EXAMENSARBETE INOM ELEKTROTEKNIK,
AVANCERAD NIVÅ, 30 HP
STOCKHOLM, SVERIGE 2017

Monitoring of doors, door handles and windows using inertial sensors

OSKAR ELIASSON

KTH
SKOLAN FÖR ELEKTRO- OCH SYSTEMTEKNIK

Abstract

The main topics of this master's project are non-linear filtering techniques, system identification, and supervised classification. The main goal is to develop, implement, and test methods using inertial sensors to monitor door- and window movement. Secondly, it aims to investigate if it is possible to monitor door movements by using event classification. The thesis begins with an overview of the available sensors and a theoretical part, in which the filtering algorithms and the modelling of door- and window movements are summarized. Later a supervised learning algorithm is developed. Lastly, the hardware used to test the algorithm and the results from the tests are shown.

The goal of the project is to develop filtering techniques using accelerometers, gyroscopes, and magnetometers. They measure various inertial units of a moving body. By combining measurements from different sensors and fusing them, the movement of a body can be tracked. Door- and window movement are modelled and the Kalman filter is used to estimate the true parameters of the model. Due to the imperfections of the sensors, properties such as offset and drift are included in the models. The models are to be used in doors and windows to track the angle of which they are open. Due to difficulties of tracking the movement when they hit the door- or window frame, the K-Nearest Neighbours classification algorithm is used to predict whether the door- or window closes fully.

A general method of implementing the filtering techniques is constructed. Due to limitations in computational power, limited sampling rate of the sensors, and the nonlinear nature of the movement, some generalizations are required. Hardware which can be placed on doors and windows are used to verify the models. As a first verification, the models and filtering techniques were tested in Matlab. Secondly, the algorithms were tested live on doors and windows. The result shows it is possible to track the movement of doors and windows with an accuracy of over 10°. The classification algorithm shows some success with tracking a closing door but requires more work to become more accurate. The result also shows that the sampling rate has to be taken into consideration when designing the algorithms, a higher sampling rate results in more accurate tracking but increases power consumption and computing power.

Sammanfattning

Huvudämnet i detta examensarbete är ickelinjära filtertekniker, systemidentifikation och handledd klassifiering. Huvudmålet är att utveckla, implementera och testa metoder där accelerometrar, gyroskop och magnetometrar används för att monitorera dörr- och fönsterrörelser. Examensarbetet ämnar även att undersöka möjligheten att upptäcka plötsliga dörrörelser genom att använda klassifiering av händelser. Uppsatsen börjar med en överblick av tillgängliga sensor och en teoretisk del där Kalmanfiltret och modelleringen av dörr- och fönsterrörelser sammanfattas. I den andra delen tas en klassifieringsalgoritm fram. Slutligen visas resultatet från den praktiska delen där hårdvara och sensorer har använts för att utvärdera algoritmerna.

Målet med projektet är att utveckla filtertekniker där accelerometer, gyroskop och magnetometer används. De mäter olika fysiska egenskaper hos en kropp i rörelse. Genom att kombinera sensorerna och fusera data kan dörr- och fönsterrörelser spåras. Pågrund av fel i sensorerna uppstår egenskaper som förskjutning och drift vilket över tid ger felaktiga resultat. Dörr- och fönsterrörelser är ickelinjärt modellerade och det utvidgade Kalmanfilter används för att estimera de riktiga parametrarna i modellerna. Modellerna används mer specifikt för att estimera vinkeln som dörrar och fönster är öppna. På grund av svårigheten med att spåra vinkeln när de slår i karmen utvecklades en K-Nearest Neighbours klassifieringsalgoritm som används för att spåra om dörren eller fönstret stängdes helt.

En generell metod för att implementera filtren används. På grund av begränsningar när det gäller beräkningskraft, samplingsfrekvens och modellernas ickelinjära beteende gjorde vissa generaliseringar. Existerande hårdvara som kunde placeras på dörrar och fönster användes för att verifiera modellerna. Som en första verifikation användes Matlab. För verifiering användes en mikrokontroller försedd med sensorer monterad på dörrar och fönster, där algoritmerna kunde utvärderas. Resultatet är att det är möjligt att spåra vinkeln med en noggrannhet på över 10°. Klassifieringen visar framgång med att estimera om en dörr eller fönster stängdes helt men behöver mer jobb för att bli stabilare. Noggrannheten på resultatet är beroende på beräkningskomplexiteten och samplingsfrekvensen för sensorerna. Högre frekvens ger bättre resultat men är kostsammare och kräver större beräkningskraft.

Contents

Symbols and Abbreviations

Abbreviations

IMU	Inertial Measurement Unit
MEMS	Microelectromechanical system
PDF	Probability Density Function
ML	Maximum Likelihood
ODR	Output Data Rate
SNR	Signal to Noise Ratio
KF	Kalman Filter
EKF	Extended Kalman Filter
KNN	K-Nearest Neighbours
FFT	Fast Fourier Transform

Chapter 1

Introduction

1.1 Background

In access systems such as doors and windows there are many ways of detecting whether they are closed or opened. They can be as simple as an integrated switch or button that monitors the open-closed state of the access system. This is of interest to know because many buildings rely on access systems to give only authorized people access and ensure that unauthorized people does not enter. An expansion of this is to monitor not only whether the door or window is open but also other properties such as the speed and acceleration with which the door was opened, the current angle and position the door or window is open, and if there are problems with the door or window that could case them to break down.

By monitoring these variables it is of interest to see what they can be used for related to access systems and security. This study is done in cooperation with ASSA ABLOY shared technologies. ASSA ABLOY is a world leading company in the field of lock- and door solutions that satisfy the customers demand for safety, security, and user-friendliness. ASSA ABLOY Shared Technologies work with and develop platforms for lock- and door solutions and also develops new technology for electronic access- and lock systems for the future.

The project also involves testing the developed methods on real doors and windows. ASSA ABLOY has worked with different sensor producers and have the required hardware to test these sensors. For this project, micro controller evaluation boards were used to run real time tests and also to acquire data from the sensors to use offline. Development boards such as the freescale platforms from NXP and the Nucleo platform from STMicroelectronics offer easy to use and program products for this purpose. They offer ways to test different sensors through expansion boards.

1.2 Purpose, Goals, and Limitations of the Master Thesis

The main goal of the project is, through the use of inertial sensors such as accelerometers and gyroscopes, to be able to reliably know how open a door or a window is. This should then be tested to see if different behaviours can be identified. These behaviours could be that a door is broken, a door is moving slowly because something is obstructing it, or burglary attempts.

To achieve this, estimation algorithms where data from multiple sensors are used. Different approaches where combinations of accelerometers, gyroscopes, and magnetometer are tried. Due to the complexity of modelling the behaviour of doors and window during the moment they hit their door- and window frame, a novel classification algorithm is used to predict if they close or remain open.

The thesis also include the use of hardware to achieve this. Several factors compose the limitations of this thesis. Among them are cost of the hardware, limited space for hardware, computation limits, and power consumption. A small micro controller which is powered by batteries limits the computational power. By integrating the sensors in door handles it is important to keep size to a minimum.

1.3 Related Work

There are several problems concerning the task at hand. Firstly, low cost inertial sensors such as accelerometers and gyroscopes are becoming increasingly popular but are prone to larger errors than classical sensors that are much more expensive and much larger. To attenuate the faults of the sensors various filtering techniques such as the Kalman Filter (KF) can be used. To use a KF a state space model of the system is required. A popular approach when using inertial sensors is to base the state space model around rotational kinematics. These use roll, pitch, and yaw or angular speed to relate the rotation of the inertial sensors to the system it is mounted on. However, for movement in a plane, it is possible to use simpler models that require fewer sensors.

Sensing when a door hits the door frame is a relatively simple task, knowing if the door is actually fully closed or just resting against the door frame while still open is a more difficult problem. Data mining, a process of finding patterns in large sets of data, have several methods that can be used to discover whether a door is closed or not. They are based on different types of machine learning algorithms that calculate a likelihood of data belonging to a certain set.

The challenge with the project lies in choosing what combination of sensors to use, how to choose the state vector and how to model the system, and what filter methods to use. In the literature the accelerometer measurement for the i-axis of the IMU is commonly modelled as [6], [9], [4]

$$a_i = R_{b,n}(a_{i,real} - g_i) + b_{i,acc} + e_{i,acc} \tag{1.1}$$

This model assumes zero cross correlation for acceleration between the different axes of the IMU. Because of this, the noise is modelled as $e_{i,a} \sim \mathcal{N}(0, \sigma_{i,acc})$. $R_{b,n}$ denotes a rotation between the earth frame to the body's (IMU) frame. The accelerometer measures both earth's gravity g_i and the sensor's acceleration $a_{i,real}$. The bias $b_{i,acc}$ of the accelerometer causes an offset from the real value. If the bias is not taken into consideration errors will arise when the data is integrated to find the position [6].

Gyroscopes measure rotational movement of objects relative to the earth's inertial system. The gyroscope signal is modelled to include an offset, scale factor error, and measurement noise

$$\omega_i = \omega_{i,real} + b_{gyro} + e_{i,gyro} \tag{1.2}$$

Here the noise $e_{i,a} \sim \mathcal{N}(0, \sigma_i)$ and it is assumed to have no cross-coupling with the other axes. One of the most important properties of the gyroscope model is the drift. Drift is unavoidable and the value of the drift depends on a lot of internal properties of the gyroscope. For the gyroscopes in LSM6DS3 [21], a low cost inertial measurement unit, the drift of the gyroscope can be up to $0.2°/s$.

One paper aims to implement a Kalman filter for the real-time estimation of a rigid-body's orientation from acceleration, angular velocity, and magnetic field strength [10]. The problem is solved by using a non-linear variant of the Kalman filter and representing the body's orientation by a 4-dimensional vector called quaternion. An state-space model is derived by using the state space vector with seven components, 4 from the quaternion and 3 from the angular velocity ω,

$$x = \begin{pmatrix} q \\ \vec{\omega} \end{pmatrix}$$

To describe the model it assumes the angular velocity is constant, $\omega_{k+1} = \omega_k$ as it is sufficient for the application. It also exploits known information about the rotation of the system during one sample period and uses a zero-order-hold to derive the non-linear state equation for q. The noise model includes the cross-coupling properties of the sensor, this means that the different axes of the sensors are assumed to influate the measurements of each other. Together, they make up the non-linear process model. The measurement model describes the measured values when they are influenced by measurement noise. Three different models are proposed, each corresponding to the measurements from the accelerometer, magnetometer, and gyroscope and each one are tested separately. A modified version of the UKF was implemented. The modified UKF consists of a measurement update and a time update and is extended to deal with properties of the quaternion. The conclusion of the paper is that the modified UKF improves on the original Kalman filter by dealing with a state vector which is subject to constraints.

Another paper uses a multiplicative error state Kalman filter (MESKF) to predict the movement of an underwater vehicle. The model uses a state vector with position, quaternion, velocity and the biases of the accelerometer and gyroscope [1]. In the derived algorithm, a state vector for the error state which is the errors for each original state is used. The algorithm to predict the movement is divided into 4 stages. In the first step the nominal state is integrated to estimate the vehicle motion. The second step uses the kalman filter to predict the error state values and the covariance matrix. Thirdly, the kalman filter is updated to correct the predicted error state and covariances. In the forth step the computed error state vector is used to adjust the nominal state data.

The model for the nominal state are based on a general unconstricted movement in 3D under constant acceleration and incorporates biases of the sensors in the state. The Kalman filter predicts the new error state, the measurement model for the error state is derived and used for the Kalman filter update. The paper also considers the importance of handling measurement delays. A FIFO (First In First Out) buffer is used to store data together with a time stamp. This way the data can be processed synchronously.

Paper [15] presents a method for estimating the angular position of a system by using accelerometers and gyroscopes. The state vector uses a quaternion to represent the rotation of the body frame. Adding other variables to the state does not increase the precision of the angular estimation significantly. The system parameters are found by using MathWorks Simulink. The algorithm compares the earth's gravity vector to the measured acceleration and is corrected to include the external accelerations. The standard Kalman filter was used for this. Interestingly, the paper does not use the steady state Kalman gain and the goal is to run the algorithm on an integrated system with minimal complexity and power consumption. One way their filter is tested is to see if the drift of the gyroscope can be removed and the result is that the maximum drift error is 0.7°during a 120 second test, when the filter was turned off the gyroscope drifted 30°during one minute.

There are other options other than Kalman filters for sensor fusion. The paper [12] compares the Kalman filter to different adaptive filters such as the Least Mean Squares (LMS) and Recursive Least Square (RLS). They use these methods to monitor human body motion and to determine the position of a subject's limbs. The paper considers the same dynamics of the sensors as the others and implements a state space model from this. The Kalman filter uses the accelerometer to predict the limb's angle and the gyroscope is used to correct the prediction. The filter can be tuned to give a lower degree of trust to dynamic acceleration such as when the subject is moving or running.

Several types of LMS are tested. Firstly, the standard LMS method where an output of a linear filter in response to the input signal is generated. This also generates an estimation error of the output by comparing the output to a desired response. The RLS filter, similarly to the LMS filter, updates its filter coefficients recursively and is typically faster than the LMS but requires more computations. The LMS/RLS filters are used in the sensor fusion by computing the angle from the accelerometer and the gyroscope separately.

The gyroscope data is run through the filter with the input error defined as

$$e_k = \theta_{k,acc} - \theta_{k,RLS}.$$

The accelerometer signal is seen as the desired signal and the filter output is subtracted from this. The filter output will depend on the parameters chosen in the LMS/RLS. In the result the RLS filter shows the best result when the subject is not moving too fast. The Kalman filter did not converge to a proper solution as it presents an increasing bias. The LMS converged but was slower than RLS.

Chapter 2

Inertial Sensors

2.1 Introduction

The base of this thesis is the sensors. Making the right choice of sensors is essential to the algorithms' performance and also to the overall performance of the system. The three types of sensors considered is the accelerometer, gyroscope, and magnetometer. In this project the goal is to measure states of doors and windows. By states one means properties such as the velocity of a door being opened or the angle of which it is open. By putting sensors in the handle of a door or a window it might be possible to accurately track properties such as these and use them for security applications.

The accelerometer measures linear acceleration, the gyroscope measures rotational speed about an axis, and the magnetometer measures its surrounding magnetic fields. These types of sensors measure the heading of the system and since these are proprioceptive properties they can be used to estimate the position when acceleration and velocity information is integrated. When the sensors are used to estimate the position, it is called dead reckoning. An Inertial Measurement Unit (IMU) is an electronic device that contains an accelerometer and a gyroscope, it can also contain a magnetometer to measure the magnetic field surrounding the IMU. In recent years, Microelectromechanical systems (MEMS) have become popular for inertial measurements. The advantage of this type of IMU is the low cost, small size, accessibility, and low power consumption. Because of this they are becoming increasingly popular in user applications such as phones and gaming hand controls [2]. The magnetometer has been used for a long time and has the same function as a compass. Commonly, a Hall effect sensor measures the direction of a magnetic field and describes the direction of the magnetic field through the electric potential in a semiconductor [13].

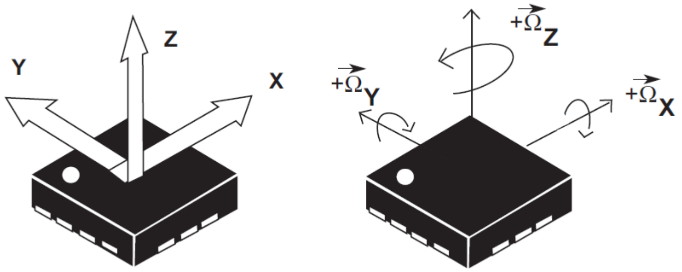


Figure 2.1: The orthogonal axes of an IMU. Taken from the LSM6DS3 datasheet [21].

A problem with the magnetometer is that it is prone to error if the sensor is placed close to other electronics that emit magnetic fields so that instead of measuring just the earth's magnetic field, it also measures other magnetic fields that affects the readings and making the measured direction less accurate. The magnetometer is also affected by drift which will cause it to show the wrong direction after being used for an extended period of time. Other drawbacks of magnetometers are their sensitivity to vibrations and limited bandwidth. Also, if the sensor is placed inside the handle of the door, depending on the material of the handle, it might not have a reading at all. For these reasons, the magnetometer is not further investigated in this project, instead the accelerometer and gyroscope is used. However, these sensor have some inherent issues that are dealt with in the next sections.

2.2 Accelerometer errors and calibration

Accurate tracking of the angle of the door and window is one of the most important goal and of main interest. The ability to track this is directly connected to the quality of the sensor. Larger errors in the sensor readings will lead to less accurate tracking. An modern IMU based on MEMS sensors commonly consists of a tri-axial accelerometer and a tri-axial gyroscope. The axes are here defined as x, y, and z and are orthogonal to each other. The sensors can be seen as having 3 independent sensors, one for each axis. However, the error sources of the sensor for one axis can be divided into two types, the first type being independent of the other axes, and the second type being dependent of the other axes. The MEMS sensors have several error sources that one has to consider when designing systems. For the accelerometer, these are scale-factor, bias or offset, and cross-coupling. There are a lot of internal parameters of the sensors that affects the reading. The ones listed above are the most important as they influence the output the most. Other parameters include properties such as temperature dependency, manufacturing imperfections, material imperfections, and orientation misalignment [2]. Their exact influence depends on the sensor but they are in general very small, often sensors are calibrated around the temperature 25°C. A common model for the accelerometer is [2, 5]

$$y_{acc}(t) = (1 + \delta_{sc})a_r(t) + \delta_o + e(t) \tag{2.1}$$

where y_{acc} is the measured acceleration, a_r is the real sensed acceleration, δ_{sc} is the constant scale-factor, δ_o is the offset, and $e(t)$ is noise. This model does not consider influence of gravity or any cross-coupling effects except for in the noise $e(t)$. This model also does not consider any non-linearity in the scale-factor. Because the scale-factor non-linearity mainly is due to temperature changes and has the size of 0.1-0.4% it is reasonable to ignore this property and instead use a constant scale-factor. Equation (2.1) also shows why it is necessary to calibrate the sensor. The parameters cause the sensor to output values that are not close to the real acceleration so by calibrating the sensors the performance is increased. The calibration of the sensors is done to attenuate internal factors so the sensor only need to be calibrated once.

To estimate the parameters of the model in Equation (2.1) several methods are possible. Previous knowledge about the sensors makes it possible to use some assumptions about them. Thus the task of calibrating the sensors will be seen as a grey-box problem. Firstly, some prior knowledge of the noise is required. The first assumption on the noise is that it is purely additive, this is a common assumption that is sufficient for most applications that only require rate grade performance of the sensors [2, 9, 16]. Next, the probability distribution of the noise influences the approach of determining the parameters. The noise of inertial sensors can typically be assumed to be zero mean Gaussian [2]. For a scalar case of a Gaussian Probability Density

Function (PDF) and constant variance σ^2 the distribution is given by

$$f(y, \sigma^2, \mu) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}. \tag{2.2}$$

The parameters to be estimated in the sensor model are δ_{sc}, δ_o, and σ^2. For simplicity, define the vector θ as

$$\theta = \begin{pmatrix} \delta_{sc} \\ \delta_o \\ \sigma^2 \end{pmatrix}. \tag{2.3}$$

The sensor calibration problem is formulated as an Maximum Likelihood (ML) problem. The input data are measurements taken from the sensor $y_{1:N} = y_1, ..., y_N$ and the parameters are calculated by solving the problem of maximizing the likelihood function

$$\hat{\theta}^{ML} = \arg\max_{\theta} p_\theta(y_{1:N}) \tag{2.4}$$

The problem has the following equivalent formulaten

$$\hat{\theta}^{ML} = \arg\min_{\theta} - \log p_\theta(y_{1:N}) \tag{2.5}$$

The problem of estimating the offset δ_{acc} in Equation (2.1) with the same assumption for the noise PDF is considered first. For each sample, it is assumed that the mean $\mu_{y_i} = (1 + \delta_{sc})a_r + \delta_o$. The likelihood then becomes

$$p(y_{1:N}, \delta_{sc}, \delta_o, \sigma_y^2) = \prod_{i=1}^{N} f(y_i, \mu_i, \sigma^2) = \frac{1}{\sqrt{2\sigma_y^2\pi}^N} e^{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(y_i - \mu_{y_i})^2} \tag{2.6}$$

Now the problem is to minimize this expression to find the optimal parameter of θ. This is done by taking the logarithm, differentiating and solving for the differentiated variable. However, this is under the assumption that the real acceleration. In the case of the accelerometer, this can be achieved by taking the calibration data when the IMU is stationary so that $a_r = 0$ and when the IMU is stationary but angled so that a_r is affected by gravity. The offset of the accelerometer is obtained when the IMU is stationary and not affected by gravity so that $a_r = 0$. Then one obtains

$$\delta_o = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{2.7}$$

and

$$\sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \delta_o)^2 \tag{2.8}$$

So far the meaning of σ_y^2 has not been explained. However, since all the parameters in the model except for the noise are assumed to be deterministic the variance of the gyroscope can be obtained obtains $\sigma_e^2 = \sigma_y^2$.

To find the scale-factor, the assumption $a_r = 0$ is removed. Differentiating the log-likelihood function with regards to the scale-factor results in

$$\delta_{sc} = \frac{1}{a_r}\left(\frac{1}{N}\sum_{i=1}^{N} y_i - \delta_o\right) - 1 \tag{2.9}$$

It is worth to note that in expression (2.9), the summation and δ_o are taken from different sets of data and the real acceleration a_r is given by $a_r = g$ where g is the gravitational acceleration constant.

From the equations above, it is possible to derive a systematic way of calibrating the accelerometer. Since the sensor has three axes, three separate calibrations are done but the same method can be used for all. Firstly, because the expression for the variance and the scale-factor contains the offset, the offset has to be calibrated first. This is done by having the IMU stationary and angled in a way so that it is unaffected by gravity, this can be done by keeping it on a table. A sufficient amount of sampling points can be gathered if the data is taken during 5 to 10 seconds at a sampling rate of 100 Hz [19]. The same data can be used to calculate the variance. Secondly, the IMU is turned so that the the axis under consideration is parallel with the normal of the table. A second measurement session is done and the scale-factor can be calculated from the new data. The precision of this estimate depends on how well the gravitational acceleration constant is known.

The routine of calibrating the sensors is necessary to do for each sensor since every sensor has unique internal properties so the parameters will vary.

Therefore, it is necessary to have a clear structure for how the calibration can be done. This is summarized first for the accelerometer and then for the gyroscope.

For a 3-axis accelerometer with the ODR of 52 Hz it is possible to measure the sensor parameters in Equation (2.3) if the axes of the IMU are known and possible to align with a flat surface. The calibration routine can then be summarized as

1. **Stationary measurement**
   - (a) Keep the accelerometer stationary and measure the two axes unaffected by gravity for atleast 10 seconds.
   - (b) Turn the accelerometer 90 degrees so that the third axis is parallel with the normal of the table and thus unaffected by gravity. Measure the output for the third axis for atleast 10 seconds.
   - (c) When the data is gathered. Compute the accelerometer offset in Equation (2.7) and the noise variance in Equation (2.8) for each axis separately.
2. **Gravity measurement**
   - (a) For each axis, align the axis so that it is parallel with the table's normal and measure the accelerometer output. Repeat for the other axes.
   - (b) For each separate axis, compute the scale-factor from Equation (2.9).

2.3 Gyroscope errors and error calibration

The MEMS sensors have several error sources that one has to consider when designing systems. For the accelerometer, these are scale-factor, bias or offset, and cross-coupling. The gyroscope has similar properties as the accelerometer but is prone to drift as well. A few of the parameters are scale-factor, bias or offset, drift, and cross-coupling which are the same as for the accelerometer. Drift will cause the gyroscope error to grow unbounded over time. There are a lot of internal parameters of the sensors that affects the reading. The ones listed above are the most important as they influence the output the most. Other parameters include properties such as temperature dependency, manufacturing imperfections, material imperfections, and orientation misalignment [2]. The parameters cause the gyroscope to output values that are not close to the real angular speeds so by calibrating the sensors the performance is increased. The calibration of the sensors is done to attenuate internal factors so the sensor only need to be calibrated once. Calibrating the gyroscope is more difficult than calibrating the accelerometer as it has no simple way of obtaining a constant angular rate. In the accelerometer, the acceleration due to earth is always present. For the gyroscope, no such inherent property is available. This makes the calibration of the scale-factor more problematic. However, the parameters of the gyroscope sensor model can still be decided in the same way as for the accelerometer. The model used for the gyroscope is

$$y_{gyr} = (1 + \delta_{sc})\dot{\psi}(t) + \delta_o + e(t). \tag{2.10}$$

In this model, the same notation is used for the scale-factor δ_{sc}, the offset δ_o, and the noise $e(t)$. The real angular rate is denoted $\dot{\psi}$ where ψ is the angle which is arbitrarily defined. An estimate of the offset is still possible when $\dot{\psi} = 0$. By taking the mean of the gyroscope readings one obtains $\mu_{y_i} = (1 + \delta_{sc})\dot{\psi}_i + \delta_o$. Under the assumption that the noise is zero mean white noise the log likelihood function is

$$L(y, \delta_o, \delta_{sc}, \sigma_y^2) = -\frac{N}{2}\log(2\pi\sigma_y^2) - \frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(y_i - \mu_{y_i})^2 \tag{2.11}$$

In the same way as for the accelerometer, the offset and the variance can be calculated by having the gyroscope stationary and differentiating the log-likelihood function.

$$\delta_o = \frac{1}{N}\sum_{n=1}^{N} y_i \tag{2.12}$$

and

$$\sigma_y^2 = \frac{1}{N}\sum_{n=1}^{N}(y_i - \delta_o)^2. \tag{2.13}$$

To find the scale-factor, some assumption on the angular velocity is needed. One solution is to use a corresponding known angular velocity for some time interval so that $\dot{\psi} = \dot{\psi}_i$. Then the expression for δ_{sc} becomes

$$\delta_{sc} = \frac{1}{\dot{\psi}} \left(\frac{1}{N} \sum_{n=1}^{N} y_i - \delta_o \right) - 1 \tag{2.14}$$

To find the scale-factor it is necessary to have a known angular rate of which the IMU is rotating. One solution to this is to use a single axis turn table can be used to impose a true reference angular rate. The turn table rotates the IMU about its axis. Another solution is to use a spin table where the gyroscope is mounted on top. The table is accelerated to the wanted angular rate and when the speed has stabilized the gyroscope data is measured [25].

The calibration of the gyroscope is more complicated as it is not possible to calibrate the scale-factor without using equipment to rotate the IMU. However, for many applications it is sufficient to assume that $\delta_{sc} = 0$. In the gyroscope calibration routine the scale-factor calibration is still included but can be left out. The calibration routine for a gyroscope with ODR of 52 Hz is summarized as

1. **Stationary measurement**

   (a) Keep the gyroscope stationary and measure the angular rate of all three axes for atleast 10 seconds.

   (b) When the data is gathered. Compute the gyroscope offset in Equation (2.12) and the noise variance in Equation (2.13) for each axis separately.

2. **Scale-factor measurement**

   (a) Mount the IMU on a turn-table and align the gyroscope with the middle of the turn-table. Accelerate to a known angular rate and when it has stabilized, measure the gyroscope output for atleast 10 seconds. Repeat for the other axes.

   (b) For each separate axis, compute the scale-factor from Equation (2.14).

2.4 Sensor comparison and choice

Today, the availability of low cost MEMS sensors is large and it is still growing. There are several parameters of the sensors that are important when making the choice of what sensor to use. Thus an investigation and comparison of sensors is called for. Today it is common to have a module with integrated gyroscope and accelerometer, this saves space and is easier to fit into a module. Another important scope is the current consumption, many IMUs today have a standby mode where they consume considerable less power than when they are being used. This makes it possible to compare both stand-by power consumption and operational power consumption. Furthermore, there are multiple parameters regarding the performance of the IMU. These performance parameters may deal with Signal to Noise Ratio (SNR), the range in which the sensor can measure, the Output Data Rate (ODR), non-linearity, and sensitivity to changes.

In this project, sensors from three different suppliers are compared. One goal of the project is to implement the electronics in the door- and window handle which can be made out of magnetic materials and have other electronic objects in it. Because of the problems with the magnetometer, it is not considered for use. The sensors are listed in Appendix A together with their features.

Six different IMUs are listed. All of the sensors compared have 3 axes for each sensor. The first sensor, ADXL362 [17], is an accelerometer only, this sensor is a popular sensor due to its low price, relatively good features, and low current consumption. Comparing its current consumption to the other IMUs, the other sensors have considerably higher current consumption due to having multiple sensors and better values of their properties, among which scale-factor is one of them [17,18,20–23]. For all the sensors, the scale-factor is listed for different ranges. The range of the sensor is an user selected parameter and decides in which range it can sense. This property is given in terms of earth's gravity and all of the sensors in the comparison can atleast measure accelerations in the range $\pm 8g$. For this project, this range is acceptable as the only time the accelerometer will measure values close to the limit is when the door is hitting the door frame. Even in this case, from practical experience, accelerations are way below this range. Summarizing all of this, the LSM6DS3 has the best properties of the sensors. It has the lowest scale-factor and the lowest current consumption. The LSM6DS3 also has a large range of the ODR. It can be chosen to be 13Hz up to 1.6kHz. When implementing algorithms using this sensor it can be of interest to see what effects the sampling time can have on the system. A lower ODR means lower current consumption but might affect the algorithms negatively due to the lower sampling time.

Chapter 3

System Description and Filtering Methods

3.1 Door Movement Model

One goal of the project is to develop algorithms to accurately measure doors and windows positions. This means that the angle of which the door or window is open is to be monitored. In this system, the IMU is mounted on the door so that two axes are aligned with the door's surface and the third axis is perpendicular to the surface as shown in Figure 3.1. There are many possible ways to monitor this. One way is to use a quaternion based approach. In this approach, the quaternion vector consists of 4 elements. To use the quaternion in a state space representation it can be related to the angular velocity of the system. Define

$$q = \begin{pmatrix} q_0 & q_1 & q_2 & q_3 \end{pmatrix}^T \tag{3.1}$$

as the quaternion vector. The elements q_1, q_2, q_3 are the vector part of the quaternion, and can be thought of as a vector about which rotation should be performed. The element q_0 is the scalar-part of the quaternion and specifies the amount of rotation that should be performed about the vector part. It can be shown that, when q is constricted to the unit sphere so that $q^T q = 1$, then

$$\frac{d}{dt}q(t) = -\frac{1}{2}S(\omega)q(t), \tag{3.2a}$$

$$\frac{d}{dt}q(t) = -\frac{1}{2}\bar{S}(q)q(t) \tag{3.2b}$$

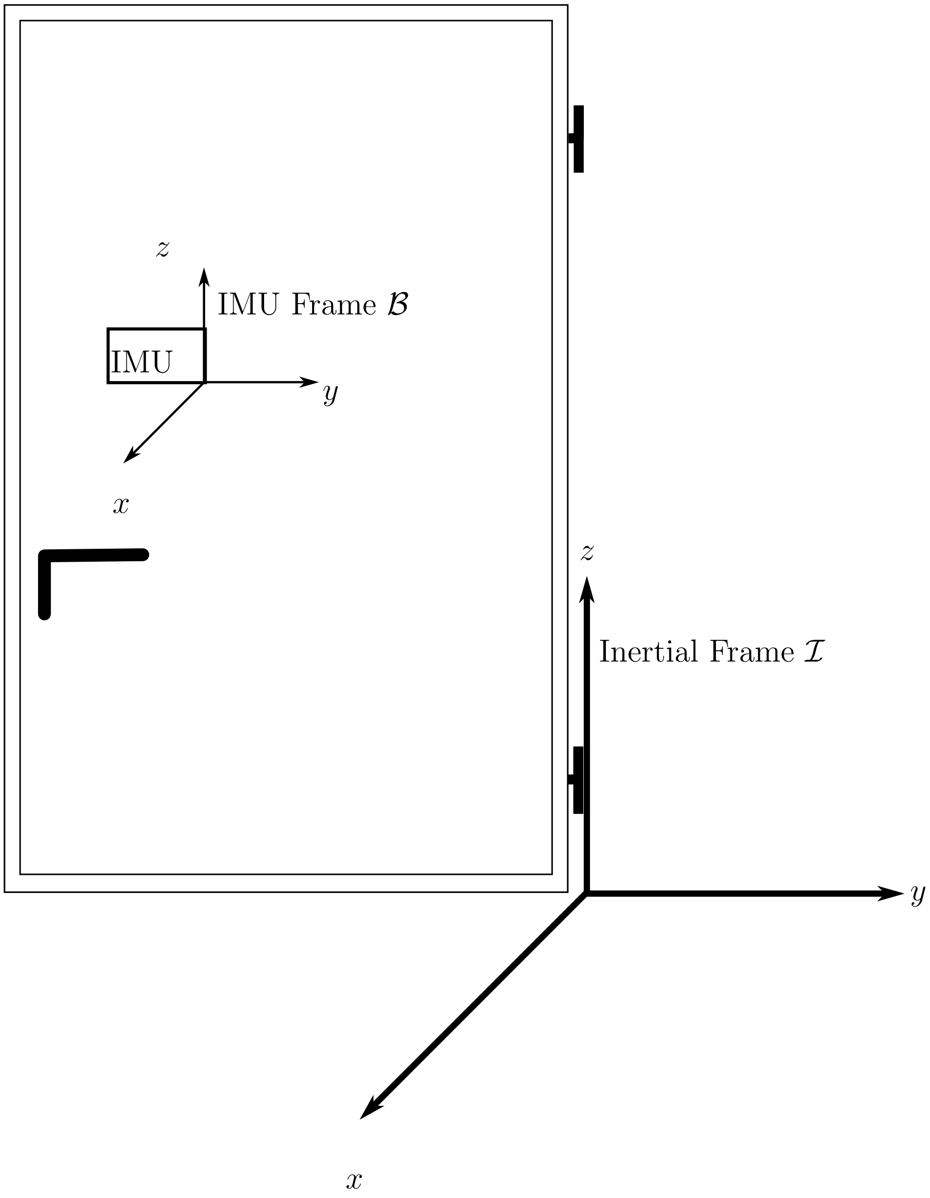


Figure 3.1: Illustration of the system. The IMU is mounted on the door.

where $\bar{S}(q)$ and $S(w)$ are two predefined matrices. This approach is the preferred one if the movement is happening in three dimensions. If that is the case, using quaternions can decrease the computational needs for the system. Quaternions are a popular approach for systems involving IMU's and GPS [7, 24] and while they are good for decreasing computational needs it is still possible to use a system in 2D for monitoring a door or a window as the movement primarily happens in 2 dimensions. The door only moves in the xy-plane of Figure 3.1 and the only movement that can happen in the z-direction is due to friction and vibrations which are very small and not of interest.

However, with the assumption that the door movement is happening in only the xy-plane of Figure 3.2, a simpler model of the door movement can be derived. Start from a position $p(t)$ as shown in Figure 3.2. It is known that the movement is circular as the radius of the door is constant and the IMU's position does not change. The radius r in this model is defined as the horizontal length from the IMU to the z-axis of the inertial frame $\mathcal{I}$ in Figure 3.1. There is a choice to be made on whether to model the movement in the inertial frame or in the body frame of the IMU. The advantage of using the body frame is the avoidance of having to do a transformation from the inertial frame, thus reducing the computational cost, and simplifying the model. The angle of the door starting from when the door is closed is the most interesting property to track. Secondly, the angular speed and acceleration is of interest as these can be used for analysis of different patterns when the door is opened and closed.

A simpler option for modelling coordinated turns in 2D is to construct a model around Equations (3.3) instead based on [4]. This model requires a smaller state space model than the previous one. The motion is described by the motion in figure 3.2.

$$\dot{\psi} = \frac{v_x}{R}, \tag{3.3a}$$

$$a_y = \frac{v_x^2}{R} = v_x\dot{\psi}, \tag{3.3b}$$

$$a_x = -v_y\frac{v_x}{R} = -v_y v_x R^{-1} = -v_y\dot{\psi} \tag{3.3c}$$

The equations state a coordinated turn in 2D in the body frame. The model requires 5 states for the velocity, acceleration, angle, and velocity in polar

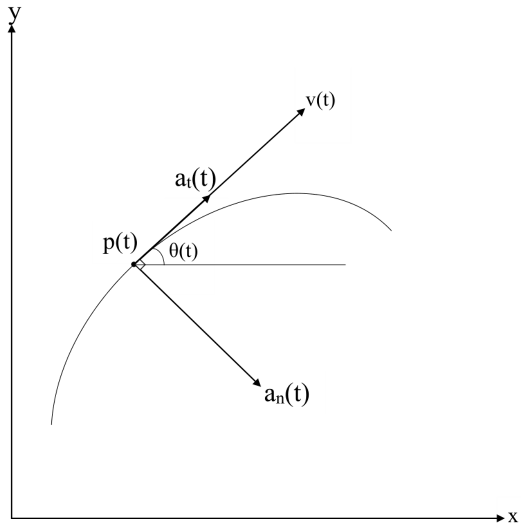


Figure 3.2: The geometry of a coordinated turn in 2D. The appearance of the curve depends on how the radius varies.

coordinates in the world frame $\mathcal{I}$ of figure 3.1. To discretize this model the exact sampling formula [4] in Equation (3.4) can be used.

$$x(t+T) = x(t) + \int_{t}^{t+T} a(x(\tau))d\tau \tag{3.4}$$

In Equation (3.4), T is the sampling time and $a(x(t))$ is the continuous time non-linear system. For the system above, $a(x(t))$ is

$$a(x(t)) = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{v} \\ \dot{\psi} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} v\ cos(\psi) \\ v\ sin(\psi) \\ 0 \\ \omega \\ 0 \end{pmatrix} \tag{3.5}$$

where the state x is

$$x = \begin{pmatrix} X \\ Y \\ v \\ \psi \\ \omega \end{pmatrix} \tag{3.6}$$

To discretize the model the Taylor expansion of $a(x(t))$ can be used. The discrete state space matrix can then be acquired as $F = e^{AT}$ where A is the first derivative part of the Taylor expansion of $a(x(t))$. However, in this case it is possible to use the sampling formula to acquire an analytical solution to the discretized system. The sampling formula is solved by using some symbolic computation programs such as Matlab or Maple. The discretized solution to Equation (3.4) using Equation (3.1) is

$$x(t+T) = f(x(t)) = \begin{pmatrix} X(t) + \frac{2v(t)}{\omega(t)} sin(\frac{\omega(t)T}{2} cos(h(t)) + \frac{\omega(t)T}{2}) \\ Y(t) - \frac{2v(t)}{\omega(t)} sin(\frac{\omega(t)T}{2} sin(h(t)) + \frac{\omega(t)T}{2}) \\ v(t) \\ h(t) + \omega(t)T \\ w(t) \end{pmatrix} \tag{3.7}$$

The measurement model is straight forward to calculate. Since the available measurements from the IMU are a_x, a_y, and ω. They can be used to derive a simple measurement model. The output equations are then

$$y = h(x(t)) = \begin{pmatrix} a_x \\ a_y \\ \dot{\psi} \end{pmatrix} \tag{3.8}$$

The only inputs to the system is the physical force by people opening the door. That force is modelled as an external force. The door will be affected by external forces whenever someone is entering or leaving through the door. The exact model and properties of this is unknown but can be included in the filter. By adding an uncertainty after every prediction step, i.e. adding some uncertainty on state x_{t+1} after it has been predicted from x_t. This can be seen as the state x_{t+1} could be inside some area but the exact location is uncertain.

So far no attention has been paid to the noise models. It is fair to assume that the noise variable $v(t)$ is stochastic and constant between each sampling interval. If it is modelled as white noise and has the covariance matrix Q in continuous time then the total influence of the noise on the process during one sample interval is $Q_d = TQ$ [4]. The noise in the measurement Equations are also modelled as zero mean white noise with their variances estimated from the sensor calibrations. The model is then summarized as

$$x(k+T) = f(x(k)) + v(k) \tag{3.9a}$$

$$y(k) = h(x(k)) + e(k) \tag{3.9b}$$

3.2 Window movement Model

Measuring the angle of which a window is open is similar to that of measuring the door's angle. The movement of a window can be described by the same equations as the door. The movement is circular and the IMU is placed on the window's body frame so that two axes are aligned with the window's surface. However, the movement of the window takes place relative to earth's gravitation. This means that the sensor readings measure the gravitation vector **g** as well as the acceleration of the window. Thus the measurement model in (3.8) cannot be used without modification.

By using the same model as for the door with the state vector

$$x = \begin{pmatrix} X \\ Y \\ v \\ \psi \\ \omega \end{pmatrix} \tag{3.10}$$

and the states

$$x(t+T) = \begin{pmatrix} X(t) + \frac{2v(t)}{\omega(t)} sin(\frac{\omega(t)T}{2} cos(h(t)) + \frac{\omega(t)T}{2}) \\ Y(t) - \frac{2v(t)}{\omega(t)} sin(\frac{\omega(t)T}{2} sin(h(t)) + \frac{\omega(t)T}{2}) \\ v(t) \\ h(t) + \omega(t)T \\ w(t) \end{pmatrix} \tag{3.11}$$

the measurement model can be modified to measure the accelerations due to external forces such as people opening and closing the window. In the state equations x is defined as the downwards direction of the window and y is defined as the direction of the window's normal vector. When the window is closed the gravitation vector will only affect x. In this position the gravity vector is entirely in the same direction. When the window is being opened, the angle ψ changes and thus the body frame coordinate system changes as well. The gravity vector is still pointing towards the ground but the acceleration of the system has been rotated by an angle ψ. Thus, using vector algebra, the acceleration due to gravity is obtained as

$$\begin{aligned} g_x =& g\ cos(\psi), && \text{(3.12a)} \\ g_y =& -\ g\ sin(\psi), && \text{(3.12b)} \end{aligned}$$

where g is the gravitational constant. If this is seen as entering the measurement model the state equations of the system will not change. The measurement model become

$$y = \begin{pmatrix} a_x \\ a_y \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} a_z - g\ cos(\psi) \\ a_y + g\ sin(\psi) \\ \omega \end{pmatrix} \tag{3.13}$$

3.3 The Kalman Filter

So far a state space model for circular movement of windows and doors has been derived. The goal is to find the angle of which the door and window is opened but the angle is not directly observable. The non-linear filtering problem aims to find this hidden state when partial information can be obtained through observations. In this case the hidden state is the angle and the observable states are accelerations in two directions and the angular velocity about the axis of rotation. When external sensors such as the IMU is used this is considered a tracking problem.

A classical approach to this problem is the Kalman filter (KF). The KF computes a posterior distribution for linear Gaussian systems by updating a finite dimensional statistics recursively [26]. The KF consists of a measurement update and a time update.

The Kalman filter finds the optimal linear filter, given a linear state space model

$$
\begin{aligned}
x_{k+1} &= F_k x_k + G_{u,k} u_k + G_{v,k} v_k, \quad Cov(v_k) = Q_k, && (3.14a)\\
y_k &= H_k x_k + D_k u_k + e_k, \quad Cov(e_k) = R_k, && (3.14b)\\
E(x_0) &= \hat{x}_{1|0}, && (3.14c)\\
Cov(x_0) &= P_{1|0}. && (3.14d)
\end{aligned}
$$

The design choices for the KF is the state and of the dynamic and measurement models. The notation $k|m$ is read as "at the moment k given measurements up to moment m". The variables e_k and v_k are modelled as white noise with zero mean. In this project, only tracking of doors and windows is considered and in this case, no input to the system in Equation (3.14) will be used. The standard form of the Kalman filter's update equations consist of a measurement update and a time update [26],

Measurement update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} (y_k - H_k \hat{x}_{k|k-1} - D_k u_k), \quad (3.15a)$$

$$P_{k|k} = P_{k|k-1} + P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} H_k P_{k|k-1} \quad (3.15b)$$

Time update:

$$
\begin{aligned}
\hat{x}_{k|k} &= F_k \hat{x}_{k|k} + G_{u,k} u_k, && (3.16a)\\
P_{k+1|k} &= F_k P_{k|k} F_k^T + G_{v,k} Q_k G_{v,k}^T && (3.16b)
\end{aligned}
$$

The extended Kalman filter uses a non-linear state-space model,

$$
\begin{aligned}
x_{k+1} &= f_k(x_t, u_k, v_k), && (3.17a)\\
y_k &= h_k(x_k) + e_k && (3.17b)
\end{aligned}
$$

Once again v_k and e_k are white noise with zero mean. Similarly to the Kalman filter, the EKF estimates the new state by performing a time update and a measurement update. Also, the design choices are the choice of state and of the dynamic and the measurement models. This filter has proven effective when the model is close to linear but there are no guarantees [9]. The EKF linearises the system at each time step according to

$$A_k = \frac{\partial f_k(x_t, u_k, v_k)}{\partial x_k}, \ G_k = \frac{\partial f_k(x_t, u_k, v_k)}{\partial v_k} \quad (3.18)$$

where G_k is used to update the state covariance.

The Extended Kalman Filter (EKF) is the non-linear version of the KF. The EKF linearizes around the current state estimate. For a non-linear system

$$\begin{aligned} x_{k+1} =& f(x_k, u_k, v_k) \qquad (3.19a) \\ y_k =& h(x_k, u_k, e_k) \qquad (3.19b) \end{aligned}$$

is linearized around the Taylor expansion of $f(x_k, u_k, v_k)$ if the standard EKF is used, here written compactly as $f(x)$ and $h(x)$. It is assumed that the functions are differentiable. The Taylor expansion of $f(x)$ around a point $\hat{x}$ is given by

$$f(x) = f(\hat{x}) + f'(\hat{x})(x - \hat{x}) + \frac{1}{2}(x - \hat{x})^T f''(\xi)(x - \hat{x}). \tag{3.20}$$

Here $f'(x)$ denotes the Jacobian and $f''(x)$ denotes the Hessian of $f(x)$ and ξ is a point in the neighbourhood of $\hat{x}$. It is also possible to use the linearized Kalman filter to linearize around a reference trajectory instead. Other variants of the EKF is the error state Kalman filter that estimates the state error $\tilde{x}_k = x_k - \hat{x}_k$ with respect to some approximate or reference trajectory.

It is initialized by $\hat{\mathtt{x}}_{1|0}$ and $P_{1|0}$. The algorithm consists of a prediction stage and an update stage. Depending on if f''_x and h''_x are assumed to be zero, two different versions of the EKf can be stated [4]. The first version is given by the following recursion:

$$\begin{aligned} S_k &= h'_x(\hat{x}_{k|k-1})P_{k|k-1}(\hat{x}_{k|k-1})^T + R_k \qquad (3.21a) \\ K_k &= P_{k|k-1}(h'_x(\hat{x}_{k|k-1}))^T S_k^{-1} \qquad (3.21b) \\ \epsilon_k &= y_k - h(\hat{x}_{k|k-1}) \qquad (3.21c) \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \epsilon_k \qquad (3.21d) \\ P_{k|k} &= P_{k|k-1} - P_{k|k-1}(h'_x(\hat{x}_{k|k-1}))^T S_k^{-1} h'_x(\hat{x}_{k|k-1})P_{k|k-1} \qquad (3.21e) \\ \hat{x}_{k+1|k} &= f(\hat{x}_{k|k}) \qquad (3.21f) \\ P_{k+1|k} &= f'_x(\hat{x}_{k|k})P_{k|k}(f'_x(\hat{x}_{k|k}))^T + Q_k \qquad (3.21g) \end{aligned}$$

Another controller is used to estimate the drift of the gyroscope. By dividing the system into two parts reduces the computational cost [14]. The cause of the drift of the MEMS gyroscopes is mainly the temperature dependence [3]. To combat the drift a novel method for drift compensation is designed. This method uses the knowledge that when the window is stationary, the accelerometer will only measure its internal noise and no external acceleration caused by the door. This also means that no rotation is taking place and the only contribution to the gyroscope measurement is caused by its drift and noise parameters. Starting with the same model for the gyroscope as earlier

$$\omega_k = \omega_{r,k} + b_k + e_k \tag{3.22}$$

the goal is to use the accelerometer to compensate the the drift so that the drift b_k disappears. This can be done by using a simple controller that uses the saturation function to limit the system to only use the stationary measurements to reduce the bias. This means that when the door or window is moving the drift compensation is not running. This heuristic controller can be described by

$$\omega_{k+1} = \omega_k - f(a_k)\ \omega_k \tag{3.23}$$

where $f(a_k)$ is the saturation function with a threshold a_{th} given by

$$f(a_k) = \frac{1}{2}(sat(a_k + a_{th} - 1) - sat(a_k - (a_{th} + 1)) \tag{3.24}$$

and

$$sat(x) = \begin{cases} 1 & x \geq 1 \\ x & |x| < 1 \\ -1 & x \leq -1 \end{cases} \tag{3.25}$$

This will make sure that the drift is attenuated when the door or window is not moving. It is essentially the same as unplugging the gyroscope but without the consequences of missing out on the first samples when the door or window starts moving again. The system relies on a good choice of the threshold value for the accelerometer. However, a value of this threshold can be derived from the fact that the accelerometer noise's mean and variation is known from the calibration routine. The SNR of the accelerometer is also high when it is in use making the value of a_{th} have less influence on the functionality of the system.

3.4 System Design

The complete design is shown in Figure 3.3. The system consists of two stages, the first stage serves as a gyroscope drift compensator by utilizing the fact that the door or window can be stationary and does not move for longer periods of time. The system is intended to be used on a simple micro controller with limited computational power. Because of this it is attractive to use simpler models and a state space model with as few states as possible as this minimizes the size of the matrices and thus making the system faster. In this project, a thorough investigation into the processor speed and power consumption was not done. However, the system is able to run on the micro processor. It is able to complete the cycle shown in Figure 3.3 without falling behind.

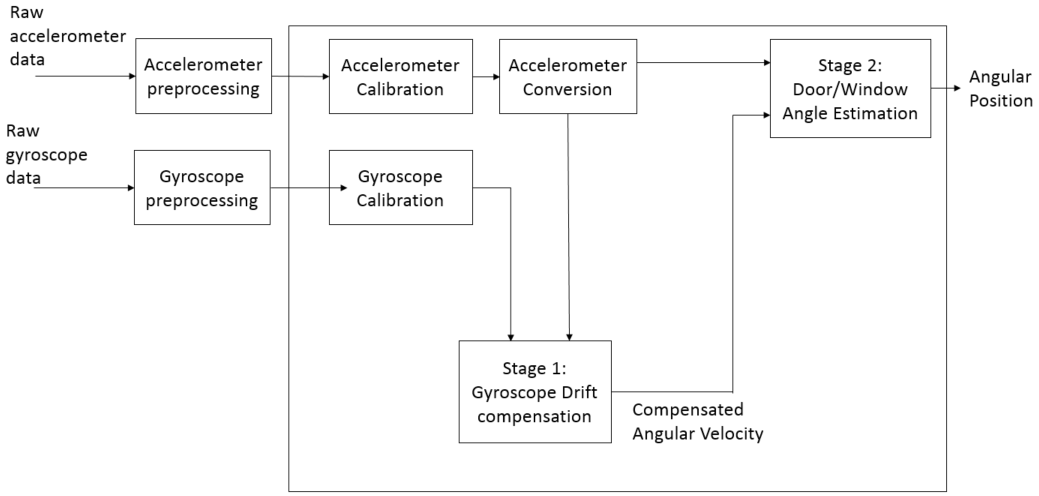

Raw accelerometer data
Accelerometer preprocessing
Accelerometer Calibration
Accelerometer Conversion
Stage 2: Door/Window Angle Estimation
Angular Position
Raw gyroscope data
Gyroscope preprocessing
Gyroscope Calibration
Stage 1: Gyroscope Drift compensation
Compensated Angular Velocity

Figure 3.3: Overview of the proposed system.

The IMU module sends the data in different formats. The accelerometer data is given in [g] so that the axis affected by gravity outputs values close to 1. This means that except for the calibration routine the accelerometer data needs to be converted to m/s^2 through a simple conversion. The gyroscope outputs data with the unit DPS meaning degree per seconds which can either be used directly or changed to radians per second.

The system uses the model in Section 3.1 and 3.2 together with the sensor data as a starting point. It also uses the equations in Section 2. In Stage 1 the Kalman filter estimates the gyroscope drift and in the second stage, the Kalman filter estimates the door- or window angle by using the drift compensation. The attenuation of the angular drift is also important for the system to function, as is the threshold value for the accelerometer. This makes tuning the system important. The parameters that can be tuned are not changing so they were not shown here. Also not included in this system is the algorithm used to detect if the door closes fully is not shown.

Not shown in Figure 3.3 is the door- and window closing classification algorithm. Due to the difficulty of modelling the movement of the window and door when it hits the frame, a separate classification algorithm is used. This subsystem uses the system above to check whether the door is close to its frame (low angular position). If so, the algorithm tries to calculate if the door closed fully or not.

Chapter 4

Event Classification Based on Spectrogram Pattern Matching

4.1 Introduction

4.1.1 Supervised Learning

Due to the limits of the door and window model, the system is aided by a supervised learning algorithm to sense when the door or window hits the frame and either closes entirely or is stuck. The algorithm relies on previous data from the accelerometer to compare the new data to. Because each door and window closing is different, the analysis of the time series are difficult. Instead their spectrums are utilized to improve the classification.

The objective of supervised learning is to take a set of previous observations of patterns and labels and use a functional model to derive relationships between the previous observations and the new observations. The previous observed data and its corresponding labels are used to determine the label of new data with unknown labels. For example, in this case a data base of previous accelerometer data of the door hitting the door frame is used. Each instance of data has a corresponding label of the door either closing properly or getting stuck in the door frame.

This can be summarized as observations $x_i \in \mathbb{R}^q$ are given together with a label $y_i \in \mathbb{R}$ in pairs $(x_i,\ y_i),\ i = 1, 2, ...N$ where the dimension of the observations is q and the dimension of the label is one. A functional model f can be trained for the observations that later is used to map new instances of data. There are many different ways to derive model for this with different complexity. Different algorithms are suited to solve some types of problems better than others, this often depend on what kind of observations are available.

4.1.2 K-Nearest Neighbours

The K-Nearest Neighbours (KNN) algorithm aims to classify a target pattern $\boldsymbol{x}$ by assigning it to a class based on previous data. It does this by determining a distance from $\boldsymbol{x}$ to the k nearest data points where k is an positive integer. Depending on the labels of these data points, $\boldsymbol{x}$ is assigned to the class with a majority of the k nearest data points. This means that k can be chosen arbitrarily. However, there are limitations on how it can be chosen. K must be chosen so that each pattern can be given a class. This means that if the targets are to be classified into 2 classes, then k must be an odd number.

To find the distance between the target and the previous data, a distance metric must be used. A common distance metric is the Minkowski metric (p-norm) [11]

$$||\boldsymbol{u} - \boldsymbol{v}||^q = \left(\sum_{i=1}^{q} |u_i - v_i|^q \right)^{\frac{1}{q}} \tag{4.1}$$

where q is the dimension of the two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. When $q = 2$ this becomes the euclidean norm. This way of measuring distances works well for non binary data. For binary data the Hamming distance can be used instead. The term *locality* of the KNN is defined by the number of neighbours k. If a very small number is chosen for k, then the KNN algorithm will have a large locality and little neighbourhoods arise in the regions. If, instead k is very large, patterns with labels in the minority are ignored. This means that for small k the KNN classifier overfits and becomes very local while for larger k, KNN generalizes and ignores small clusters of patterns. For example, if training data of observations of 2 different classes are given and a new set of data is to be classified from these. Suppose the two classes contains blue and red points and the training data is separated by their classes, then the goal is to classify new data correctly to either the blue or the red set. Suppose an outlier of the blue points is located amid the red cluster, then if $k = 1$ a data point located very close to this blue point is classified as blue instead of red when it should be classified as red. This is avoided by increasing k [11]. This raises the question of how to choose k to achieve the best classification for the neighbourhood size. Another way to state this question is how the model is to be chosen. One way to answer this is by using cross-validation.

An often employed strategy to decide on a model in supervised learning is cross-validation. For the problem observations of measurement and class $(\boldsymbol{x}_i, y_i),\ i = 1, ..., N$ are given and defined as the training sets. Cross-validation divides the training sets into 3 different sets. One for training, one for validation, and one for testing. First the validation set is used to evaluate different models for the training set. When an optimized model is found, the testing set is used as verification of the model. This requires a large enough set of measurements to be useful.

4.2 Spectrogram Patterns From Accelerometer

Due to the limitations of the sensor fusion system a method for detecting the door state when it hits the door frame is developed. It is of interest to know if the door is properly closed or if the door is stuck in a state where it rests against the door frame but not entirely closed. This can be a security breach that can be communicated to an alarm center. However, while detection of the door hitting the door frame is easy, detecting whether the door is closed or not is not. Furthermore, the gyroscope does not give valuable information in this case due to the small movements of the door. Instead, the accelerometer data can be used to detect changes. This algorithm also serves as a way to correct the angle in the door movement system. Because the system is not ensured to be free from drift the classification algorithm can be used to correct this and thus improve the door tracking system.

There are several problems when trying to detect and classify the way the door closes. The first problem arises when the door is closed from different angles. For this project a door which automatically closes is used. Because of the nature of how the door closes, the angular velocity and the acceleration of the door can vary depending on from what angle the door is closed. The algorithm needs to take this into consideration or be unaffected by it. Another problem is the sampling frequency of the IMU. Because the movement of the door hitting the door frame can take less than a second, it is important to use an adequate sampling frequency. However, a higher sampling frequency requires more power to run and thus it is a trade-off between accuracy and power consumption. Another problem is how well the accelerometer detects the movement. From tests it is clear that the motion is much stronger in the x-direction of the accelerometer in Figure 3.1. In the training sets the acceleration in the x-direction during the movement is up to 5 times as big as the acceleration in the y-direction. The acceleration data of the x-axis also contains more interesting features that can be used for the classification.

Due to the mentioned problems it is difficult to perform any classification directly on the time series of the accelerometer. However, by using features obtained from the spectrogram these problems are avoided and clear features for classification can be extracted. The spectrogram is computed by letting the acceleration signal be $a(n)$ for the samples $n = 1, 2, ..., N$ and calculating the discrete Fourier transform for each sample instant. To do this, the Fast Fourier transform (FFT) is used, then the Fourier transform $A(r)$ of the signal is computed as

$$A(r) = \sum_{n=0}^{N-1} a(n)e^{-j2\pi nr/N} \tag{4.2}$$

where N is the signal length. To generate the spectrogram the spectrum for each time instant are stacked side by side and the power spectral density is calculated in dB by

$$P(r) = 10\log_{10}(A(r)). \tag{4.3}$$

The reason for using the power spectral density is to partition the range of $A(r)$ into boxes [8]. For each sample of the signal the power spectral density is used as its features for classification. Figure 4.2 shows an example of one of the training sets used for classification.

4.3 Classification

The two types of movements desirable to detect are when the door is closed correctly and when the door is stuck. The classes used for classification were decided by looking at the spectrograms and comparing the different

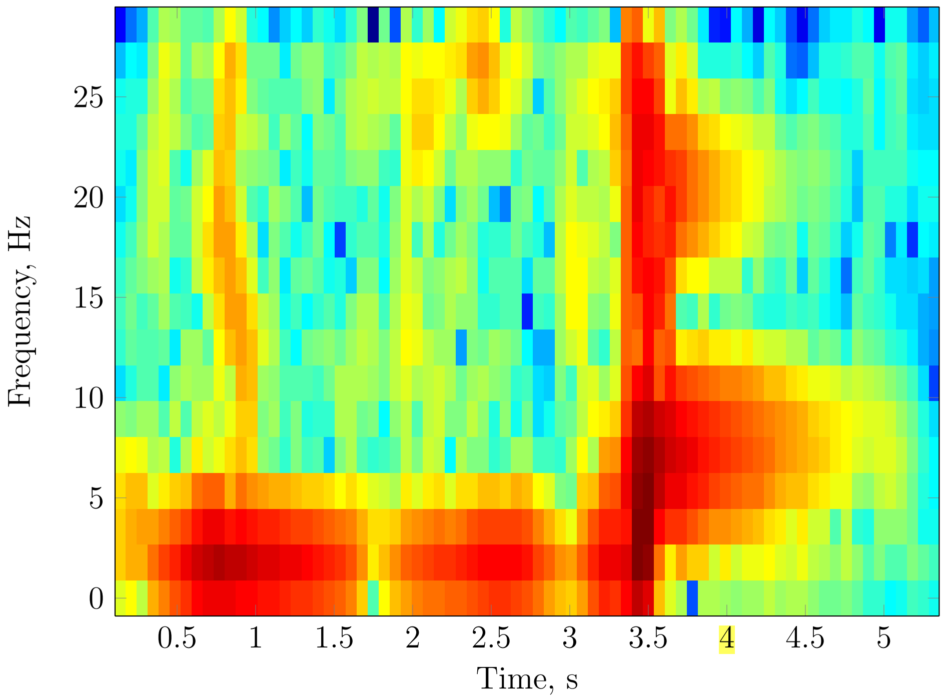


Figure 4.1: The spectrogram of a door closing. In this case the door did not close entirely and an interesting pattern appears around the 4 second mark.

distinguishable frequency content for different time instants. From the spectrograms there were 5 different distinguishable states and distinct differences between the two types of door movements. This is what motivated the use of the spectrogram classification method. The method also solves the problem of the different amount of time it can take for the door to close [8]. Because the time it takes for the door to close and become stationary depends on how the door was opened it hits the door frame with different speed and acceleration and while this changes the frequency components slightly the patterns in the spectrogram remains largely unchanged. It does affect the amount of samples of the spectrogram but these can still be classified into the existing classes.

A potential problem with the KNN classifier is the curse of dimensionality. The data space is defined as the space where the feature vectors are located. Because each feature vector of the spectrogram contains many points of data the computational speed is affected and intuition of the patterns break down.

Since more elements are used for each feature vector the distance metric becomes less reliable. To solve these problems dimension reduction is used. One of the simplest ways to reduce the dimension is to attenuate the feature vectors by only using every second or every third element. This is only reliable if the vector's features are very similar to their neighbours.

The process of classifying the door closings is summarized in Figure 4.3. The outcome of the classification of the spectrogram is an array of data points with the same length as the length of the spectrogram, each containing a class. This is used to compare the known classes from old spectrograms. The comparison finds the best fitting spectrogram with least errors between its classes and the new classified spectrogram's classes. Since the type of door closing is known for the old best fitting spectrogram the new door closing can be classified as the same.

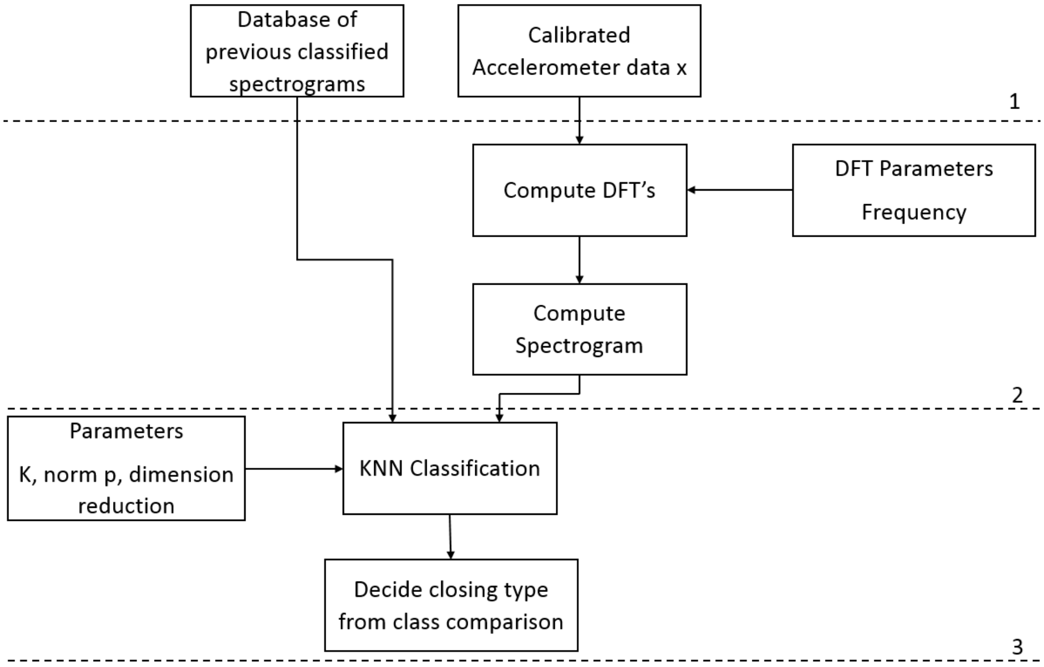

Database of previous classified spectrograms
Calibrated Accelerometer data x
1
Compute DFT's
DFT Parameters
Frequency
Compute Spectrogram
2
Parameters
K, norm p, dimension reduction
KNN Classification
Decide closing type from class comparison
3

Figure 4.2:

Chapter 5

Door and Window system Results

5.1 Setup and Methods

5.1.1 Door Measurement Setup

To test the model and algorithm of measuring the door angle a simple setup with a micro controller board is mounted on the door as seen in Figure 5.1. The micro controller is programmed in C++ and in this setup parameters such as the sampling time can be chosen. For rapid testing the algorithm is developed in Python, the micro controller is connected to a computer through the serial bus and sends the sensor values directly to the computer to allow python to run the code in real time. Because it is intended for the IMU to be mounted in the handle of the door, the sensor board is placed at the same distance from the door frame.

5.1.2 Window Measurement Setup

The same setup as for the door measurements is used for the window. The sensor board is mounted on the window frame so that it has the same length from the axis it is turning about as the window handle. The setup is shown in Figure 5.2. Testing is done the same way as for the door with the sensor board connected to a computer running the algorithms in Python.

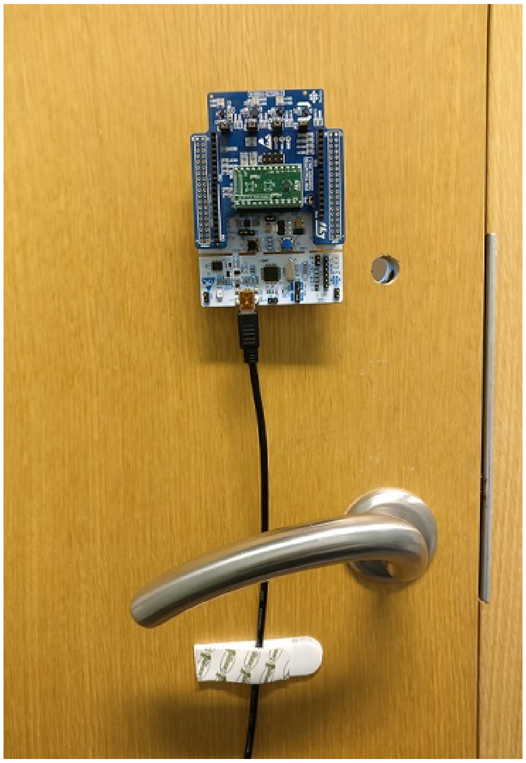


Figure 5.1: The setup for using the sensors on a door. The setup uses a Nucleo F401RE development board and the Nucleo IKS01A1 sensor expansion board to mount the LMS6DS3 IMU on.

5.2 Door Angle Measurement

To give initial values of the sensor offset and bias the calibration routine is run as the micro controller and sensor board is started. An initial state of the system can then be given as a initialization for the recursion since the IMU is stationary and thus all the states are 0. The state measurement covariance matrix is also calculated from the calibration routine and used as initialization for the filter. The covariance matrix is initialized as a identity matrix multiplied by a small constant $\delta = 0.1$.

To test the accuracy of the model several tests were conducted. It is of interest to see how the speed of which the door is opened affects the accuracy of the EKF. It is also interesting to see how the door's angle drift over time.

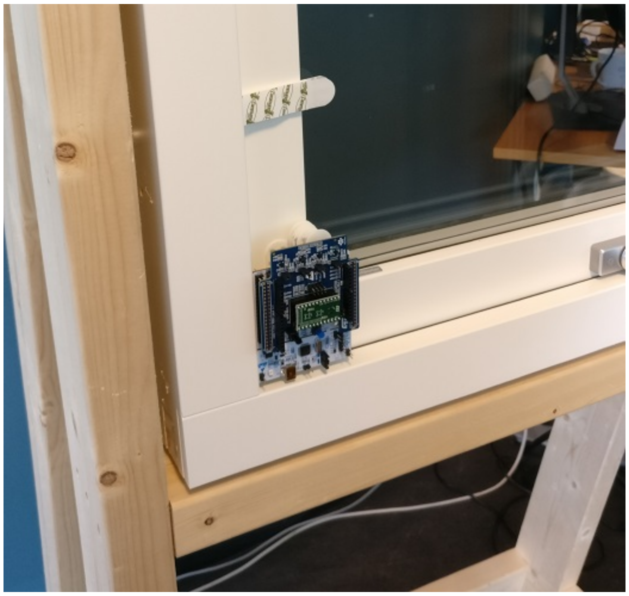


Figure 5.2: The setup for using the sensors on a window. The setup uses a Nucleo F401RE development board and the Nucleo IKS01A1 sensor expansion board to mount the LMS6DS3 IMU on.

To test this several cases were defined, all of them starting with the door closed:

- **Door opened slowly.**
  - Door opened 30 degrees.
  - Door opened 60 degrees.
  - Door opened 90 degrees.
- **Door opened with moderate speed.**
  - Door opened 30 degrees.
  - Door opened 60 degrees.
  - Door opened 90 degrees.
- **Door opened quickly.**
  - Door opened 30 degrees.

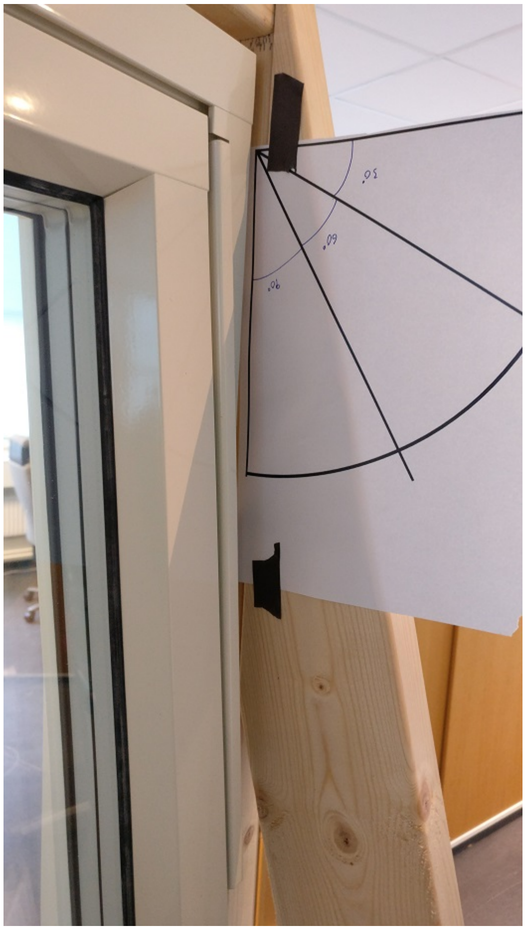


Figure 5.3: The setup for measuring the angle of the window.

  - Door opened 60 degrees.
  - Door opened 90 degrees.
- Door opened with various speed and held in a open state for a prolonged period of time.

The last case is to see the effects of the imperfect sensors and how they cause the system to drift over time. It is worth to note that the speed of which the door is opened is not the exactly the same for the different tests. This is because of human error. Unless a test setup where the door can be opened with accurate speed is used, there will be variance in the speed. However, this is not a problem and the difference in speed between the tests are still relatively small. The angle of which the door is opened is more accurately measured by using a paper on the floor with different angles written on it as seen in Figure 5.4.

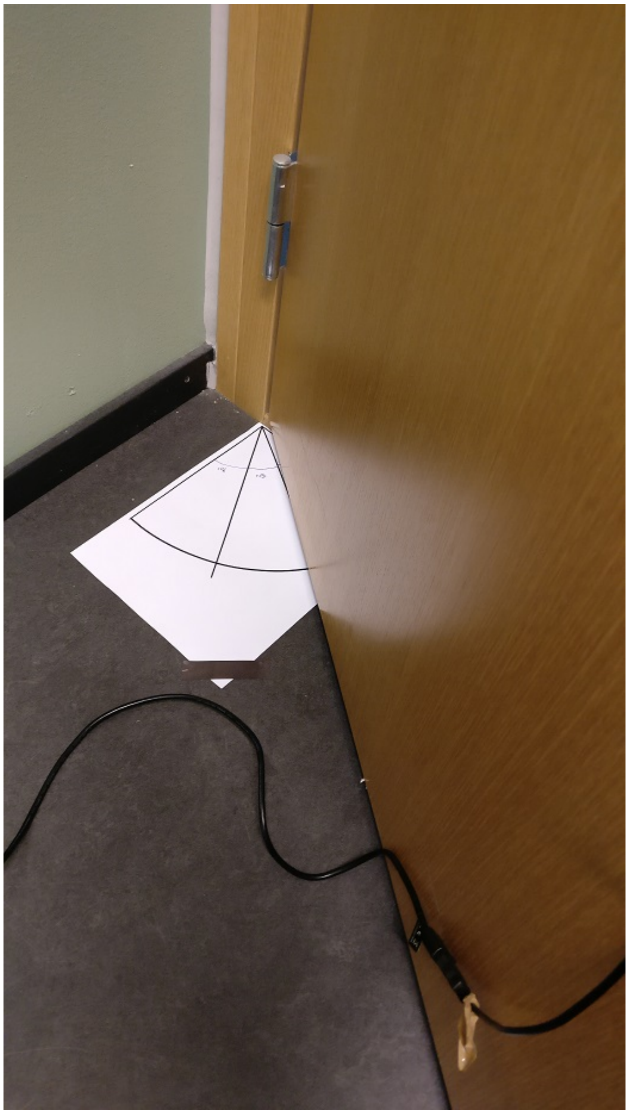


Figure 5.4: Test setup. The door is opened to an angle and held there before being closed again. The opening angle is written down and used to compare to the result from the algorithm.

5.3 Door Results

The results from the tests of the door is listed here. The results are tested with the following parameters As it is preferable to use a low sampling frequency to conserve energy the tests are conducted with the two lowest IMU settings for frequency. The first result shows tests where the door was opened to around 30°. The door is opened at various speeds from what was defined as a slow door opening to fast door openings. After the door was opened it was held at a certain angle to allow the angle to be measured before being closed again. This angle is compared to the results acquired from the algorithm in table 5.2. However, the measured result is done with a very crude method the certainty of the angle is only around 1°.

Parameter	Value	Unit
Door Radius	0.83	m
Sampling Frequency 1	52	Hz
Sampling Frequency 2	13	Hz

Table 5.1: Parameters for door tests.

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	30°	30.3°	0.3°
Test 2	31°	31.4°	0.4°
Test 3	32°	34.9°	2.9°
Test 4	28°	27.2°	0.8°
Test 5	31°	31.3°	0.3°

Table 5.2: Comparison between measured angle and the angle calculated by the algorithm. The test is run for a 30°door opening 5 times.

Secondly the algorithm was tested when the door was opened 60°. The tests were conducted in the same way as the first test. The results from the third test is summarized in Table 5.3 and Figure 5.6.

The last test was conducted for when the door was opened all the way to 90°. The test was conducted the same way as for the previous results. The results from the third test is summarized in Table 5.4 and Figure 5.7.

Tests were also done for the sampling frequency 13Hz. In this case the door was opened either 45°or 90°. In each case 3 test were done to compare to

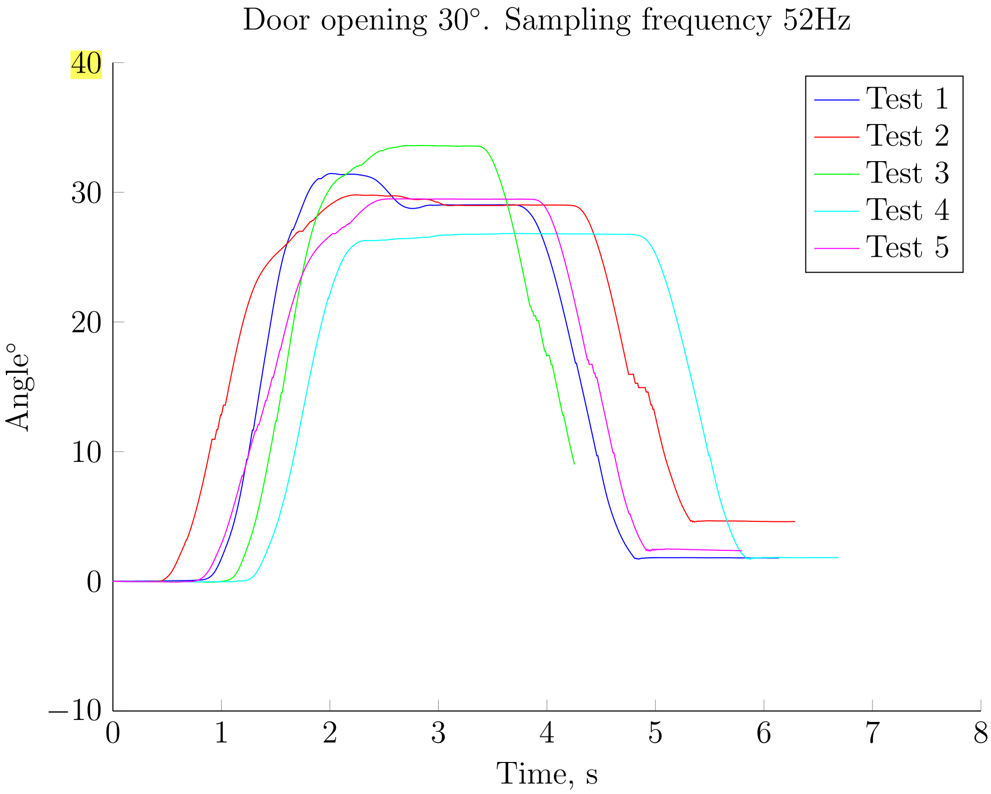


Figure 5.5: Door opening 30°, 5 tests. Sampling frequency 52 Hz

the real measured angle. The result of the case when the door was opened 45°are summarized in Table 5.5 and Figure 5.8. The results for the case of 90°door openings are summarized in Table 5.5 and Figure 5.6.

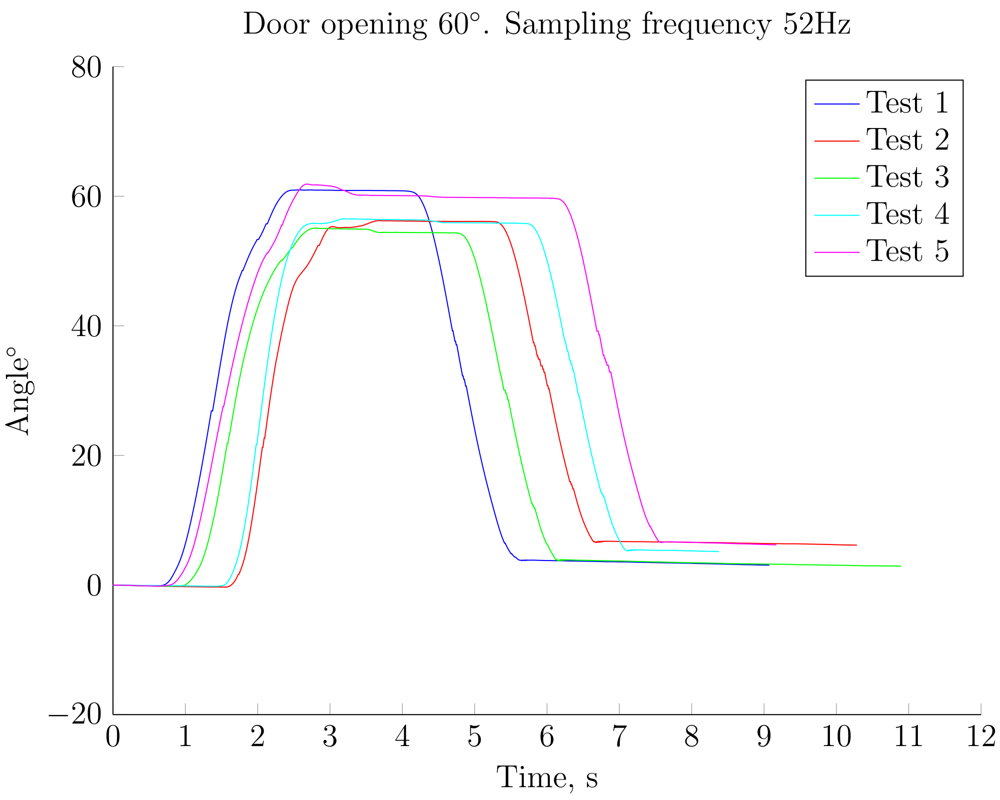


Figure 5.6: Door opening 60°, 5 tests. Sampling frequency 52 Hz

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	62°	63.2°	1.2°
Test 2	60°	58.3°	1.7°
Test 3	58°	55.8°	2.2°
Test 4	59°	56.4°	2.6°
Test 5	61°	60.0°	1.0°

Table 5.3: Comparison between measured angle and the angle calculated by the algorithm when the door is opened around 60°.

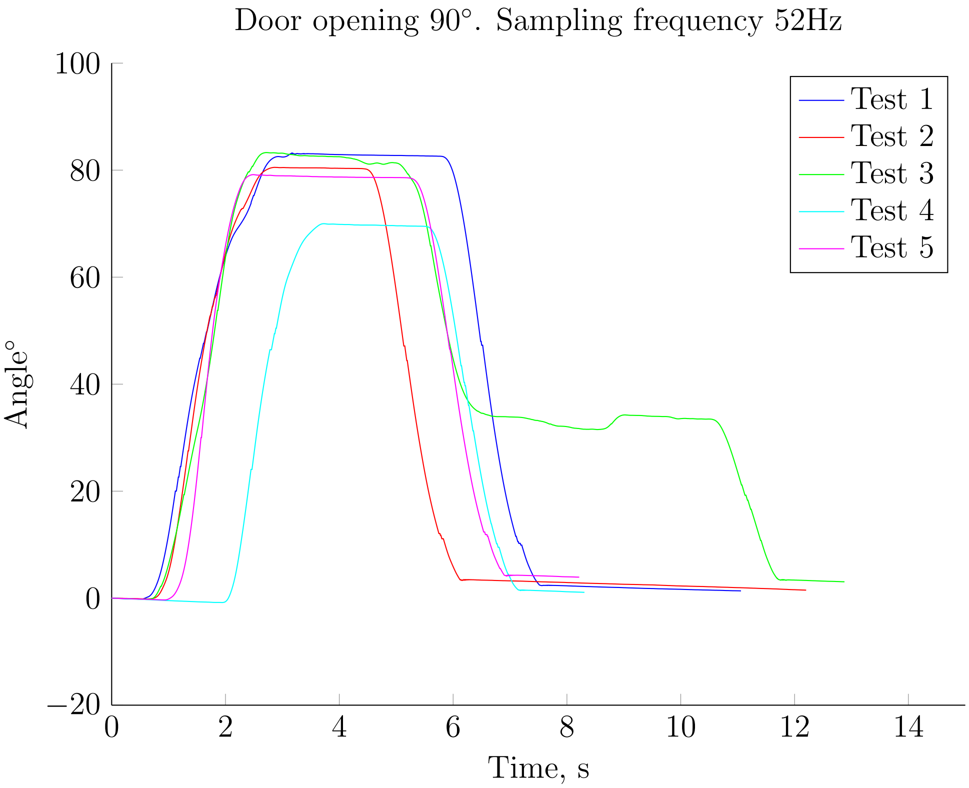


Figure 5.7: Door opening 90°, 5 tests. Sampling frequency 52 Hz

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	90°	87.3°	2.7°
Test 2	88°	84.5°	3.5°
Test 3	89°	85.0°	4.0°
Test 4	80°	75.8°	4.2°
Test 5	85°	81.1°	3.9°

Table 5.4: Comparison between measured angle and the angle calculated by the algorithm when the door is opened around 90°.

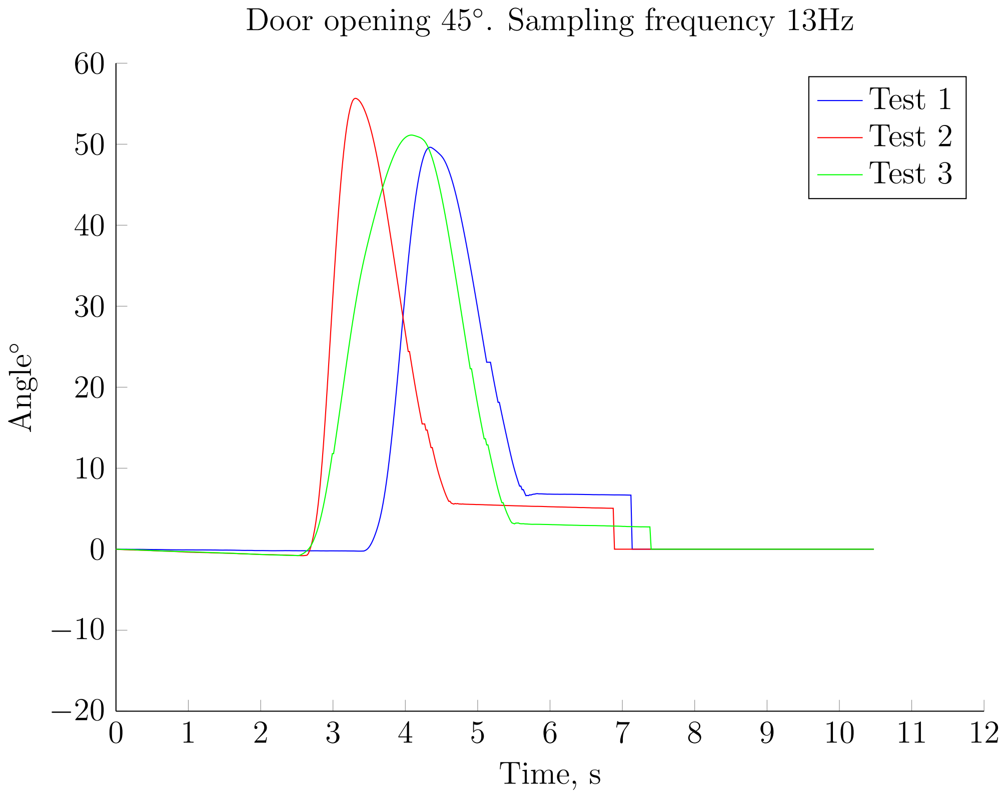


Figure 5.8: Door opening 45°, 5 tests. Sampling frequency 52 Hz. Door held still after being opened.

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	44°	49.6°	5.6°
Test 2	47°	55.6°	8.6°
Test 3	45°	51.1°	6.1°

Table 5.5: Comparison between measured angle and the angle calculated by the algorithm when a sampling time of 13Hz is used and the door is opened around 45°.

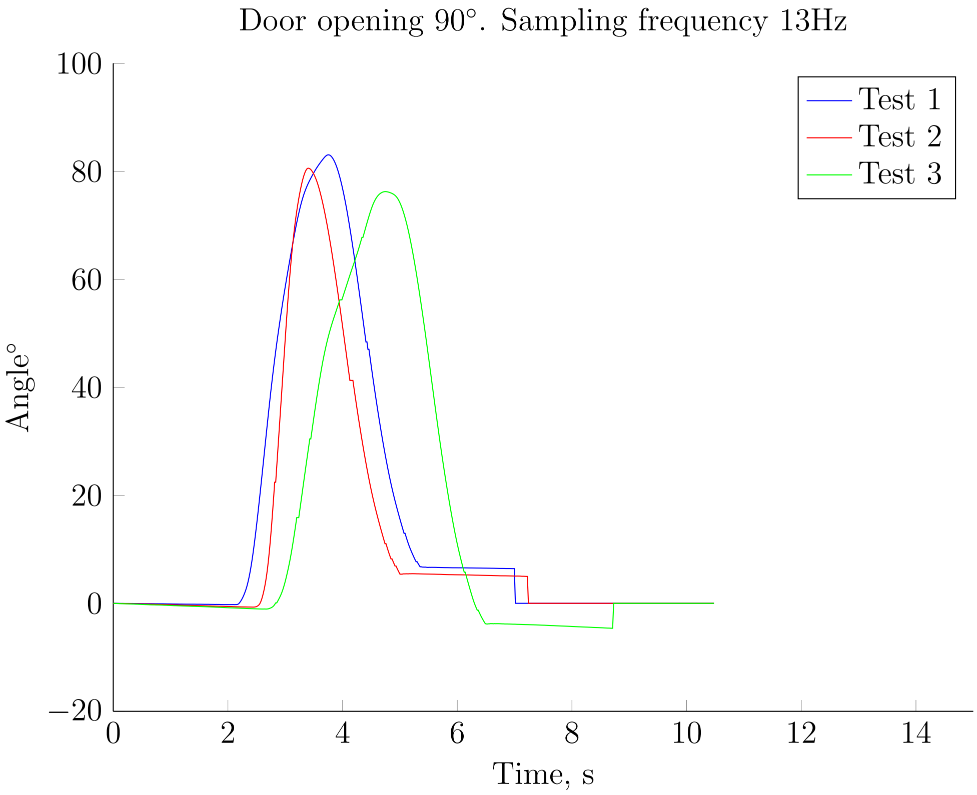


Figure 5.9: Door opening 90°, 3 tests. Sampling frequency 13 Hz.

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	89°	83.0°	6.0°
Test 2	86°	80.4°	5.6°
Test 3	82°	76.3°	5.7°

Table 5.6: Comparison between measured angle and the angle calculated by the algorithm when a sampling time of 13Hz is used and the door is opened around 90°.

5.4 Window Results

The calibration of the sensors is done the same way as for the door case. One thing that needs to be taken into consideration is the presence of the acceleration due to gravity for one of the sensors. In stationary conditions this will be constant but it affects the sensor calibration. Thus, to determine the offset, this needs to be removed. One way to do this is to turn the IMU so that the direction is unaffected by gravity. However, if the IMU is already mounted this will be impossible. If that is the case, one has to subtract the values due to gravity from the measurements. Gravity is dependent on the location on earth but can generally be described as a constant where a small change in this constant is due to location on earth. Similar to the way the door was tested, the window was tested by opening it to 30°, 60°, and 90°. The window that was used for tests has a lock that activates when the window is opened around 15°. For each test, this has to be unlocked which causes a pause in the angle of the window plots where the angle is not changing. The algorithm is initialized with the known values of the state vector $\mathbf{x_0} = \mathbf{0}$ and the covariance matrix is initialized as a identity matrix multiplied by a small constant $\delta = 0.1$. The measurement variance is given by the calibration of the sensor.

Parameter	Value	Unit
Window Radius	0.82	m
Sampling Frequency	52	Hz

Table 5.7: Parameters for window tests.

In the first test the window was opened to around 30° 5 times to test the algorithms ability to track the angle of the window. The result of the first test is shown in Figure 5.10 and Table 5.8. In the second test the window is opened to angles around 60°, the result of this is shown in Figure 5.11 and Table 5.9. In the third test the window was opened to around 90°and the result of this test is shown in Figure 5.12 and Table 5.10.

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	37°	30.3°	6.7°
Test 2	32°	31.4°	0.6°
Test 3	27°	34.9°	7.9°
Test 4	28°	27.2°	0.8°
Test 5	32°	31.3°	0.7°

Table 5.8: Comparison between measured angle and the angle calculated by the algorithm for the window when it is opened around 30°.

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	62°	65.0°	3.0°
Test 2	55°	46.8°	8.2°
Test 3	57°	49.5°	7.5°
Test 4	59°	54.3°	4.7°
Test 5	62°	55.2°	6.8°

Table 5.9: Comparison between measured angle and the angle calculated by the algorithm for the window when it is opened around 60°.

Test	Measured angle	Algorithm calculated angle	Δ
Test 1	89°	90.1°	1.1°
Test 2	88°	88.4°	0.4°
Test 3	89°	94.8°	5.8°
Test 4	84°	89.2°	5.2°
Test 5	87°	95.7°	8.7°

Table 5.10: Comparison between measured angle and the angle calculated by the algorithm for the window when it is opened around 90°.

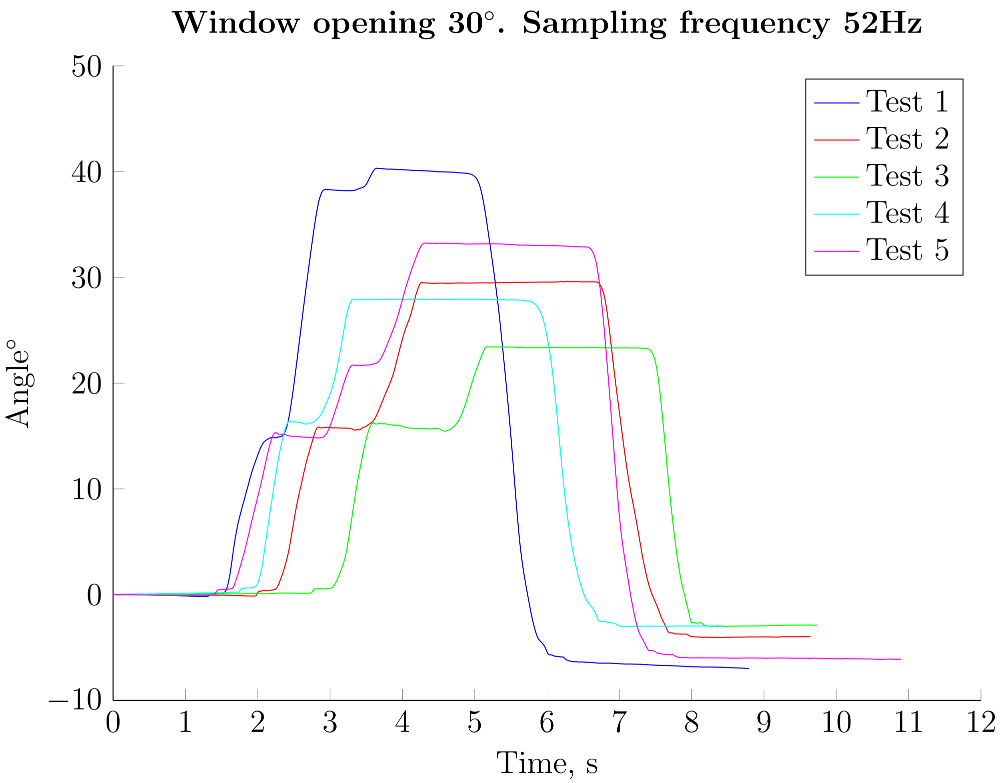


Figure 5.10: Window opening 30°, 5 tests. Sampling frequency 52 Hz.

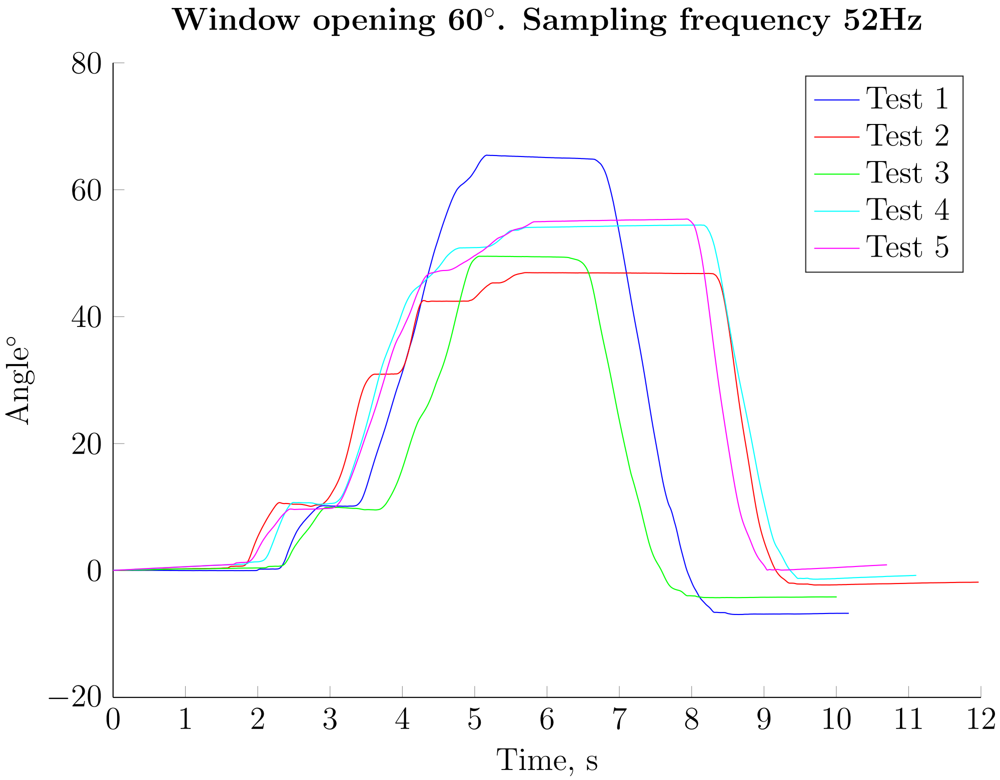


Figure 5.11: Window opening 60°, 5 tests. Sampling frequency 52 Hz.

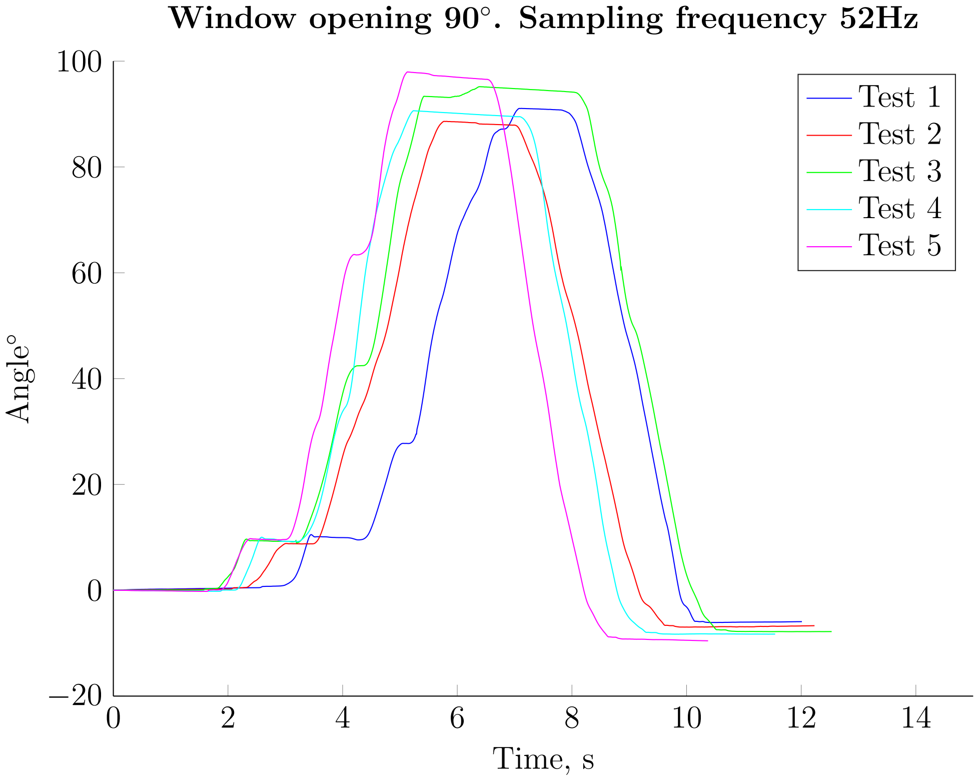


Figure 5.12: Window opening 90°, 5 tests. Sampling frequency 52 Hz. Window held still after being opened.

Chapter 6

Door Closing Classification Result

There are several parameters in the KNN classifier that affects how well classification can be achieved. Different combinations of the parameters were tested to see how they affected the classification. The parameters of the model is the number of neighbours k and the choice of p for the p-norm. They are compared by their classification accuracy for the training and validation sets. The two types of door movements were investigated by comparing the average classification accuracy for a range of different values of p and k. The accelerometer data was taken over several days for one door. Both of the two door movements are tested for each pair of values. To evaluate the best number of neighbours k the classification accuracy is plotted against k for a certain value of p.

An important detail of the algorithm is how it depends on the type of door that is used. In this experiment a door with a door closer is used. The purpose of the door closer is to automatically close the door and stop the door from hitting the frame too hard. By having a door closer the acceleration of the door is also limited. A door closer can increase the risk of the door getting stuck and make classification harder due to the smaller values of acceleration. It is still interesting to use this type of door to test on as it is common in office buildings and poses a bigger challenge than usual doors.

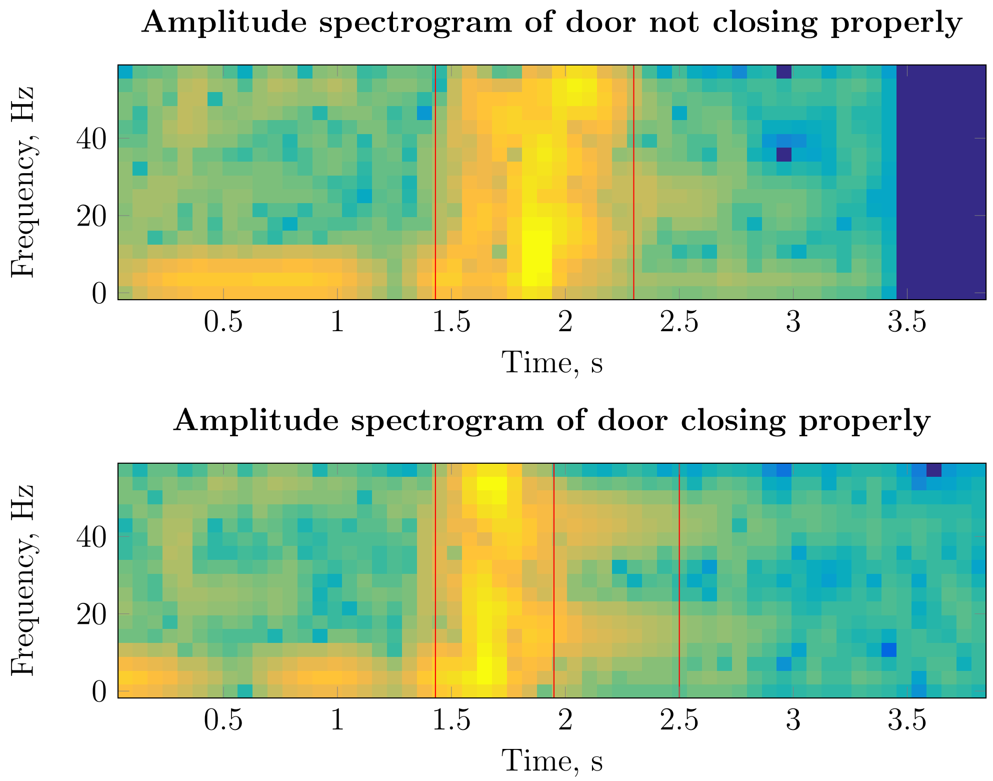
For testing, a database of 100 door closings was used. It was divided into two sets with one set containing cases where the door gets stuck as it is being closed and one set in which the door closes properly. Each of the two sets contain 50 door closings and all were done for the same door. As preprocessing the IMU was calibrated to remove the influence of offset and the scale-factor. The two sets were used as the training set. To further test the classification algorithm, different angles of which the door closes from was used. Because only the way the door closes is of interest the exact angle from which the door closes does not need to be measured strictly. The two sets of data can be summarized as:

- Database of 100 door closings
  - Subset with data containing cases where the door gets stuck
    1. Set of 10 door closings from an angle between 10° and 20°
    2. Set of 10 door closings from an angle between 30° and 40°
    3. Set of 10 door closings from an angle between 50° and 60°
    4. Set of 10 door closings from an angle between 70° and 80°
    5. Set of 10 door closings from an angle between 80° and 90°
  - Subset with data containing cases where the door closes properly
    1. Set of 10 door closings from an angle between 10° and 20°
    2. Set of 10 door closings from an angle between 30° and 40°
    3. Set of 10 door closings from an angle between 50° and 60°
    4. Set of 10 door closings from an angle between 70° and 80°
    5. Set of 10 door closings from an angle between 80° and 90°

The spectrograms that are used for training are partitioned into 4 different classes. The classes are chosen according to the type of information contained in the spectrogram for that time frame. They are summarized as:

- **Class 1:** Spectrogram content before the door hits the door frame.
- **Class 2:** Spectrogram content during the first impact of the door.
- **Class 3:** Spectrogram content after the first impact and the door has closed properly.
- **Class 4:** Spectrogram content after the first impact and the door did not close.

An example of how the classes are divided is shown in Figure 6. The figures show that when the door closes properly there is a distinct pattern in the spectrogram that can be used to classify if the door closes.



Amplitude spectrogram of door not closing properly

Amplitude spectrogram of door closing properly

Figure 6.1: Examples of how the spectrograms are divided into classes. The first figure shows an example of how the spectrogram looks when the door hits the door frame but does not close properly. The second figure shows the spectrogram of a door closing properly. The short time after the door hits the door frame is what distinguish the movement.

Classification is tested for different combinations of p and k. Since four different classes are used only even numbers of k are used. The p-norm is taken from $p = 2$ to $p = 15$ as no significant difference in performance occur with higher values of p. To evaluate the performance of different combinations the percentage of correctly classified spectrograms are compared to the values of p and k. The number of neighbours must be less than 20 as this is the number of data points in each feature vector.

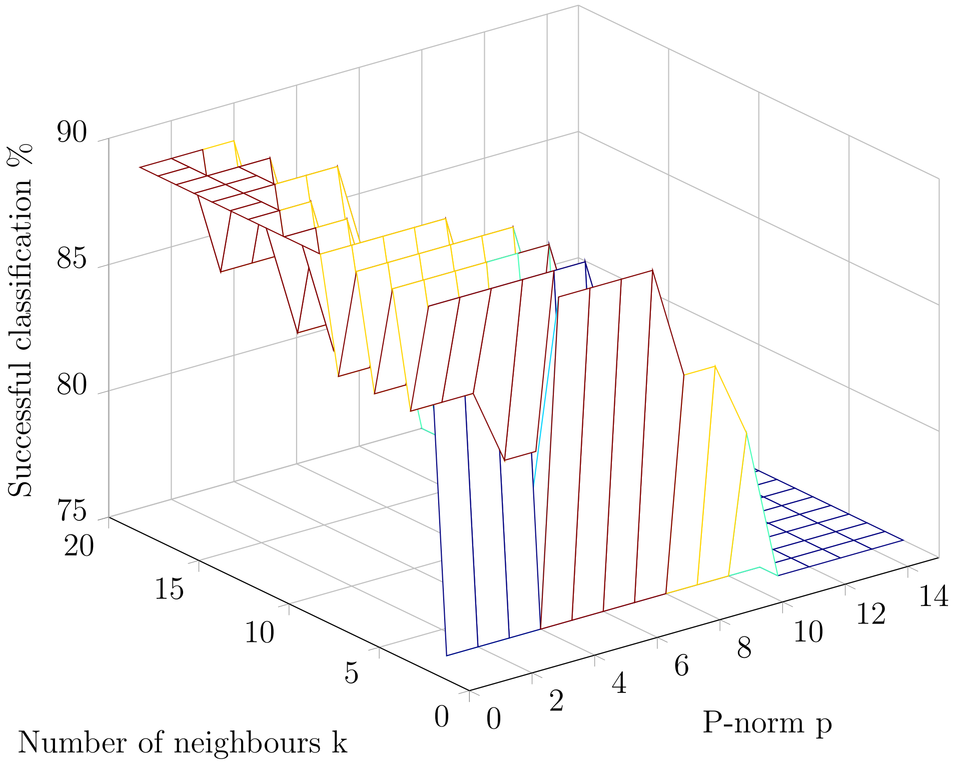


Figure 6.2: Results of the classification analysis. The values of the p-norm are between $p = 2$ and $p = 15$. The value k of the neighbours is varied between $k = 1$ and $k = 19$. A set of 50 spectrogram containing 25 faulty door closings and 25 proper door closings was used. A validation set of 20 spectrograms was compared to the training set to determine the number of successful classifications.

It was not possible to reduce the dimension of the feature vectors because to the sparse information contained in them.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The algorithms perform better if the sensor board is placed so that the radius is maximized. Because it is intended for the sensors to be placed inside the handle of the door and window the radius is almost the same as the length of the window or door so that the radius is sufficiently large to achieve realistic results. The algorithm performs better when it is run with a higher sampling frequency as is apparent from Table 5.2-5.6. The difference is notable as the accuracy of the angle is improved by several degrees with a higher sampling frequency. In Figure 5.8-5.9 it is also notable that the door closing is not fully detected as there is a sudden drop in the angle when the door hits the door frame. It is possible that this is caused by the gyroscope being affected by the vibrations caused by the door frame. Another reason for the inaccuracy for both sampling frequencies is that the sampling time is not exactly what is specified in the datasheet of the sensor. In fact, the company that build the sensor has stated that there can be up to 10% change from the specified sampling frequency. This could be corrected by using the platform's internal clock to give a more accurate sampling time.

The window angle accuracy is also worse than the door angle accuracy. A reason for this could be the modified model not properly attenuating earth's gravity or that the window's lock mechanism is causing a spike in the accelerometer and gyroscope that the model is not able to remove. One solution to improve the window algorithm could be to weight the accelerometer data more to use it as a way to directly detect the angle from the two axes relative gravity readings.

The door closing classification analysis shows that the highest classification rate is achieved when the p-norm is decreased and when the number of neighbours k is uneven. The increased classification rate for uneven numbers of k is expected as there are two clusters. The highest classification rate is given for the p-norm with $p = 1$ to $p = 4$. The best value of the p-norm is dependent on the type of data. The data in this case is in $\mathbb{R}^{17}$ which makes it difficult to see why one value of p would be better than another.

7.2 Future Work

The classification algorithm can be further improved on by using smart event detection scheme. As the angle is known to a certain degree it is possible to detect the door closing by looking at the angle of the door and spikes from the accelerometer. By doing this the classification time can be decreased. The time it takes for the door to close and become stationary is always around a second it is only interesting to calculate the spectrogram just before any spikes in the accelerometer until it has become stationary. However, relying only on the accelerometer for this would not be robust because spikes can arise at any time due to people accidentally hitting the door with their feet or bumping the door into walls. That is why incorporating the angle is necessary.

Different fields of application could be tried such as using artificial intelligence to detect if the door is moving slowly or is about to break. By using a sampling frequency that is not constant but dependent on when the real measurement is taken could also improve the accuracy of the algorithms.

Appendices

Appendix A

Sensor Comparison

SF± xg	Scale Factor for the ±x range
CC	Current Consumptions (operational)
ODR	Output Data Rate given in Hz
NL	Non-linearity
ARS	Angular Rate Sensitivity
LSB	Least Significant Bit
Acc	Accelerometer
Gyr	Gyroscope
Magn	Magnetometer

IMU	ADXL362 [17]	FXOS8700CQ [18]	LSM6DS3 [21]
Axes	3, Acc	6, Acc and Gyro	6, Acc and Gyro
Range [g]	±2 to ±8	±2 to ±8	±2 to ±16
SF ±2g [mg/LSB]	1	0.244	0.061
SF ±4g [mg/LSB]	2	0.488	0.122
SF ±8g [mg/LSB]	4	0.976	0.244
SF ±16g [mg/LSB]	-	-	0.488
CC	1.8μA	240μA at 100Hz	70μA at 104Hz
ODR [Hz]	100	1.5 to 800Hz	0
NL [%/C°]	±0.05	±0.01	-
ARS	-	-	4.375 to 70 mDPS/s

IMU	LSM9DS1 [23]	LSM303AGR [20]	LSM9DS0 [22]
Axes	9, Acc, Gyro, Magn	6, Acc, Magn	9
Range [g]	±2 to ±16	±2 to ±16	±2 to ±16
SF ±2g [mg/LSB]	0.061	3.9	0.061
SF ±4g [mg/LSB]	0.122	7.82	0.122
SF ±8g [mg/LSB]	0.244	15.63	0.244
SF ±16g [mg/LSB]	0.732	46.9	0.732
CC	310μA at 119Hz	250μA at 100Hz	310μA at 119Hz
ODR [Hz]	10 to 952Hz	100 to 150Hz	10 to 952 Hz
NL [%/C°]	-	0.01	-
ARS	-	-	-

Bibliography

[1] Francisco Bonin-Font, Joan-Pau Beltran, and Gabriel Oliver. Multisensor aided inertial navigation in 6dof auvs using a multiplicative error state kalman filter. *OCEANS - Bergen, 2013 MTS/IEEE*, 2013.

[2] John Weston David Titterton. *Strapdown Inertial Navigation Technology.* Institution of Engineering and Technology, 2004.

[3] Yibo Feng, Xisheng Li, and Xiaojuan Zhang. An adaptive compensation algorithm for temperature drift of micro-electro-mechanical systems gyroscopes using a strong tracking kalman filter. 2015.

[4] Fredrik Gustafsson. *Statistical Sensor Fusion.* Studentlitteratur AB, 2010.

[5] M Jafari, M Sahebjameyan, B Moshiri, and T A Najafabadi. Skew redundant mems imu calibration using a kalman filter. *Measurement Science and Technology*, 26, 2015.

[6] Volker Kempe. *Inertial MEMS: Principles and Practice.* Cambridge University Press, 2011.

[7] Wassim Khoder, Bienvenu Fassinut-Mombot, and Mohammed Benjelloun. Quatemion unscented kalman filtering for integrated inertial navigation and gps. 2008.

[8] Peerapol Khunarsal, Chidchanok Lursinsap, and Thanapant Raicharoen. Very short time environmental sound classification based on spectrogram pattern matching. 2013.

[9] Manon Kok. *Probabilistic modeling for positioning applications using inertial sensors.* PhD thesis, Linköping University, 2014.

[10] Edgar Kraft. A quaternion-based unscented kalman filter for orientation tracking. *Sixth International Conference of Information Fusion*, 1:47–54, 2003.

[11] Oliver Kramer. *Dimensionality Reduction with Unsupervised Nearest Neighbors*. 2013.

[12] A Olivares, J M Górriz, J Ramirez, and G Olivares. Accurate human limb angle measurement: sensor fusion through kalman, least mean squares and recursive least-squares adaptive filtering. *Measurement Science And Technology*, 22(2), 2011.

[13] Davide Scaramuzza Roland Siegwart, Illah Reza Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press Ltd, 2011.

[14] Simone Sabatelli, Marco Galgani, Luca Fanucci, and Alessandro Rocchi. A double stage kalman filter for sensor fusion and orientation tracking in 9d imu. 2012.

[15] Simone Sabatelli, Francesco Sechi, Luca Fanucci, and Alessandro Rocchi. A sensor fusion algorithm for an integrated angular position estimation with inertial measurement units. 2011.

[16] Glauco Garcia Scandaroli and Pascal Morin. Nonlinear filter design for pose and imu bias estimation. pages 4524–4530, 2011.

[17] Analog Semiconductors. Adxl362. micropower, 3-axis, ±2g /±4g /±8g digital output mems accelerometer, 2016.

[18] NXP Semiconductors. Fxos8700cq 6-axis sensor with integrated linear accelerometer and magnetometer, 2016.

[19] STMicroelectronics. *Parameters and calibration of a low-g 3-axis accelerometer*, 2014.

[20] STMicroelectronics. Lsm303agr. ultra-compact high-performance ecompass module: ultra-low-power 3d accelerometer and 3d magnetometer, 2016.

[21] STMicroelectronics. Lsm6ds3. inemo inertial module: always-on 3d accelerometer and 3d gyroscope, 2016.

[22] STMicroelectronics. Lsm9ds0. inemo inertial module: 3d accelerometer, 3d gyroscope, 3d magnetometer, 2016.

[23] STMicroelectronics. Lsm9ds1. inemo inertial module: 3d accelerometer, 3d gyroscope, 3d magnetometer, 2016.

[24] Roberto G. Valenti, Ivan Dryanovski, and Jizhong Xiao. Keeping a good attitude: A quaternion-based orientation filter for imus and margs. *Sensors*, (15):19302–19330, 2015.

[25] Andrea Vitali. *6-point tumble sensor calibration*, 2015.

[26] Jinling Wang Wei Li. Effective adaptive kalman filter for mems-imu/magnetometers integrated attitude and heading reference system. *THE JOURNAL OF NAVIGATION*, 66:99–113, 2013.

TRITA EE 2017:080
ISSN 1653-5146

www.kth.se