table over frame format of J1939 extended, CAN2.0B

Field	Bits	Purpose	Standard
Start of frame	1	Identifies the beginning of a message	0
Identifier A	11	First part of the address	
SRR	1	Substitute remote request	1
IDE	1	Identifier extension bit	1
Identifier B	18	Second part of the address	
RTR	1	Remote transmission request	
ro, r1	2	Reserved bits	0
DLC	4	Data length code, number of bytes in data	
Data field	0-64	Data field, 0-8 bytes	
CRC	15	Cyclic redundancy check	
CRC delimiter	1	Used for error detection	1
ACK slot	1	Transmitter sends ACK for received message	1
ACK delim	1	Must be recessive	1
EOF	7	End of frame	1

Showing the general layout of a data frame of the CAN protocol. The message can contain between 0-8 bytes of data. Sequences of 5 bits of the same type requires the addition of an opposite type of bit by the transmitter. This yields a maximum of 135 bits in one frame, if using 8 bytes of data and requires the maximum of additional bits due to the 5 consecutive bits of the same type [21].

C

Monte Carlo Simulation

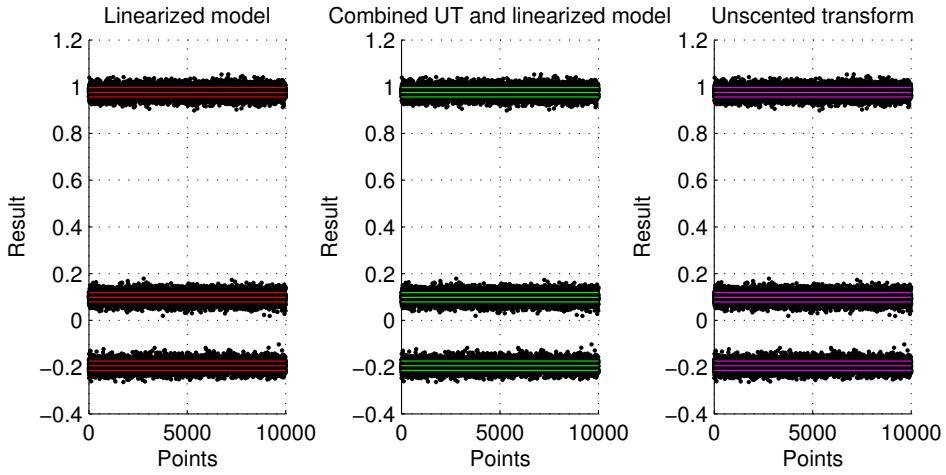


Figure C.1: Comparison between different ways of propagating the covariance of the expected acceleration, with Monte Carlo simulation of 10000 points