

# Characterization of low-cost Accelerometers for Use in a Local Positioning System

Morten Stakkeland



Master Thesis

Department of Physics  
University of Oslo

31. July 2003

	Acknowledgements .....	i
	Mathematical Notation .....	ii
1	Introduction.....	1
	1.1 Position Acquisition Systems.....	1
	1.2 Inertial Navigation.....	2
	1.3 Thesis Outline.....	3
2	Background and Theory .....	4
	2.1 The Local Positioning System (LPS) .....	4
	2.2 Inertial Tracking and Navigation.....	7
	2.3 Implementations of an Inertial System.....	17
	2.4 Accelerometer Characterization.....	18
	2.5 Estimation and Smoothing .....	22
3	Materials and Methods .....	29
	3.1 The Accelerometers.....	29
	3.2 Temperature Control Systems .....	33
	3.3 DAQ systems.....	39
	3.4 Experimental Environment.....	44
	3.5 The Static Experiments.....	45
	3.6 Experiments using dynamic Accelerations .....	58
	3.7 Simulations of a Three-dimensional Inertial Tracking System .....	72
	3.8 Simulations of a Local Positioning System (LPS) .....	78
4	Results.....	84
	4.1 Results from Experiments using Static Positions.....	84
	4.2 Results of Dynamic Experiments .....	93
	4.3 Results from Simulations of an Inertial Tracking System.....	102
	4.4 Results from Simulations of a Local Positioning System .....	107
5	Discussion.....	116
	5.1 Data Acquisition Setup.....	116
	5.2 Static Experiments.....	117
	5.3 Dynamic Experiments .....	119
	5.4 Simulations of an Inertial Tracking System .....	121
	5.5 Simulation of a local positioning system .....	123
	5.6 General Discussion.....	124
6	Conclusion and Recommendations.....	125
	References.....	I

## Appendixes

A	Matlab Code .....	III
B	The Analog Signal Conditioning Circuit.....	XIV
C	Extracts from the Accelerometer Datasheets .....	XV
D	The IEEE Accelerometer Model Equation .....	XVI

## **Acknowledgements**

This thesis was presented for the degree Candidatus Scientarum at the Department of Physics of the University of Oslo.

The work described here was a part of a project at ABB Corporate Research. Some of the work has also been done in cooperation with SINTEF Electronics and Cybernetics. The work on the thesis started in October 2001, and has been going on until this date.

I would first of all like to thank my mentors at ABB, Steve T. Pedersen, Gunnar Prytz and Knut Asskildt, who have supported me continuously through the process and given good feedback in all phases of the thesis. They have taken my work seriously from the beginning.

I would also like to thank to my mentor at the physics department, Torfinn Lindem, for handling administrative tasks and giving feedback on technical solutions and the form of the thesis.

Also thanks to Wilfred E. Booij, Geir Uri Jensen and Dag Wang from SINTEF for support and inputs on sensors and experimental procedures.

At last I would like to thank the students Frank Bakken and Sigmund Gundersen from Høyskolen i Oslo, who provided valuable craftsmanship during the building of the dynamic experimental setup.

## Mathematical Notation

Vectors are written as underlined lower-case characters, for instance  $\underline{v}$ .

Matrices are written as upper-case characters for instance  $D$ .

A vector with a subscript as  $\underline{v}_a$  means that the vector is defined in the coordinate system given the subscript  $(a)$ .

A Direction Cosine Matrix (DCM) which transforms a vector from the coordinate system  $a$  to the coordinate system named  $b$ , is given 'a' as subscript and 'b' as superscript,  $D_a^b$ .

The rotation of the coordinate system  $a$  with respect to the coordinate system  $b$  expressed in the coordinate system  $b$  is denoted  $\underline{\omega}_{ab}$ .

The skew-symmetric equivalent of the rotation vector is denoted written as upper-case omega  $\Omega_{ab}$ .

The norm of a vector is written as  $|\underline{v}|$  and is defined by  $|\underline{v}| = \sqrt{\sum (v_k)^2}$

The derivative of a vector or scalar with respect to time is written with as  $\frac{d}{dt}\underline{v} = \dot{\underline{v}}$  and

$$\frac{d^2}{dt^2}\underline{v} = \ddot{\underline{v}}$$

# 1 Introduction

## 1.1 Position Acquisition Systems

**The main goal of the project which this thesis is a part of, is to develop a system that can track hand movements within a restricted area. The last few years has seen the development of several different relatively cheap systems with the ability to track position, velocity and heading of a device. These systems are usually based on ultrasound, electromagnetism or cameras with image recognition capabilities. The camera based systems can only function if there is a clear line of sight to all of the cameras. Ultrasonic systems also need line of sight between receiver and transmitter in order to achieve the wanted accuracy. The line-of-sight problem can to some extent be solved by integrating inertial sensor components into the system.**

The inertial system can be used to improve the performance of the total system, and as an alternative in the 'out-of-sight' case. The data from the primary system then provides the initial position and velocity needed to integrate the acceleration data correctly.

The main problems with inertial systems are the high cost of the high accuracy systems and the low accuracy of the low-cost systems. The emerging micromachined inertial components, gyros and accelerometers, makes it reasonable to consider whether these components can be used in an inertial tracking system. Micromachined components of today still have a quite poor performance compared with conventional mechanical and optical components, but the accuracy is increasing and they are predicted to replace many of the systems using conventional mechanical components in the future (Barbour and Schmidt, 1998).

The main advantages of mass produced Micro Electro-Mechanical Devices (MEMS) are their relatively low cost and small size. Using low-cost MEMS devices in an Inertial Measurement Unit (IMU) would make the price of the unit an order of magnitude lower than the systems that are considered low-cost today.

The sensor errors in an IMU can be modelled mathematically to a certain extent (Titterton and Weston, 1997), and hence be compensated for if the errors are repeatable. One way to do this is to run each sensor through a series of laboratory tests, to estimate the input-output characteristics of each sensor. This method may be complicated and time-consuming, but may increase the performance of the IMU significantly. The experimental tests may also provide more detailed information than the unit datasheets, which often are given for a wide temperature range and have conservative estimates of the errors.

The challenge is to develop experimental setups that can be used to test and characterize accelerometers. The experiments described in this thesis concentrates mainly on characterization of linear accelerometers, and characterization of micromachined gyros is left for later. The goal of the experiments is to obtain models of low-cost accelerometers. How much can be learned from the experiments? What errors are repeatable and can thus be compensated for? What errors have to be

modelled as random variables? Are the parameters constant over longer periods of time?

## 1.2 Inertial Navigation

**The fundamental principle of inertial navigation is to measure the applied accelerations and integrate them twice to obtain the velocity and position,**

**$\underline{a}(t) = \ddot{\underline{r}}(t)$ . The accelerations are measured with an Inertial Measurement Unit (IMU). In a *strap-down* solution, see section 2.3.2, the IMU measures the linear accelerations along three orthogonal axes and the rotations about the same axes. The measured accelerations then have to be transformed to a chosen reference frame to do the integration.**

Inertial navigation systems are used in fields like aviation, space travel, submarines, long range rockets and other systems that need to keep track of heading and position. These systems are usually more accurate and more expensive than the systems considered in this thesis.

Data from inertial navigation systems are often combined with data from other sources in an estimation algorithm, for instance a Kalman filter. Data from a global positioning system (GPS) is a common solution, as described by Farrel and Barth (1998) and Shin (2001). In the application considered in this thesis the data from the inertial system is combined with data from a local positioning system (LPS) as described in the next section.

An inertial system is an unstable system by nature because of the double integration needed to get the position from the measured acceleration. A small offset, often called bias, in the measured acceleration will be integrated twice and hence give error which is proportional to  $t^2$ . An offset in one of the gyros could in worst case give an error in the estimated position proportional to  $t^4$  (Titterton and Weston, 1997). This means that the system can stay within a specified accuracy only for a limited period of time.

The theoretical background developed for global navigation applications given for instance by Titterton and Weston (1997) will be used in this thesis. The question is how the navigation equations described in the background literature best can be implemented for this kind of system, a local coordinate system fixed relative to the earth.

The scope is to choose a reference frame suitable for the application, solve the navigation equations and implement a discrete-time numerical solution in MATLAB. The state of the system does not have to be known in real time, which means that the data processing can start after the system has stopped logging the sensor outputs. The developed algorithms will then be used to do simulations using accelerometer models based on results from the conducted experiments. How do accelerometer errors influence the performance of an inertial navigation system? Can the performance be improved by extensive testing and modelling of each individual component?

If the development of micromachined inertial parts follows the development described by Barbour and Schmidt (1998), what performance can be achieved with future components? This thesis will focus on these questions.

## 1.3 Thesis Outline

Several different reference frames will be described, and one of them will be chosen for the application. The navigation equations will be listed and solved for the chosen reference frame, and some possible numerical implementations of the navigation equations will be outlined.

Two different experimental setups will be tested, and used to characterize the low-cost ADXL105 accelerometer. One of the setups will measure the static specific force due to gravity to characterize the accelerometers, this procedure is called a multi-position tumble test. The second experimental setup will use dynamic accelerations in one direction and a capacitive position sensor to calculate the applied acceleration.

The capacitive position sensor will also be used to do simulations of a one-dimensional local positioning system (LPS) consisting of an inertial system and a primary system, by using the position data to simulate a primary system with different specifications. The accelerometer output and the measurement updates from the primary system will be combined using a linear Kalman filter. These simulations which will be based on real accelerometer outputs, will also be used to model the variations in the accelerometer parameters as random processes.

A virtual environment will be constructed, using a numerical implementation of the navigation equations solved in chapter two. The accelerations and rotations sensed by an inertial measurement moving in a given path in a local reference frame will be calculated, and accelerometer error models will be used to simulate the output from a model of an IMU. The data will then be run through the navigation algorithms to measure the performance of the system.

*Chapter two* contains background theory needed to perform the experiments and simulations described in the thesis. Some specifications on the local positioning system are listed, and the navigation equations are derived and solved for a reference frame fixed relative to the earth. The next sections are theory on accelerometers and accelerometer characterization, and the last part of the chapter contains theoretical background on estimation and smoothing.

*Chapter three* describes the experiments conducted in order to characterize accelerometers and their errors. It contains descriptions of static and dynamic experiments, and of the performed simulations.

*Chapter four* contains a summary of the results obtained from the conducted experiments and simulations.

*Chapter five* discusses the results.

*Chapter six* contains the conclusion of the thesis, and points out possible future work.

## 2 Background and Theory

This chapter contains theory and background knowledge needed to conduct and understand the simulations and experiments described in this thesis.

The first section describes the aim of the project, and contains some information about possible technical solutions. A short description of several available local positioning systems is given to show possible technical solutions and for completion, but this thesis concentrates on inertial systems.

The second part is dedicated to inertial tracking and navigation. A reference frame will be chosen and the inertial navigation equations solved for that frame. The solution will in the next chapter be implemented numerically, and used in simulations.

The following section describes some algorithms from the field of estimation which also will be implemented numerically in the next chapter. Optimal estimation using a Kalman filter is described, and also smoothing and Kalman smoothing.

### 2.1 The Local Positioning System (LPS)

The local positioning system described in this thesis is a part of a project where a inertial navigation system is to be designed and implemented as a part of an LPS. The system should be able to track the absolute position, velocity and heading of a hand-held device with relatively good accuracy.

The unit for which position and heading is to be kept track of is going to be hand-held. This makes several simplifications and generalizations possible. These are listed in section 2.1.1. The position and heading does not have to be known in real time, the trajectory and heading can be calculated after the run experiment, which makes further simplifications possible.

Several different technologies are used in local positioning systems, and the technical solutions for this system have not yet been chosen. It is possible to use a single system as a stand-alone solution, or more subsystems can be combined in a system, interpolating the results using a Kalman filter as described in section 2.5. For instance an ultrasonic system can be combined with an inertial system, which would be similar to combining GPS and inertial systems in global navigation applications as described by Farrel and Barth (1998) or Shin (2001). The theoretical background for this kind of systems are well developed and described in the literature.

Ultrasonic, camera-based systems and magnetic heading systems are described briefly in section 2.1.2. Models of these systems are described in section 2.1.5, and will be used in simulations later.

#### 2.1.1 Possible Assumptions and Simplifications on the System

Given the nature of the system, several simplifications can be made. These are listed below.



### **Assumptions for a hand-held system**

- The measured acceleration can be limited to  $[-2\text{ g}, 2\text{ g}]$ , and the rotation of the unit can be limited to  $[-180^\circ/\text{s}, 180^\circ/\text{s}]$ . The tracking unit cannot strike against any surfaces, which most likely would cause shocks where the acceleration will exceed these limits
- The hand will damp most vibrations. Vibration-induced errors can be ignored, and the vibration sensitivity of the inertial sensors does not have to be measured and modelled.
- The bandwidth of a wrist is normally below 12Hz. Components with higher frequencies in the signal may be filtered out.

### **Assumptions for a LPS without real-time processing**

- All movement will be constricted to a local area. The longitude, latitude and altitude can be considered fixed and known in the system. Thus the Coriolis acceleration can be calculated with approximately the same geographical data, given by the system operator.
- The local gravity can be approximated as constant over the whole area
- The calculations do not have to be done real time, meaning that unlimited calculation power is available. All algorithms can be implemented at a high order.
- Kalman smoothing can be used to calculate the track and heading. This is described in the section 'Estimation and Smoothing'.

## **2.1.2 Some primary Systems**

### **Ultrasonic Systems**

These system use the travelling time of ultrasound between a transmitter and a receiver to calculate the distance between them. This process is called trilateration. The position with more degrees of freedom can be found by interpolating the data from several receivers placed at different positions in the area. The phase difference between the signals can be used to calculate the position.

### **Vision-based Systems**

Camera based systems usually use image recognition systems, and calculate the distance to an object with a known size by measuring its image size. At the other end of the performance and price range are laser based systems, which can measure distance very accurately.

### **Magnetic Heading Systems**

Systems based on the direction of a magnetic field functions basically like a compass. The direction is obtained either from the static magnetic field of the earth, or a generated stronger magnetic field, either static or dynamic. The main disadvantage with these systems is that the magnetic fields can be disturbed and changed by strong currents or metal objects nearby.

### 2.1.3 Out-of-sight Problems

Both ultrasonic and vision-based systems described in the previous sections need to have a clear line of sight between the transmitter and the receiver. If this line is blocked by the operator or other objects, information is lost, and the system will not work.

This puts great demands on how the system should be built, and restricts the operator movement. A secondary system is probably needed to get the heading in out-of-sight periods.

If the number of points which position has to be tracked is increased from one point to three as described in the next subsection, the line-of-sight problems will get worse. If two of the points are situated in the same plane as one of the cameras or receivers, they might obstruct each others line-of-sight.

### 2.1.4 Getting Heading Information with an Ultrasonic or Vision-based System

As long as there is a clear line of sight to the cameras or the ultrasonic receivers, the position of the system can be obtained from the position of one receiver. In order to calculate the heading of the system with six degrees of freedom, the position of three separate points has to be known. The heading can then be extracted as shown in figure 2.1. The two vectors  $\underline{v}_1$  and  $\underline{v}_2$  define the heading of the system. With two points, the number of degrees of freedom is limited to five, but the calculations are simplified as only the direction of one single vector has to be kept track of.

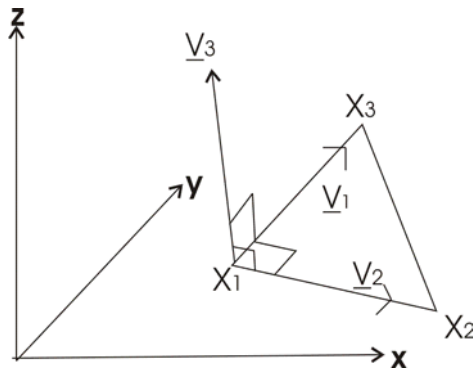


Figure 2.1: Heading information can be obtained by getting the position of three points.

In one dimension the heading error can crudely be approximated to be

$$\varepsilon = \text{atan}\left(\frac{2\sigma}{L}\right) \quad (2.1)$$

where  $\sigma$  is the standard deviation in the measurement of one points position, and  $L$  is the distance between the point. If  $\sigma$  equals 1 cm and the distance between the points are 10 cm, the error is given by  $\text{atan}(0.1)=11.3$  degrees which is a relatively big error. Getting heading information with this method puts huge demands on the accuracy of the positioning system.

### 2.1.5 Model of Ultrasonic and Vision-systems

The primary position acquisition system can be modelled by the equation  $\underline{z}=\underline{x}+\underline{w}$ , where  $\underline{z}$  is the measured position of the device,  $\underline{x}$  is the real position and  $\underline{w}$  is the measurement noise. The process noise is estimated to be zero-mean and Gaussian for simplicity, and can be simulated with Monte Carlo analysis. In a ultrasonic system, the variance of the measurement noise may vary as a function of the distance between the receiver and the transmitter, but is estimated to be constant.

A Direction Cosine Matrix (DCM), described in section 2.2.3, can be made up from the vectors  $\underline{v}_1, \underline{v}_2$  and the cross product of the two vectors,  $\underline{v}_3$ , as  $D=[\underline{v}_1, \underline{v}_2, \underline{v}_3]$  if the three vectors are orthonormal. The vectors will most likely neither be orthogonal nor normal due to the measurement errors, and will have to be orthogonalized and normalized through some kind of extrapolation algorithm.

The estimated DCM can be written in terms of the true DCM and an error term as follows:

$$\tilde{C} = BC \quad (2.2)$$

For small misalignment angles, the error matrix  $B$  can be approximated as a function of an identity matrix and a skew-symmetric matrix (Titterton and Weston, 1997):

$$B = (I - \Psi) \quad (2.3)$$

$$\Psi = \begin{bmatrix} 0 & -\delta\gamma & \delta\beta \\ \delta\gamma & 0 & -\delta\alpha \\ -\delta\beta & \delta\alpha & 0 \end{bmatrix} \quad (2.4)$$

The elements  $\delta\alpha$  and  $\delta\beta$  corresponds to attitude errors with respect to the vertical, or tilt errors, while  $\delta\gamma$  represents the heading error. In simulations these variables can be modelled as zero-mean Gaussian random variables with variance given as  $\varepsilon$  in equation (2.1).

## 2.2 Inertial Tracking and Navigation

### 2.2.1 Main Principles and Applications

Inertial tracking and navigation is based on the concept that the acceleration equals the position twice differentiated,  $\frac{d^2}{dt^2}\underline{r} = \ddot{\underline{r}} = \underline{a}$ . In a Newtonian coordinate system, the position of a vehicle can then be found by measuring the initial position, velocity and the applied acceleration, and integrate it with respect to time as

$$\underline{x}(t) = \underline{x}(t_0) + \underline{v}(t_0) \cdot (t - t_0) + \int_{t_0}^t \int_{t_0}^{\tau} \underline{a}(\tau) d\tau dt \quad (2.5)$$

By using gyroscopes which measure the applied rotation, the heading of the vehicle can also be calculated. This is necessary in strap-down solutions, described in section 2.3.2 in order to transform the measured acceleration from the body frame to the local coordinate system, see section 2.2.3.

Inertial navigation has developed since Newton formulated the laws of mechanics and gravity in the seventeenth century, and made several advances in the 1950's and 1960's as better sensors were made. The discipline has later been further developed and is now used in a wide range of military and civilian applications like long-range rocket systems, submarines, on ships and in aviation. These are for the most part high performance applications, and use high cost components. As mentioned in the introduction, this thesis will focus on low-performance, low-cost components.

It is chosen to distinguish between the terms inertial *navigation* and inertial *tracking*. The main goal of inertial navigation is to find or estimate the current position of the vehicle, while inertial tracking is concerned with calculating the path and heading after the vehicle or unit has finished running, making the assumptions listed in section 2.1.1 valid. Inertial systems is chosen as a common name for both methods in this thesis.

## 2.2.2 Choosing a Coordinate System

A set of coordinates has to be defined in order to be able to follow paths and positions in space. The choice of coordinate system should be based on the nature of the given application. A satellite revolving around the sun, may use a coordinate system with origo in the mass centre of the solar system, while one orbiting around the earth may use ECEF coordinates as described in section 2.2.7.

The selection of an appropriate coordinate representation is a matter of finding a system suitable for the given application.

The following subsections shortly describe some frames commonly used in inertial navigation. The different frames are described in several different ways with varying names in the references, and the frames described here are a combination of the frames given by Titterton and Weston (1997), Farrel and Barth (1998) and Vik (2000). More information on different coordinate frames can be found in most of the literature on inertial navigation.

## 2.2.3 Vector Transformations between rotating Coordinate Systems

The acceleration of a device with the position  $\underline{r}_l$ , moving with velocity  $\underline{v}_l$  and acceleration  $\underline{a}_l$  relative to a local frame or coordinate system denoted  $l$ , expressed in a global frame or coordinate system with the name  $g$  is given by Coriolis law (Williams, 1996):

$$\underline{a} = \frac{d^2}{dt^2} \underline{R}_0 + \underline{a}_l + 2\underline{\omega}_{gl} \times \underline{v}_l + \underline{\omega}_{gl} \times (\underline{\omega}_{gl} \times \underline{r}_l) + \dot{\underline{\omega}}_{gl} \times \underline{r}_l \quad (2.6)$$

The term  $\underline{a}$  in equation (2.6) is the acceleration and could be expressed in both the chosen global frame or the local frame,  $\underline{R}_0$  is the position of the origo of the local

coordinate system and  $\underline{\omega}_{gl} = p\underline{u} + q\underline{v} + r\underline{w}$  is the rotation of the coordinate system expressed in the local coordinate system.

The term  $\underline{\omega}_{gl} \times (\underline{\omega}_{gl} \times \underline{r}_l)$  is the centripetal acceleration sensed by the vehicle due to the rotation of the system, while the term  $2\underline{\omega}_{gl} \times \underline{v}_l$  is the experienced Coriolis acceleration. The last term on the right hand side is called the Euler term, and is due to angular accelerations.

The cross-product between the rotation vector and a vector can for small rotations be expressed as the product between a matrix  $\Omega$  and the vector  $\underline{v}$  as  $\underline{\omega}_r \times \underline{v} = \Omega_r \underline{v}$  (Titterton and Weston, 1997). The matrix  $\Omega_r$  is the *skew-symmetric* matrix of the rotation vector  $\underline{\omega}_r = [p, q, r]^T$  defined as

$$\Omega_r = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (2.7)$$

Equation (2.6) can then be rewritten as

$$\underline{a}_g = \frac{d^2}{dt^2} \underline{R}_0 + \underline{a}_l + \Omega_{gl} \underline{v}_l + \Omega_{gl} \Omega_{gl} \underline{r}_l + \dot{\Omega}_{gl} \underline{r}_l \quad (2.8)$$

Where the matrix  $\dot{\Omega}_{gl}$  here is defined as the skew-symmetric matrix formed by the vector  $\dot{\underline{\omega}}_{gl}$ .

The transformation of points and vectors between two coordinate systems is usually done in one of three different ways, Euler angles, direction-cosine matrices (DCM) or quaternions. Transformation with Euler angles is usually not chosen because of singularities which appear at some angles, described by Farrel and Barth (1998).

As discussed by Titterton and Weston (1997), using quaternions for coordinate-transformations may increase the numerical precision in the solution of the navigation equations, and the number of variables that have to be stored is four compared to nine with DCMs. On the other hand, matrices are easier to handle since the mathematical operations on matrices are built into most mathematical utilities as for instance MATLAB. The method with DCMs is chosen in this thesis for simplicity, and as mentioned by Titterton and Weston (1997), it is possible in practice to implement the algorithms with the same accuracy as with quaternions.

If the direction information is stored in a DCM, vectors are transformed from the body frame to chosen global frame like  $\underline{a}^g = D_l^g \underline{a}^l$  where  $D_l^g$  is the DCM from the local to the global coordinate system.

A DCM is defined by

$$D_1^g = \begin{bmatrix} \cos(\angle(\hat{x}_g, \hat{x}_1)) & \cos(\angle(\hat{y}_g, \hat{x}_1)) & \cos(\angle(\hat{z}_g, \hat{x}_1)) \\ \cos(\angle(\hat{x}_g, \hat{y}_1)) & \cos(\angle(\hat{y}_g, \hat{y}_1)) & \cos(\angle(\hat{z}_g, \hat{y}_1)) \\ \cos(\angle(\hat{x}_g, \hat{z}_1)) & \cos(\angle(\hat{y}_g, \hat{z}_1)) & \cos(\angle(\hat{z}_g, \hat{z}_1)) \end{bmatrix} \quad (2.9)$$

where notation  $\hat{x}_l$  here means that vector  $x_l$  is a unit vector of the local coordinate system. The orientation of the system is also needed, and is stored in the DCM. The heading of the three local axes expressed in global coordinates,  $\hat{x}$ ,  $\hat{y}$  and  $\hat{z}$ , can be found by using the equations:  $\hat{x} = C_l^g[1, 0, 0]^T$ ,  $\hat{y} = C_l^g[0, 1, 0]^T$  and  $\hat{z} = C_l^g[0, 0, 1]^T$ . Note that  $\hat{x}$  equals the first column vector in  $C_l^g$ ,  $\hat{y}$  the second and  $\hat{z}$  the third.

It is shown by Titterton and Weston (1997) that for a small angular rotation  $\underline{\omega}_{gl}$  the DCM propagates in time as

$$\dot{C}_l^g = C_l^g \Omega_{lg} \quad (2.10)$$

where the matrix  $\Omega_{lg}$  is the skew-symmetric matrix formed by  $\underline{\omega}_{gl}$ . This result is later used to calculate the DCM as a function of time.

## 2.2.4 Output from Accelerometers in a Gravity Field

Accelerometers that use a proof mass to measure acceleration can not separate between true acceleration and attraction due to gravity. The output of a complete IMU is a function of gravity and acceleration as

$$f = \underline{F}(t)/m = \ddot{r}(t) - \underline{G}(r) \quad (2.11)$$

This output is called *specific force*. To simplify expressions,

$\underline{g}(r) = \underline{G}(r) - \underline{\omega} \times (\underline{\omega} \times r)$  is often used in calculations, where  $\underline{g}$  is the local plumb-bob-gravity vector measured by a pendulum at rest.  $\underline{g}$  is in the direction that is normally said to be ‘down’, as shown in figure 2.4, while the vector  $\underline{G}(r) = -\gamma \frac{M}{|r|^3} r$  is the gravity field directed to the centre of the earth.

## 2.2.5 The Body Frame

In strap-down systems, see section 2.3.2, the body frame is the local coordinate system and follow the axes of the IMU, as shown in figure 2.2. The body frame follows the movement of the tracked point or vehicle. The three acceleration sensors, measures the specific force  $f_b = (f_w, f_v, f_w)$  along the axes  $\underline{u}$ ,  $\underline{v}$  and  $\underline{w}$  as given in figure 2.2. The gyros measure the rotational velocities  $(p, q, r)$  about the same axes. These axes are often called the roll-, pitch-, and yaw-axis.

Vectors in the body frame are given the subscript  $b$ .

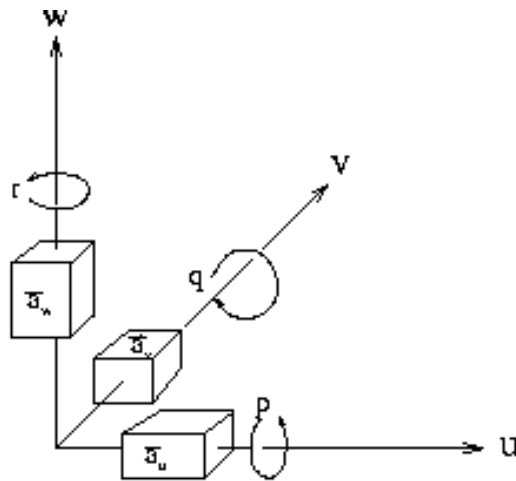


Figure 2.2: The Inertial Measurement Unit (IMU) as the body frame

As described in section 2.2.4, non-optical accelerometers measure what is called specific force,  $f_b$ , because they are unable to separate between acceleration and mass attraction due to gravity. The measurement equation in the body frame is then given by:

$$f_b = \ddot{r}_{ib} - \underline{G}(r_b) \quad (2.12)$$

$\underline{G}(r_b)$  is here the mass attraction due to gravity transformed to body frame coordinates, and  $\ddot{r}_{ib}$  is the acceleration relative to an inertial reference frame.

The rotation rates provided by the gyros are used to keep track of the heading of the axes, so that the acceleration measured in the local system can be from the local coordinates,  $(a_u, a_v, a_w)$ , to global coordinates.

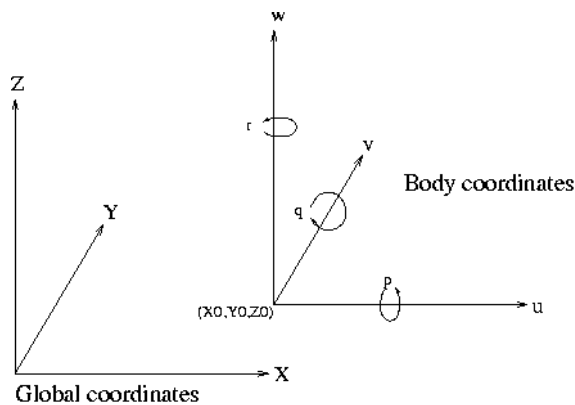


Figure 2.3: The global frame and the body frame

If the applied acceleration and rotation is measured in the same point in space, defined as origo in the body coordinate system,  $r_b = \dot{r}_b = \ddot{r}_b = 0$  such that equation (2.6)

gives  $\underline{a}_g(t) = \frac{d^2}{dt^2}R(t)$ . The measured local acceleration has to be transformed to the chosen global frame with a DCM which changes equation (2.12) to

$$\underline{a}_g = \underline{G}(r_g) + D_b^g f_b \quad (2.13)$$

In practical implementations of an IMU, there will always be a distance between the different sensors in the IMU such that  $r_b \neq 0$ , and thus there will be errors due to size-effects. If the position of the sensors within the IMU are known, the size-effects can be calculated and possibly compensated for.

## 2.2.6 The Earth-centred non-rotating Frame

This frame is called the inertial frame by Titterton and Weston (1997). The inertial frame has its origo fixed at the centre of the earth, with its axes fixed with respect to fix-stars. This system can be approximated to an inertial frame for periods much shorter than a year, although the frame rotates with respect to inertial space with a period of one year as the earth revolves around the sun. Vectors in the inertial frame are given the subscript 'i' in this thesis.

The inertial frame is by definition non-rotational, so that equation (2.13) can be applied directly

$$f_i = \underline{G}(r_i) + D_b^i f_b \quad (2.14)$$

One problem with this method is the difficulty in calculating the attraction due to mass attraction  $\underline{G}(r_i)$  (Farrel and Barth, 1998), see figure 2.4. The mass-attraction vector does only coincide with the plumb-bob gravity vector  $\underline{g}(r_i)$  at the poles. The plumb-bob gravity vector is the position a plumb-bob pendulum would point at a given position on the surface of the earth. A position expressed in inertial coordinates over a long time is not very useful, as the earth rotates around itself and around the sun, which could make a simple path on the surface of the earth very complicated in inertial coordinates.

A commonly used result often used in later derivations is velocity and acceleration relative to the surface of the earth expressed in inertial coordinates. Given equation (2.6) the velocity relative to a point on the earth expressed in the inertial frame is given by

$$v_i = D_e^i (v_e + \underline{\omega}_{ie} \times r_e) \quad (2.15)$$

Differentiating this equation with respect to time, and using equation (2.14) gives:

$$\underline{a}_i = D_e^i (f_e - \underline{\omega}_{ie}^e \times v_e - \underline{\omega}_{ie} \times \underline{\omega}_{ie} \times r_e + \underline{G}(r_e)) \quad (2.16)$$



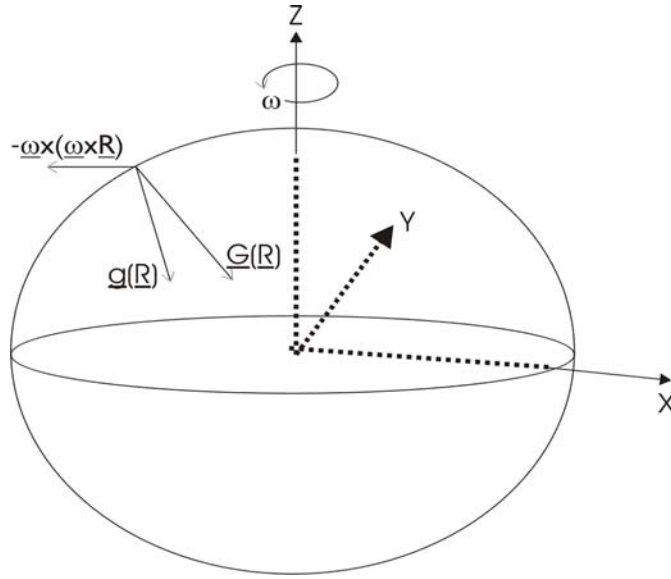


Figure 2.4: The plumb-bob gravity vector,  $\underline{g}$ , is the sum of the mass attraction and the centripetal acceleration

which when using the relationship

$$\underline{g}(r_e) = \underline{G}(r_e) - \underline{\omega}_{ie} \times (\underline{\omega}_{ie} \times r_e) \quad (2.17)$$

equation (2.16) can be written as

$$\underline{a}_i = \underline{f}_i - \underline{\omega}_{ie} \times \underline{v}_i + \underline{g}_i \quad (2.18)$$

The inertial frame is not commonly used in navigation implementations (Farrel and Barth, 1998), but the results are useful in following derivations.

### 2.2.7 Fixed Earth-based Frames

These frames are all fixed with respect to the earth. That means that they are rotating with respect to inertial space with a rotational velocity of approximately

$$\omega_{ie} \approx \frac{1 + 365,25}{365,25 \cdot 24 \cdot 3600s} = 7,2921 \cdot 10^{-5} s^{-1} \quad (2.19)$$

Using equation (2.6) combined with equation (2.13) and assuming that  $\omega_{ie}$  is constant, gives

$$\underline{a}_e = \underline{f}_e - 2\underline{\omega}_{ie}^e \times \underline{v}_e + \underline{G}(r_e) - \underline{\omega}_{ie}^e \times \underline{\omega}_{ie}^e \times r_e \quad (2.20)$$

Using equation (2.17) again gives the following navigation equation for earth-fixed frames.

$$\underline{a}_e = \underline{f}_e - 2\underline{\omega}_{ie}^e \times \underline{v}_e + \underline{g}(r_e) \quad (2.21)$$

One of these frames is the Earth Centred Earth Fixed frame (ECEF) which has its origin at the centre of the earth, and the x-axis at the Greenwich meridian. The z-axis is directed along the rotational axis of the earth, and the y-axis completes an orthogonal set of axes.

The shape of the earth can be approximated by an ellipsoid with semimajor axis  $a=6378137\text{m}$  and semiminor axis  $b=6356752\text{m}$  rotated around its semiminor axis.

Another earth based frame use the set of coordinates  $(\lambda, \varphi, h)$ , called geodetic coordinates.  $\lambda$  is the angle between the ellipsoid normal and the equatorial plane,  $\varphi$  is longitude and  $h$  is the height above the sea level.

The third earth-fixed coordinate system or frame mentioned here is what is chosen to call the local geographical frame which is described in the next section.

### 2.2.8 The Local Geographical Frame

This frame is called the geodetic frame by Farrel and Barth (1998). In this thesis vectors and points in the geodetic frame are given the subscript 'n'. This frame is also sometimes called the north-east-down-frame, the NED frame.

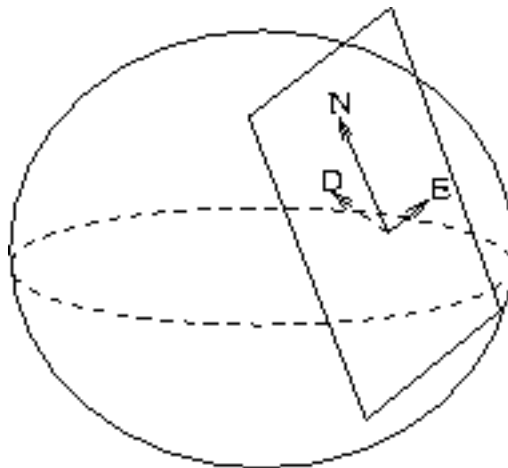


Figure 2.1: The local geographical frame, or the geodetical frame

This frame is defined at a fixed point on the surface of the earth. The axes point towards geographical north, east and along the local plumb-bob gravity vector  $\underline{g}(\underline{r})$  as shown on figure 2.1. This may be the best choice of coordinate system for the application considered in this thesis, as it is an intuitive description of the space in which this application will be used.

The direction of true north, the direction and value of the local gravity vector, the altitude above sea level and the latitude of the origin has to be known parameters.

#### Development of heading in the geodetic frame

A local geographical frame with latitude  $L$  rotates with respect to inertial space as

$\underline{\omega}_{in} = \omega_0([\cos(L), 0, -\sin(L)])^T$  where  $\omega_0 = 7292115 \times 10^{-5} \text{ rad/s}$ . This is the rotation of the earth expressed in the local frame as a component one component up, and one component north. The location of the origo does not move or rotate relative to the earth. The rotation of the body frame with respect to the geodetic frame equals the difference between the body frames rotation with respect to inertial space expressed in the body frame and the rotation of the geodetic frame with respect to the inertial frame expressed in body coordinates,  $\underline{\omega}_{nb} = \underline{\omega}_{ib} - C_b^in \underline{\omega}_{in}$ , which together with equation (2.10) gives (Titterton and Weston, 1997)

$$\dot{C}_b^n = C_b^n \Omega_{ib} - \Omega_{in} C_b^n \quad (2.22)$$

This equation is the last of the two navigation equations. The first is given in equation (2.21).

Equation (2.22) can be solved numerically by for instance using a fourth order Runge Kutta algorithm. Following Bortz (1971) Titterton and Weston (1997) suggests an alternative algorithm where the solution is split into two parts.

The solution to this equation is given in two parts, where the discrete solution of the first term is given by:

$$C_{k+1} = C_k A_k = C_k \exp \left( \int_{t_k}^{t_{k+1}} \Omega_{ib} dt \right) \quad (2.23)$$

The notation  $C_{k+1}$  means the matrix C at time  $t=t_0+k*dt$ . The solution to the second term is given by

$$\begin{aligned} C_{k+1} &= B_k C_k = \exp \left( \int_{t_k}^{t_{k+1}} -\Omega_{in} dt \right) C_k, \\ &= \left( I + \frac{\sin(|\theta|)}{|\theta|} [\underline{\theta}x] + \frac{(1 - \cos(|\theta|))}{|\theta|^2} [\underline{\theta}x]^2 \right) C_k \end{aligned} \quad (2.24)$$

where the rotation angle is defined by

$$\underline{\theta} = - \int_{t_k}^{t_{k+1}} \underline{\omega}_{in} dt \quad (2.25)$$

because the rotation is fixed in inertial space.

The solution of the first term is a bit more complex depending on the direction of the rotation of the body. If the rotation is fixed or approximately fixed in inertial space, the solution is very similar to equation (2.24), with A replaced by the expression for the matrix B. If this not is the case, rotation correction has to be implemented.

If direction of the rotation vector is approximated to be constant in space during the update interval, the term can be approximated to be

$$A_k = \exp\left(\int_{t_k}^{t_{k+1}} \Omega_{ib} dt\right) = \exp\left(\int_{t_k}^{t_{k+1}} [\underline{\sigma}x] dt\right) \quad (2.26)$$

where  $\sigma$  is the rotation vector with direction and magnitude such that a rotation about the vector with the magnitude of the vector would rotate the frame at  $k$  to the frame at  $k+1$ . The complete solution of equation (2.22) is given by

$$C_{k+1} = B_k C_k A_k = \left(I + \frac{\sin(\theta)}{\theta} [\underline{\theta}x] + \frac{(1 - \cos(\theta))}{\theta^2} [\underline{\theta}x]^2\right) C_k \cdot \exp\left(\int_{t_k}^{t_{k+1}} \Omega_{ib} dt\right) \quad (2.27)$$

where  $\theta$  is defined as in equation (2.25).  $B_k$  can be implemented at a slower computation rate than  $C_k$  and  $A_k$ , but as described in section 2.1.1, the computation powers available makes this unnecessary.

When not fixed in space, the rotation vector is given by the solution of

$$\dot{\underline{\sigma}} = \underline{\omega} + \frac{1}{2} \underline{\sigma} \times \underline{\omega} + \frac{1}{|\underline{\sigma}|^2} \left(1 - \frac{|\underline{\sigma}| \sin(|\underline{\sigma}|)}{2(1 - \cos(|\underline{\sigma}|))}\right) \underline{\sigma} \times \underline{\sigma} \times \underline{\omega} \quad (2.28)$$

which shown by Titterton and Weston (1997) can be approximated by the calculations

$$\underline{\alpha}(t) = \int_{t_k}^t \underline{\omega} dt, \quad \delta \underline{\alpha}(t) = \int_{t_k}^t \underline{\alpha} \times \underline{\omega} dt, \quad \underline{\sigma} = \underline{\alpha}(t_{k+1}) + \delta \underline{\alpha}(t_{k+1}) \quad (2.29)$$

### Accelerometer resolution and integration

The gathered data from the IMU can now be gathered and integrated as:

$$\underline{v}_n = \int_0^t \underline{f}_n dt - \int_0^t 2 \underline{\omega}_{in}^n \times \underline{v}_n dt + \int_0^t \underline{g} dt \quad (2.30)$$

where  $\int_{t_k}^{t_{k+1}} \underline{f}_n dt = \int_{t_k}^{t_{k+1}} C_b^n \underline{f}_b dt$ . The DCM at time  $k$  is denoted  $C_k$ . Using the results obtained in equation (2.27) the previous expression is now

$$\int_{t_k}^{t_{k+1}} \underline{f}_n dt = \int_{t_k}^{t_{k+1}} \left(I + \frac{\sin(\theta)}{\theta} [\underline{\theta}x] + \frac{(1 - \cos(\theta))}{\theta^2} [\underline{\theta}x]^2\right) C_k \exp(\underline{\sigma}x) \underline{f}_b dt \quad (2.31)$$

This can be implemented in for instance MATLAB and solved numerically. The Coriolis term can be solved by

$$-\int_0^t 2\omega_{in}^n \times v_n dt = -2\int_0^t \Omega_{in}^n v_n dt \quad (2.32)$$

$$= -2\omega_0 \int_0^t [\sin(L) \cdot v_y, -\sin(L) \cdot v_x + \cos(L) \cdot v_z, \cos(L) \cdot v_y]^T$$

$$v_n(t_{k+1}) = -2\omega_0 \int_{t(k)}^{t(k+1)} \Omega_{in}^n v_n(t_k) dt \quad (2.33)$$

The position can then be found by integrating the velocity,

$$r_n(t_k) = \int_{t_k}^{t_{k+1}} v^n dt \quad (2.34)$$

These are all the basis equations needed to process the data from the IMU and calculate the wanted information.

## 2.3 Implementations of an Inertial System

The main component in an inertial system is the Inertial Measurement Unit (IMU), which measures the applied acceleration, and in addition the applied rotation if a strap-down solution is chosen. There are two different ways to implement an IMU. The first and oldest kind of systems are called platform systems. Technological development and ever increasing computational power has pushed the development towards strap-down systems eliminating the complex mechanical structures of the platform system. A description of Platform-based Systems is given in the next section for completion, but the strap-down technology is chosen as a basis in this thesis.

### 2.3.1 Platform-based Systems

In a platform system, the whole system is mounted on a structure with gimbal rings. A gyro is used to keep the direction of three accelerometers fixed in inertial space. The measured acceleration can then be measured directly to find the velocity and the position of the platform. This kind of systems requires expensive and complex mechanical structures with bearings, slip rings and torque motors.

An additional system is needed to keep track of the heading of the vehicle, while a strap-down system can keep track of both heading and position.

Some of the advantages of these systems are that they can self align by the use of gyro compassing, and that they tolerate high vehicle turning rates, greater than 1000 deg/s (Lawrence, 1998).

The accuracy of platform systems can be made better than the accuracy of strap-down systems, and will also be the choice in the future when high accuracy is needed (Barbour and Schmidt, 1998). The systems used on nuclear submarines are reported to drift as low as one nautical mile in 24 hours (Titterton and Weston, 1997).

### 2.3.2 Strap-down Systems

The inertial system in a strap-down solution follows the movement of the vehicle. The IMU now has to measure the acceleration along three axes, and the rotation rates around the same axes.

The IMU defines a local coordinate system with three axes, usually named  $(u, v, w)$ . The measured acceleration vector has to be transformed to the chosen coordinate system in order to integrate the vector. This means that the heading of the IMU has to be stored, usually in a DCM or with quaternions, see section 2.2.3.

The strap-down system has the advantage that it can be constructed from relatively cheap and easily available MEMS devices. It is also smaller and have lower power consumption than platform systems. The need for more computational power is not usually a problem in applications without real-time-processing, because the collected data can be stored and processed after the experiment.

## 2.4 Accelerometer Characterization

The aim is to obtain models of the accelerometers, given the system specifications in section 2.1.1. The following subsection contains some different accelerometer models and a short explanation of the terms in the equations.

In addition to obtaining values of the parameters in the model equations, one of the goals is to look at variations in the parameters, like run-to-run differences and long time variations. Any correlations between temperature and changes in accelerometer parameters are also important. If the changes are repeatable, they can be measured and compensated for, thus avoiding the need for any external temperature control system.

### 2.4.1 Accelerometer Models

A common way to model the accelerometer output in acceleration units as function of the input acceleration or specific force is as follows

$$E = K_1(k + K_0 + a_i + K_2 a_i^2 + K_3 a_i^3 + K_{ip} a_i a_p + K_{io} a_i a_o + \varepsilon) \quad (2.35)$$

$E$  is the accelerometer output, usually with units Volts.  $a_i$  is the input acceleration, or more correctly the input specific force defined by equation (2.11). The units of the input is g or  $m/s^2$ .  $K_1$  is the scale factor of the accelerometer and has units Volt/g or Volt/( $m/s^2$ ). The term  $K_0$  is called bias and is defined as accelerometer output not is correlated with input. The bias has units g or  $m/s^2$ .  $K_2$  and  $K_3$  are higher order nonlinear parameters and have units  $V/(m/s^2)^2$  and  $V/(m/s^2)^3$ . The coefficients  $K_{ip}$  and

$K_{io}$  are the sensitivities of the accelerometer for accelerations normal to the input axis.  $K_{ip}$  is the sensitivity to accelerations along the principle axis, and  $K_{io}$  is the sensitivity to accelerations along the output axis. They both have units  $V/(m/s^2)^2$ . The input-, output- and principle-axis are defined as a right hand coordinate system as shown in figure 2.1. The directions of the principle- and the output- axis relative to the input axis are determined by the mechanical structure of the accelerometer.  $\varepsilon$  is unmodelled errors and accelerometer noise and has units  $m/s^2$ .

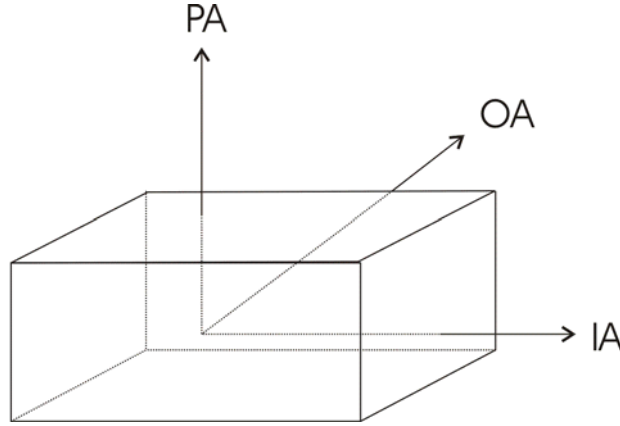


Figure 2.1: The axes of an accelerometer

The complete IEEE accelerometer model equation from the IEEE linear accelerometer testing standard (IEEE Std 1293-1999) is listed in appendix D. Several terms in this equation are often omitted in the modelling, and will thus bias the other terms. For instance Titterton and Weston (1997) use the model

$$\tilde{a}_x = (1 + S_x)a_x + M_y a_y + M_z a_z + B_f a_x a_y + n_x \quad (2.36)$$

where  $S_x$  is scale factor error,  $M_y$  and  $M_z$  are called cross-axis coupling factors which are named misalignment in the IEEE standard (IEEE Std 1293-1999).  $B_f$  are called vibro-pendulous error coefficient while  $n_x$  is random bias.

The model may also be adapted to the accelerometer which is to be tested. For instance a capacitive open-loop micromachined capacitive accelerometer is said to have mainly nonlinearities of even order (IEEE Std 1293-1999). Thus  $K_3$  could be estimated to be zero. Knowledge about the accelerometer prior to testing could ease the modelling process.

In this thesis, a modified version of the IEEE model equation in appendix D is used. It was chosen to model the electric output as function of input acceleration, instead of estimated acceleration as a function of accelerometer input. To achieve this, the scale factor is multiplied with the coefficients inside the delimiters on the right hand side of equation (2.35). The new equation is then defined by

$$E = O + K a_i + K_2' a_i^2 + K_3' a_i^3 + K_{ip}' a_i a_p + K_{io}' a_i a_o + \varepsilon' \quad (2.37)$$

where the new coefficients are defined by  $O = K_1K_0$ ,  $K = K_1$ ,  $K_2' = K_1K_2$ ,  $K_3' = K_1K_3$ ,  $K_{ip}' = K_1K_{ip}$ ,  $K_{io}' = K_1K_{io}$  and  $\varepsilon' = K_1\varepsilon$ . The parameter  $O$  now has units Volts and is called the accelerometer offset. The scale factor has the same units, and is still called the scale factor. The rest of the coefficients changes units, but they are called the same as the corresponding coefficients in the original IEEE model equation. The complete modified equation with parameter names and units is also listed in appendix D. Only this model will be used in the rest of this thesis, and the mark on the symbols will not be used. The equation will be written as follows in the rest of the thesis.

$$E = O + Ka_i + K_2a_i^2 + K_3a_i^3 + K_{ip}a_i a_p + K_{io}a_i a_o + \varepsilon \quad (2.38)$$

Other terms from the complete modified equation listed in appendix D may be added to the equation as well.

To model the accelerometer output directly in electrical units simplifies the estimation process, as the offset can be measured directly. The estimation of the scale factor is also eased in this way, as there is no need to use a nominal scale factor and a scale factor error as described in the IEEE standard (IEEE Std 1293-1999).

## 2.4.2 Noise Modelling

The error term  $\varepsilon$  in equation (2.38) should also be modelled in order to use the model of the accelerometer in an inertial system. As given in the IEEE accelerometer testing standard, having correct noise models in the guidance and navigation model leads to better performance for the calibration, alignment and guidance Kalman filters.

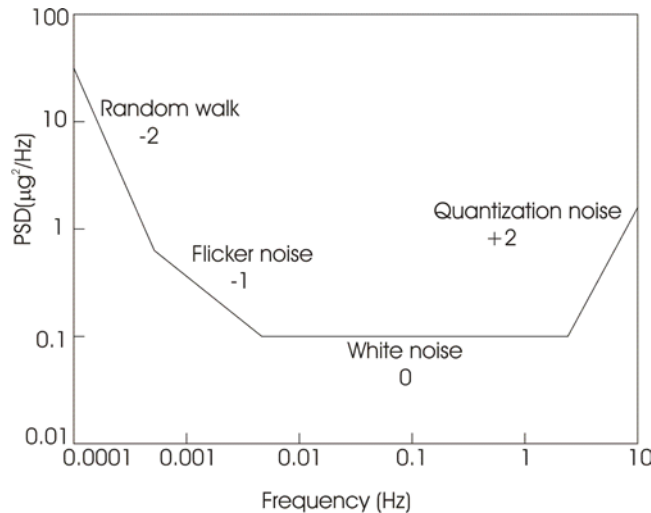


Figure 2.2: Noise model of an accelerometer

The characteristic noise from an accelerometer run through a power spectral density analysis looks somewhat like the plot on figure 2.2.

The white noise given in the figure is the white process noise, and has by definition constant PSD over all frequencies and thus a 0 slope on the log-log plot. The



quantification noise is due to discretization during measurement of the signal, and can be moved to the right by increasing the sampling frequency and removed by applying analog and digital filtering. The flicker noise and random walk is due to temperature or other environmental changes and long time changes in the instrument because of stress, aging and so on.

### Selected Noise Models

Several different Gauss-Markov processes useful for modelling errors are listed in the following subsections. The models are taken from Gelb (1974).

- Random constants are described by the differential equation

$$\dot{x} = 0 \quad (2.39)$$

- Random walk is characterised by

$$\dot{x} = w \quad (2.40)$$

where  $w$  is zero mean white Gaussian noise.

- Random Ramp. Signals that exhibit a time-growing behaviour can be approximated by a random ramp. A random ramp is characterized by the equations.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 0 \end{aligned} \quad (2.41)$$

- First-order Gauss-Markov processes. A random quantity with the autocorrelation function

$$\varphi_{xx} = \sigma^2 e^{-\beta|\tau|} \quad (2.42)$$

is called a first-order Gauss-Markov process, and is often used to describe a band-limited signal, whose spectral density is flat over a finite bandwidth (Gelb, 1974). The differential equation for the state variable is

$$\dot{x} = -\beta x + w \quad (2.43)$$

- Periodic random quantities. Variables with periodic behaviour can have an autocorrelation function which looks like

$$\varphi_{xx} = \sigma^2 e^{-\beta|\tau|} \cos(\omega|\tau|) \quad (2.44)$$

These systems are represented by differential equation system

$$\begin{aligned} \dot{x}_1 &= x_2 + w \\ \dot{x}_2 &= -\alpha^2 x_1 - 2\beta x_2 + (\alpha - 2\beta)w \\ \alpha &= (\beta^2 + \omega^2)^{\frac{1}{2}} \end{aligned} \quad (2.45)$$

The spectral density of the white noise  $w$  is given by  $2\beta\sigma^2\delta(\tau)$ .

### **Empirical noise modelling**

The method suggested by Gelb (1974), is to implement an ideal smoother, calculate the initial condition, and adjust the noise models in order to minimize the calculated residuals.

Gelb (1974) suggests the calculated autocorrelation of the noise signal as an initial guess. Several different techniques are listed to obtain the autocorrelation of a sampled signal.

The IEEE document (IEEE Std 1293-1999) describes a method based on calculating the PSD of the acquired signal, and how to identify different noise models from the power spectrum. It also describes a second method using Allan variance.

The third method mentioned in the document is model fitting via the Kalman filter. Instead of minimizing the residuals, a method using the maximum likelihood function is briefly described.

### **Bias model**

Bar-Shalom et al. (2001) operates with the following way to model the accelerometer bias. The accelerometer bias  $\beta$  is here modelled as the sum of two distinct Markov processes  $\beta_1$  and  $\beta_2$  with exponential autocorrelations and with different time constants.

$$\begin{aligned}\dot{\beta}_1 &= \frac{-\beta_1}{\tau_1} + \tilde{v}_1 \\ \dot{\beta}_2 &= \frac{-\beta_2}{\tau_2} + \tilde{v}_2\end{aligned}\tag{2.46}$$

Which of these models which are chosen has to be considered later after experiments on real components and simulations.

## **2.5 Estimation and Smoothing**

### **2.5.1 State Space Models**

The state of a continuous system is defined by Farrel and Barth (1998) as the smallest set of numbers at a known time  $t_0$  that together with knowledge of the input of the system input for  $t \geq t_0$ , is sufficient to determine the system response for all  $t > t_0$ . If the state  $\underline{x} \in X$  is the state of the system, then  $X$  is referred to as the state-space of the system.

The system is expressed by a number of differential equations. A system described in state-space form is described in the time domain instead of the frequency plane. This is an advantage because a lot of statistical features can be added to the system.

A general linear system on state space form is given by

$$\dot{\underline{x}} = A(t)\underline{x} + B(t)\underline{u} \quad (2.47)$$

A general system which may not necessarily be linear on state space form is given by:

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) \quad (2.48)$$

For instance the equation for the NED frame defined in the equations (2.21) and (2.22) described on this form is given by equation (2.49).

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{v}} \\ \dot{C} \end{bmatrix} = \begin{bmatrix} \underline{v} \\ C\underline{f} - 2(\underline{\omega} \times \underline{v}) - \underline{g} \\ C\dot{\underline{\Omega}} \end{bmatrix} \quad (2.49)$$

## 2.5.2 Estimation and Kalman Filters

Estimation is the process of getting the best possible estimate of the system state from the measurement data. If the process noise and measurement noise is *Gaussian* and the measurement is linear, the Kalman filter can be shown to be the optimal estimator (Gelb, 1974).

A linear continuous-discrete dynamic system can be described as equation (2.50), where  $\underline{x}(t)$  is the state vector,  $\underline{u}(t)$  is the system input and  $\underline{v}(t)$  is the process noise.  $H_k$  is the measurement matrix at time  $k$ , while  $\underline{w}$  is the measurement noise of the system.  $z_k$  is the measurement update to the filter. The noise vectors are defined by their autocorrelations as shown in equation (2.50).  $\tilde{Q}(t)$  is the spectral density matrix of the process noise, and  $R_k$  is the covariance matrix of the measurement noise.

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + L(t)\underline{u}(t) + G(t)\underline{v}(t) \quad (2.50)$$

$$z_k = H_k \underline{x}_k + \underline{w}_k$$

$$E\{\underline{x}(t_0)\} = \bar{\underline{x}}_0, \quad E\left\{(\bar{\underline{x}}(t_0) - \bar{\underline{x}}_0)(\bar{\underline{x}}(t_0) - \bar{\underline{x}}_0)^T\right\} = \bar{P}_0$$

$$E\{\bar{\underline{x}}(t_0)\underline{v}^T(t)\} = \underline{0}, \quad E\{\underline{v}(t)\} = \underline{0}, \quad E\{\underline{v}(t)\underline{v}^T(\tau)\} = \delta(t - \tau)\tilde{Q}(t)$$

$$E\{\bar{\underline{x}}(t_0)\underline{w}_k^T\} = \underline{0}, \quad E\{\underline{w}_k\} = \underline{0}, \quad E\{\underline{w}_k\underline{w}_l^T\} = \delta_{kl}R_k, \quad E\{\underline{v}(t)\underline{w}_k^T\} = \underline{0}$$

The Kalman filter solution of this dynamic system is given by equation (2.51).  $\hat{\underline{x}}_k$  is the estimated state after a measurement update, while  $\bar{\underline{x}}_k$  is the predicted value of the state vector.  $\hat{P}_k$  is the state covariance matrix after a measure update, while  $\bar{P}_k$  is the predicted covariance matrix.

$$\begin{aligned}
\dot{\bar{\underline{x}}}(t) &= F(t)\bar{\underline{x}}_k + L(t)\underline{u}(t), \bar{\underline{x}}(t_k) = \hat{\underline{x}}_k \\
\dot{\bar{P}}(t) &= F(t)\bar{P}(t) + P(t)F^T(t) + G(t)\tilde{Q}(t)G^T(t), P(t_k) = P_k \\
\hat{\underline{x}}_k &= \bar{\underline{x}}_k + K_k(\underline{z}_k - H_k\bar{\underline{x}}_k), \bar{\underline{x}}_k = \bar{\underline{x}}(t_k) \\
K_k &= \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1}, \bar{P}_k = \bar{P}(t_k) \\
\hat{P}_k &= (I - K_k H_k) \bar{P}_k
\end{aligned} \tag{2.51}$$

The discrete form of equation (2.50) is given by the following system.

$$\begin{aligned}
\underline{x}_{k+1} &= \Phi_k \underline{x}_k + \Lambda_k \underline{u}_k + \Gamma_k \underline{v}_k \\
\underline{z}_k &= H_k \underline{x}_k + \underline{w}_k \\
E(\underline{v}_k \underline{v}_l) &= \delta_{kl} Q_k
\end{aligned} \tag{2.52}$$

The matrix  $Q_k$  is now the covariance matrix of the process noise. The rest of the correlation expressions are defined as in equation (2.50). The Kalman filter solution to the discrete state equation is given by

$$\begin{aligned}
\bar{\underline{x}}_{k+1} &= \Phi_k \bar{\underline{x}}_k + \Lambda_k \underline{u}_k \\
\bar{P}_{k+1} &= \Phi_k \hat{P}_k \Phi_k^T + \Gamma_k Q_k \Gamma_k^T \\
\hat{\underline{x}}_k &= \bar{\underline{x}}_k + K_k(\underline{z}_k - H_k \bar{\underline{x}}_k) \\
K_k &= \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1} \\
\hat{P}_k &= (I - K_k H_k) \bar{P}_k
\end{aligned} \tag{2.53}$$

### 2.5.3 Linear Fixed Interval Smoothing

The Rauch-Tung-Striebel formulation of the optimal smoother will be used in this thesis. The backward filter for a time invariant system is given by (Gelb, 1974)

$$\begin{aligned}
\hat{\underline{x}}_{k|N} &= \hat{\underline{x}}_k(f) + A_k[\hat{\underline{x}}_{k+1|N} - \hat{\underline{x}}_{k+1}(f)] \\
A_k &= P_k(f) F^T P_{k+1}(b) \\
\hat{\underline{x}}_{N|N} &= \hat{\underline{x}}_N(f) \text{ for } k = N-1 \\
P_{k|N} &= P_k(f) + A_k[P_{k+1|N} - P_{k+1}(b)]A_k^T \\
P_{N|N} &= P_N(f) \text{ for } k = N-1
\end{aligned} \tag{2.54}$$

The smoothed estimates of the state vector and the covariance matrix are given by  $\hat{\underline{x}}_{k|N}$  and  $P_{k|N}$ , the forward estimate obtained with the linear Kalman filter from equation (2.53) is given by  $\hat{\underline{x}}_k(f)$

#### 2.5.4 Least Squares Estimation

Using linear least squares estimation (Gelb, 1974), the goal is to minimize the sum of the square of the deviations from the model of the process, or minimize

$$J(\underline{\alpha}) = (\underline{z} - H\underline{x})^T(\underline{z} - H\underline{x}) \quad (2.55)$$

If the measurements have different standard deviations, the least squares estimation can be implemented with a weighting matrix or

$$J(\underline{\alpha}) = (\underline{z} - H\underline{x})^T R^{-1}(\underline{z} - H\underline{x}) \quad (2.56)$$

where  $R$  is a symmetric positive definite weighting matrix.  $\underline{z}$  is the vector of measurements,  $H$  is the system model matrix and  $\underline{x}$  is the vector of inputs. If  $R$  is a diagonal matrix with the measurement standard deviation  $w_j$  on the diagonal, the expression can be reformulated as

$$J(k) = \sum_j \frac{(z_j - Hx_j)^2}{w_j^2} \quad (2.57)$$

In order to find the optimal estimation of the parameter vector  $\underline{k}$  we use

$$\frac{\partial}{\partial \alpha_i} J(\underline{k}) = 0 \quad (2.58)$$

It can be shown (IEEE Std 1293-1999) that least squares estimation is a maximum likelihood estimator if the measurement errors have Gaussian probability distributions. These equations will be used later to estimate parameters from static multi-point tumble tests, see section 3.5.

As given in the IEEE report, the Fisher information matrix in the case of linear least squares estimation is given by the normal equations

$$A_{ik} = \sum_j \frac{\partial}{\partial \alpha_i} f(\alpha, \theta_j) \frac{\partial}{\partial \alpha_k} f(\alpha, \theta_k) \Big|_{\alpha_0} \quad (2.59)$$

or in the case with weighted measurement points:

$$A_{ik} = \sum_j \frac{1}{w_j^2} \frac{\partial}{\partial \alpha_i} (f(\alpha, \theta_j) \frac{\partial}{\partial \alpha_k} f(\alpha, \theta_k) \Big|_{\alpha_0}) \quad (2.60)$$

where the parameter vector is given by alpha. The covariance of the parameters is then given by this matrix inverted times the covariance of the measurement residuals. The residuals are given by:

$$rms(res) = \sqrt{\frac{1}{n} \sum_j \frac{[A_j - f(\theta_j, \hat{\alpha})]^2}{w_j^2}} \quad (2.61)$$

In the case of uniform measurement accuracy,  $w_j=1$ .

$$\sigma_{ik} \geq (A^{-1})_{ik} \cdot rms(res) \quad (2.62)$$

or and the standard deviation of a parameter is given by

$$\sigma_k = \sqrt{\sigma_{kk}} \quad (2.63)$$

The correlation between two parameter estimates is given by

$$\varepsilon_{ik} = \frac{\sigma_{ik}}{\sigma_i \sigma_k} \quad (2.64)$$

As mentioned in the IEEE report, because of small sample times and unmodelled effects the real error is better obtained from the repeatability between several tests.

### 2.5.5 The Savitzky-Golay filter

The Savitzky-Golay filter was invented by A. Savitzky and M. Golay and published in 1964 (Savitzky and Golay, 1964). It's main purposes are smoothing and differentiation of data with significant noise. In this application the filter is used to calculate the second derivative of the position data to obtain the acceleration, and to smooth the data from the accelerometer.

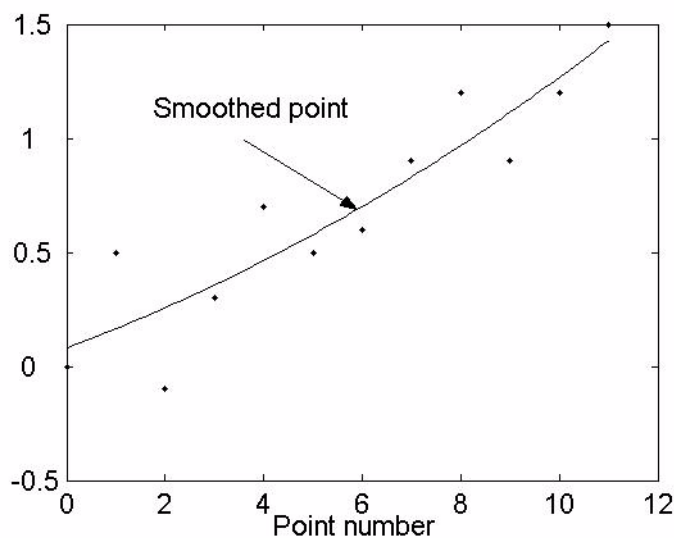


Figure 2.3: Savitzky-Golay smoothing of random function,  $n_l=6$ ,  $n_r=5$

The operating principle of the filter is to use the method of least squares making a polynomial fit around each point on the curve. The polynomial can then be differentiated to obtain the  $n$ 'th differentiated of the curve. The number of sampling points to be included in the smoothing is given as a parameter to the filter.

The number of points to be included in the smoothing to the left of the points is denoted  $n_l$ , while the number of points to the right is  $n_r$ .

### Numerical implementation

The implemented algorithm is based on an algorithm from 'Numerical recipes in C' (Flannery et al., 1992) in which the normalization constants are implemented somewhat differently than in the original article. During the implementation of the algorithm it was shown that a normalization constant had to be added in order to obtain the correct amplitude on the differentiated signal,

$$N = \frac{(-1)^d \Delta^d}{d!} \quad (2.65)$$

where  $d$  is the order of differentiation and  $\Delta$  is the sample period of the data. For smoothing,  $d=0$ ,  $N$  is set to one. The filter is implemented in the MATLAB function `savgol()` listed in appendix A.

### The amplitude response of the filter

The filter was applied to a number of generated data patterns in order to check the filter response for different curves. As seen from figure 2.4 the filter differentiates a second order curve to a very high degree of accuracy. The noise seen in the figure is probably due to round-off errors, and has a standard deviation of  $2.5756e-012$ . The amplitude response of the filter did not seem to be influenced much by the sampling period or the number of smoothing points in this case.

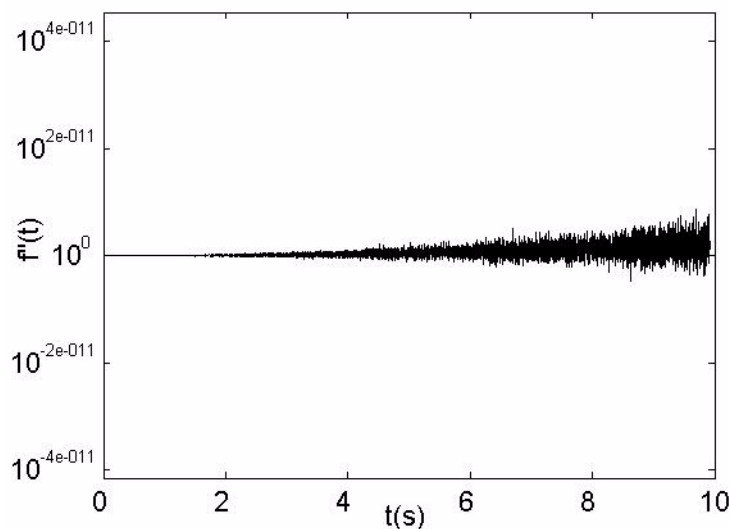


Figure 2.4: Differentiation of  $0.5t^2$  using a second degree polynomial fit

The accuracy of the filter is seen to decrease when the curve which is to be smoothed is different from a polynomial of degree 2.

Testing and simulations showed us that the amplitude response change as a function of the number of sampling points, thus making the selection of number of points important. It also changes as a function of sampling time. This means that the amplitude response has to be calculated for each sampling time, and number of sampling points.



## 3 Materials and Methods

### 3.1 The Accelerometers

#### 3.1.1 ADXL105

This accelerometer is a one-axis device manufactured by Analog Devices. It is a capacitive MEMS accelerometer, and is packed in a surface mounted, ceramic leaded capsule. The package has to be mounted on some kind of circuit board to obtain electrical connection.

It has a built in temperature sensor that can be used for calibration purposes.

Some information from the ADXL105 datasheets is listed in appendix C.

Five individual units of this type were available for testing, and they were numbered one to five and will be referred to by their number in this thesis.

#### Definition of axes

In chapter 2.4 a definition of the three axes of the accelerometer was described. The choice of these axes should probably be based on knowledge about the mechanical structure of the accelerometer. As no such information was available from Analog Devices, the output and principle axes were chosen as shown on figure 3.1.

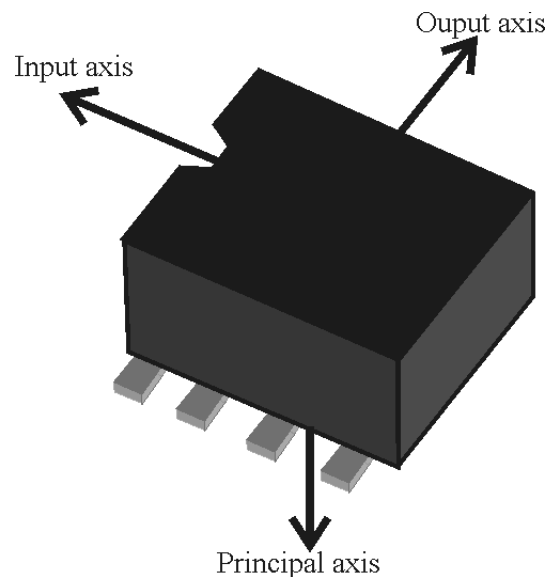


Figure (3.1): Definition of axes for the ADXL105

#### Ratiometric test

As given by datasheets the accelerometers are ratiometric, and as mentioned by Lawrence (1998), one of the main challenges in inertial navigation systems was to find a power supply that was stable and able to withstand radioactive radiation.

In order to determine the demands on the power supply, one simple test was conducted with an ADXL105 unit. The supply voltage VDD was varied around the nominal 5V, with the accelerometer in a fixed static position, and the outputs from the accelerometer and the temperature sensor were logged. The results are shown in figure 3.2.

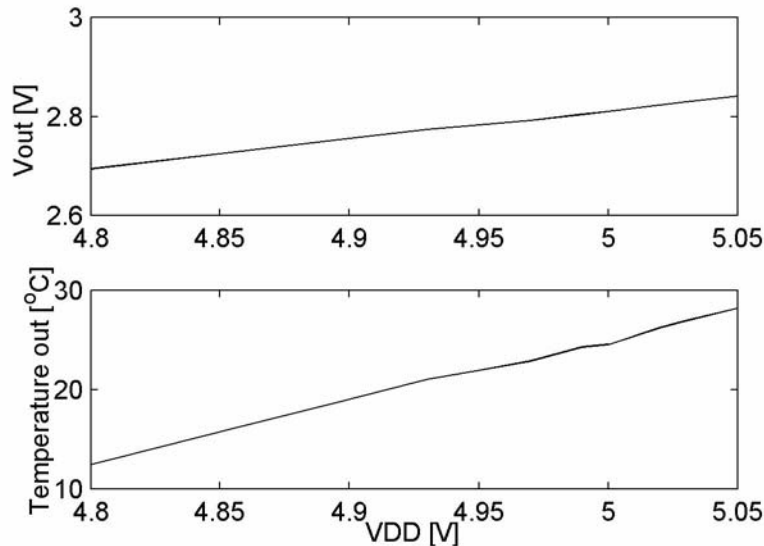


Figure (3.2): Sensor and temperature output as function of VDD for a fixed static acceleration.

The output voltage as a function of VDD varied as 0.5 V/V, meaning that a variation of 10 mV in the supply voltage would give a 5 mV offset. Calculated with the nominal ADXL105 scale factor, 0.25 V/g, this would induce a bias of 2 mg. The corresponding dependence between VDD and the temperature output was 62 °C/V which gives that a 10 mV change induces a 0.62 °C change in the temperature output. As a conclusion, the long time variations in the power supply should be in the order of 1 mV or lower.

### The Linear Power Supply (LT1086)

The LT1086-CT5 was chosen to supply VDD=5 V to the accelerometers. It is packed in a TO220 package. The specifications in the datasheets specifies an output RMS noise of 0.003% or 0.15 mV. A test run over a 24 hours period, where the output was logged with a 16 bits ADC, showed that the output voltage was stable within 1 mV, as seen in figure 3.3. The data has been filtered with an equally weighted running average digital filter to obtain the long time trends.

### 3.1.2 Testing the Built-in Temperature Sensor

The built-in temperature sensor in the ADXL105 accelerometer is specified to be ‘optimized for repeatability rather than accuracy’ in the datasheets and can have an systematic error relative to the real temperature up to

$\pm 100\text{mV}/8\text{mV}/^{\circ}\text{C} = \pm 12.5^{\circ}\text{C}$ . In addition there may be an error in the nominal temperature sensor scale factor of 8 mV/°C.

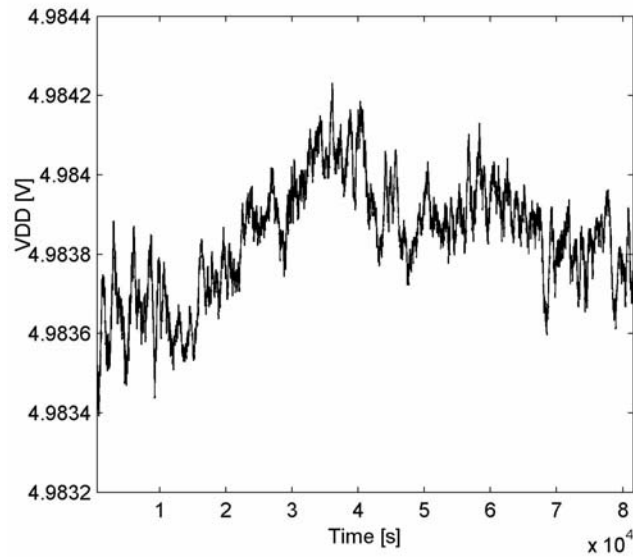


Figure (3.3): 24 Hours test of the output from the LT1086-5 linear regulator

The output from the sensor was logged for a period of 24 hours, together with the output from a AD592AN temperature transducer, which was considered to be a correct reference, at least regarding linearity and scale factor,  $1\mu\text{V}/^\circ\text{K}$ . The experiment was done with the ADXL105 unit number 3. A running average filter has been applied to both the curves to eliminate noise.

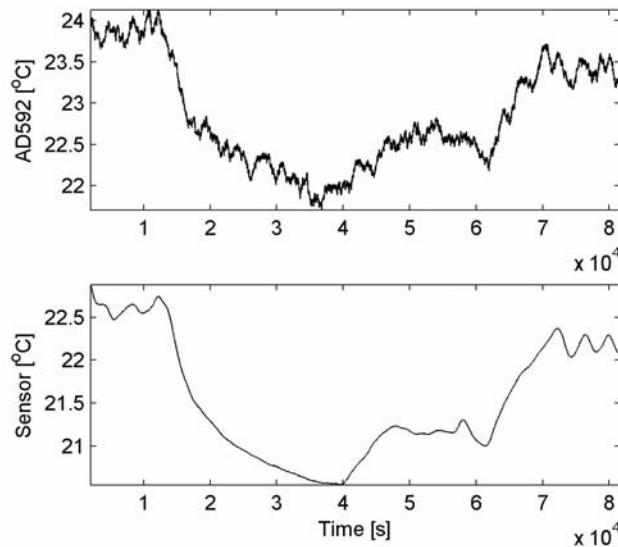


Figure (3.4): Output from the AD592 temperature transducer and an ADXL05 temperature sensor at room temperature during 24 hours period with nominal scale factor for the ADXL105 temperature sensor.

The difference between the two sensors was not constant for the whole period, indicating an error in the nominal scale factor. A scale factor of  $9 \text{ mV}/^\circ\text{C}$  gave an approximately constant difference of  $0.94 \text{ }^\circ\text{C}$  with a standard deviation of  $0.09 \text{ }^\circ\text{C}$ . The experiments showed that both the offset between the measured and the real temperature and the temperature scale factor had to be calibrated for each individual accelerometer.

The  $0.1 \text{ }^\circ\text{C}$  standard deviation could be caused by short term variations in one of the two sensors, or in both of them, and possibly by noise from the signal transmission. The AD592 datasheet specifies a repeatability of  $0.1 \text{ }^\circ\text{C}$  after a heating cycle from  $-25 \text{ }^\circ\text{C}$  to  $125 \text{ }^\circ\text{C}$  to the next cycle, but does not say anything about neither the repeatability for relatively small changes nor possible errors in the scale factor.

### 3.1.3 Mounting the Accelerometer

During all of the experiments accelerometer was soldered to Sunhyato ICB-010 SO-carrier circuit card to obtain electrical connection to the components. Two holes were drilled in the card in order to attach it to any surface, using two screws as shown on figure 3.5. The note ‘Mounting Considerations for ADXL Series Accelerometers’ by Shuster et al., recommends using a small drop of epoxy to attach the seating plane of the accelerometer’s package to the board in order to avoid resonance at high frequencies. The movements performed during the experiments described in this thesis were in a frequency area way below the possible resonance frequency, making the epoxy superfluous.

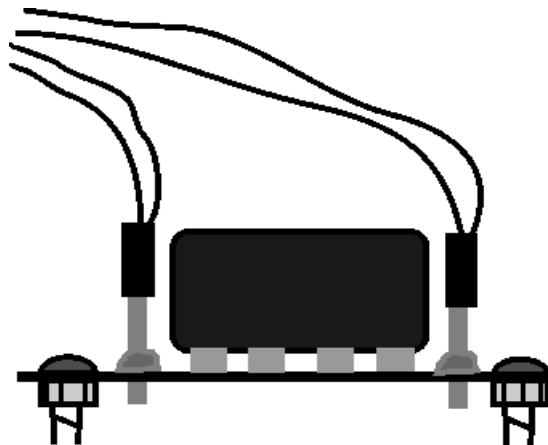


Figure (3.5): Schematic drawing of the mounting of the accelerometer mounted on the chip-carrier board

During the dynamic experiments described in section 3.6 and during some of the multi-point tumble tests about two axes, see section 3.5, the accelerometer was placed inside a small aluminium box. The box was kept at ground potential and thus acting like a Faraday-cage to shield the system from electrical noise. It also served as a part of the temperature control systems described in the following subsection.

The accelerometer was attached to the bottom of the box using two plastic screws. The cable running from the accelerometer to the DAQ system went through a PG nipple. The

purpose of the PG nipple was to keep the cables inside the box in a fixed position, and to minimize vibrations transferring from the cable to the accelerometer.

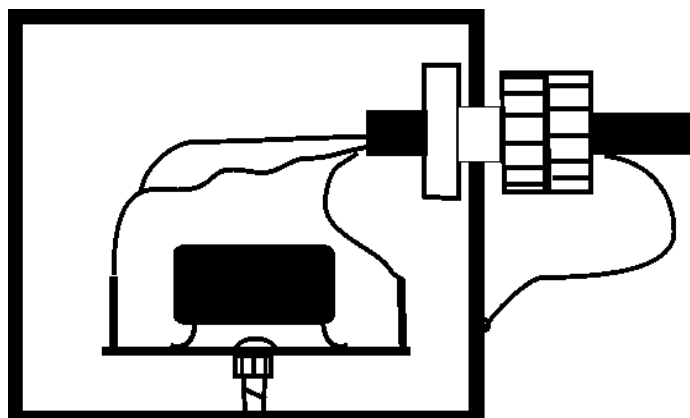


Figure (3.6): Cross section of the mounting of the accelerometer inside the aluminium box

Heat leading pasta was applied between the lid and the rest of the box in order to ensure thermal conductivity and minimize temperature gradients.

### 3.1.4 Crossbow CLX02TG3

The Crossbow CLX02TG3 is a high-accuracy three axis accelerometer. It has a built in temperature sensor, and is packed in a aluminium box with dimensions (2.49cm x 5.68cm x 3.65cm).

It has a defined set of input axes  $x$ ,  $y$  and  $z$  which was a natural basis of axis for the multi-position tumble tests in section 3.5.

This accelerometer was out of the price range for the accelerometers considered in this thesis, about 2000 US dollars pr august 2002. The main purpose of the tests done with the Crossbow unit, was to test and verify the experimental setup and methods.

Some information from the datasheets is listed in appendix C.

## 3.2 Temperature Control Systems

### 3.2.1 Heat Boxes

Two different heat boxes were used for temperature control. One of them, produced by Binder, had heating capability, while the other which was produced by Weiss Technik had both cooling and heating functionality. The specifications of both the heat boxes promised a long term stability of better than  $0.1\text{ }^{\circ}\text{C}$ . Both of the boxes had fans to circulate air inside the box, which had to be switched off during measurements because of the vibration generated by the rotating fan.

The heat box without cooling functionality was used to heat up the accelerometer to a relatively high temperature. The power was then turned off, letting the system cool

down. The accelerometer was packed in aluminium foil to slow down the cooling process. Continuously monitoring the temperature of the accelerometer, the output was logged when the system reached the selected temperatures. The cooling was considered to be slow enough so that the temperature was constant during the measurement period, ~15s.

As opening the door of the box was impossible without influencing the temperature, a cooling series was run for each position in the multi position tumble test in section 3.5. The slowly decreasing temperature caused a large time span between the first and last measurement in each experiment.

The ability to lower the temperature inside the box, made it possible to speed up the process using the other heat box. The heat box could then be adjusted to the wanted temperature, and the output from the accelerometer could be logged when the system had reached a stable temperature.

The heat closet both had the disadvantages that they were too large, in the order of 100 litres. A lot of air and a large metal structure had to be heated, making them slow, and difficult to control. In addition they both had fans to circulate air, which had to be turned off during the measurement period in order to minimize vibrations. The manual adjustment of the input angle of the rotation stage, see section 3.5.2, was very cumbersome to do inside the box. The systems described in the next subsection were better suited for the applications in this thesis.

### 3.2.2 Systems using a Peltier Element

Two different systems used a Peltier element to control the temperature of the accelerometers. A 5.3W element from Melcor Systems was used, the CP1.0-31-08L. The Peltier element was placed on top of the aluminium box, which was described in section 3.1.3, leading heat to or from the box.

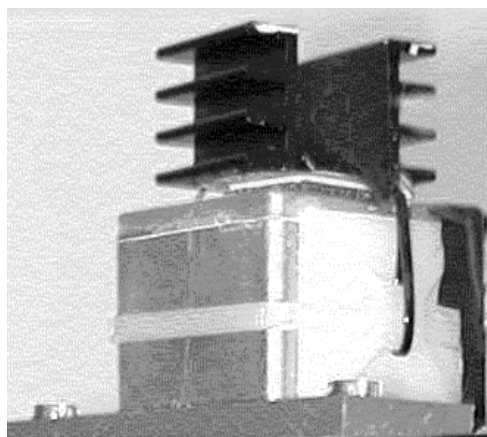


Figure (3.7): Peltier element and cooling rib attached to the aluminium box.

A cooling rib was placed on top of the Peltier element, functioning as a heat sink. The direction of the current in the Peltier element decided the direction of the heat flow. Heat leading pasta was applied on both sides of the Peltier element. The box was

attached to a slab of epoxy to isolate it thermally and electrically. A photo of the setup can be seen in figure 3.7.

### MPT5000

The MPT5000 is PID (Proportional Integrate Derivative) temperature controller with an adjustable proportional gain, produced by Wavelength Electronics. Temperature feedback to the controller was taken from an Analog Devices AD592AN temperature transducer which was glued to the inside of the lid of the aluminium box. The transducer has a proportional current output, named  $I_r$  on figure 3.8.

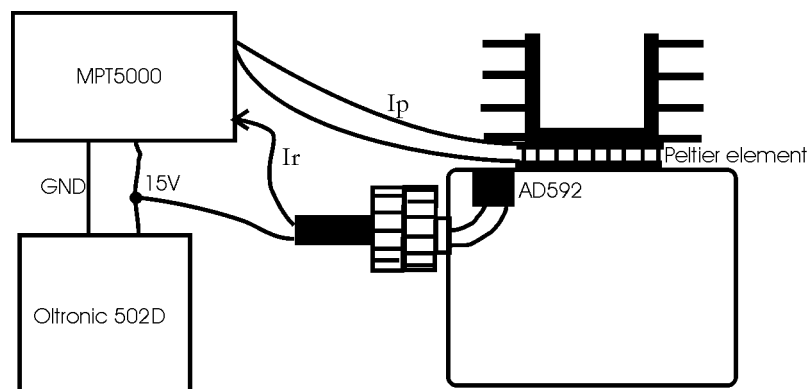


Figure (3.8): The MPT5000 temperature controller wiring

The specifications of the MPT5000 gives a short time stability of  $0.005\text{ }^{\circ}\text{C}$  or better, but was measured by the built-in ADXL105 sensor to be  $0.15\text{ }^{\circ}\text{C}$ , or  $0.1\text{ }^{\circ}\text{C}$  when the system had been allowed to stabilize for a while, typically an hour. The apparent lower performance may be because of short time variations in the built-in temperature sensor or temperature differences between the box and the accelerometer because of heat leakage through the electrical wires. It may also have been caused by short time variations in the AD592 transducer output. A short time stability of  $0.1\text{ }^{\circ}\text{C}$  was anyway considered good enough for the experiments in this thesis.

The long time stability was limited by variations in room temperature, as proves of heat leakage through the electrical wires to the accelerometer was found. This could have been improved by stopping the leakage by using thin wires, or by making some kind of thermal contact between the accelerometer and the box. The long time stability of the temperature was not important in the experiments where this temperature regulator was used.

The setpoint of the proportional gain had to be set to the lowest level, one, to keep the system stable.

The unit was driven by a separate power supply, an Oltronik B502D, as testing revealed that a linear power supply gave more stable temperature, as also specified in the datasheets.

The setpoint of the commercial temperature control unit, was shown to vary as a function of the temperature of the unit. Thus the temperature of the unit had to be kept

as stable as possible, which was done by mounting the box on a cooling rib which was supplied with a constant air flow from a fan.

### PI regulator

A PI (Proportional Integral) regulator was designed to control the current to the Peltier element. A block diagram of the total system is given in figure 3.9. This regulator used the same Peltier element as the MPT5000, but the temperature feedback was taken from the built-in ADXL105 temperature sensor. The PI regulator was chosen to obtain correct regulation.

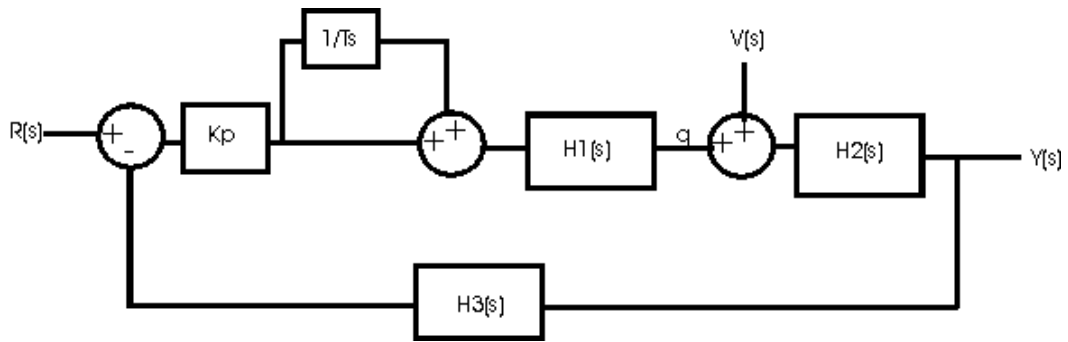


Figure (3.9): Block diagram of the PI-regulator and the controlled system.

$H1(s)$  is the transfer function of the Peltier element, and  $q(t)$  is the heat flow from the Peltier element to the aluminium box.  $V(s)$  is heat the heat flowing from the box to or from the environment.  $H2(s)$  is the heat transfer function of the box to the sensor.  $y(t)$  is the temperature of the sensor.  $H3(s)$  is the transfer function of the temperature sensor which is fed back to the system. The transfer function of a PI regulator is given by (Tyssø, 1992)

$$H(s) = K_p \left( 1 + \frac{1}{T_s} \right) \quad (3.1)$$

which is implemented by the sum of the blocks  $K_p$  and  $1/T_s$ .

The system was implemented on a circuit card, shown in appendix B. It was implemented on the same circuit card as the amplifier shown in figure 3.14, which caused some problems as discussed below.

The gain  $K_p$  and the differential part of the system was implemented with an differential amplifier as shown in the following figure, 3.10.

It had an adjustable gain, from  $K_p=0.25$  to  $K_p=2.5$ . The gain was adjusted by the two potentiometers P4 and P5. To avoid any offset, they had to be adjusted to the same value, which is not an ideal solution. The gain was not meant to be adjusted after a suitable gain had been found. The Analog Devices OP200 opamp was chosen because it is a high accuracy, low noise operational amplifier.



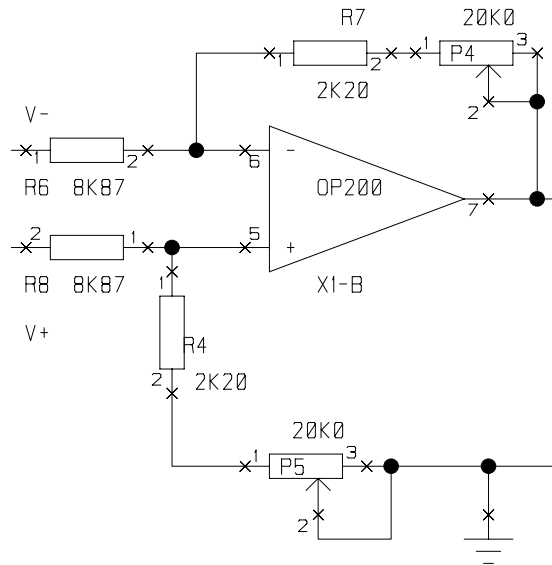


Figure (3.10): Implementation of a differential amplifier with adjustable gain. Excerpt from the circuit in appendix B.

The integrating part of the system was implemented in the following figure, 3.11. The transfer function of this circuit when the opamp is approximated to be ideal, is given by

$$H(s) = \left(1 + \frac{1}{Ts}\right) = \left(1 + \frac{1}{sRC}\right) \quad (3.2)$$

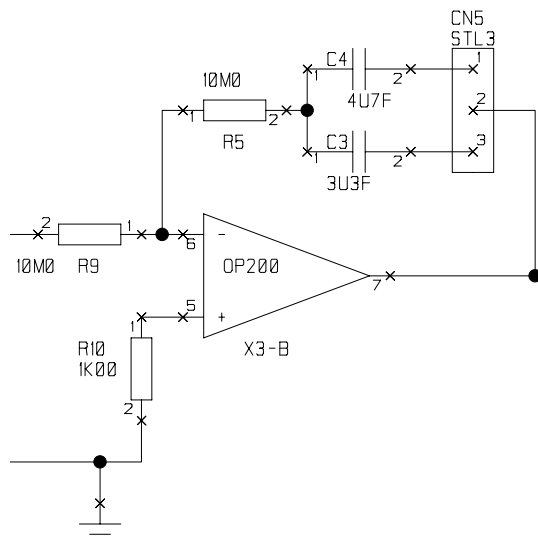


Figure (3.11): Excerpt from circuit in appendix B implementing the integrating part of the PI-regulator

R is the value of R5 and R9 on figure 3.11. C can be chosen to be C4 or C3 with a jumper on the connector named STL3. C3 and C4 were chosen as two metalized polyester capacitors because of their low leakage current. The time constant of the integrating loop was thus 33s or 47s configured by the jumper.

In order to provide the relatively high currents, up to 2.5A, to drive the Peltier element two effect transistors were used as shown on figure 3.12. The opamp provided feedback, and provides a gain of 0.1 in order to hold the output voltage within the limits of the Peltier element,  $\pm 3.75$  V. The circuit was based on examples by Hill and Horowitz (1989).

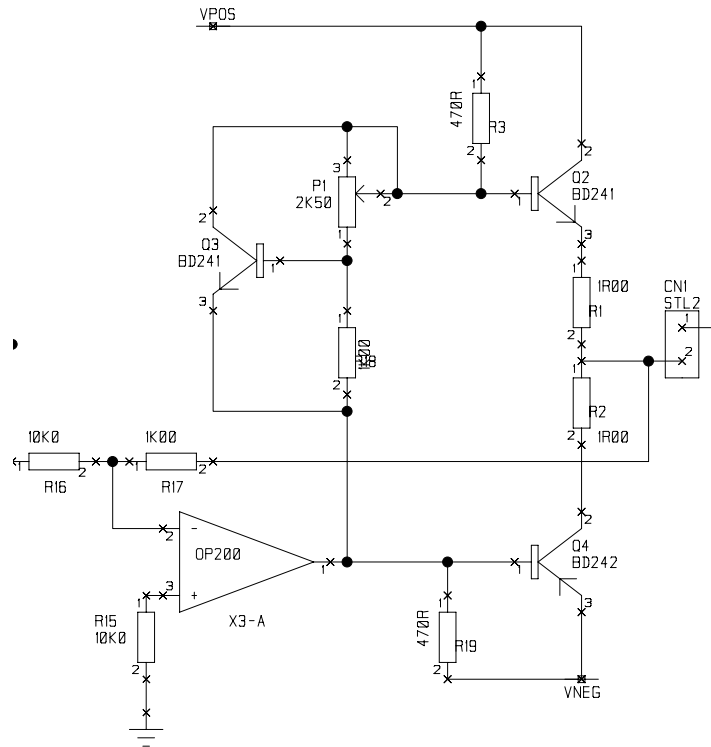


Figure (3.12): The effect amplifier constructed of PNP and a NPN transistors and an OP200 opamp to provide feedback. Excerpt from the circuit in appendix B.

As mentioned previously in this subsection, this regulator was implemented on the same circuit card as the DAQ system shown in figure 3.15. Tests revealed that the relatively high currents floating to the Peltier element influenced and distorted the analog signal conditioning because of the way the circuit was routed.

Testing showed that the circuit worked and showed good long time stability when the gain was adjusted to about one, but the dynamic performance was relatively poor, as expected. The turn-in time was much longer than that of the MPT5000 controller, which made the MPT5000 better suited for tests with varying temperatures.

The fact that the temperature of the ADXL105 package was used as feedback made it possible to obtain better long time stability as temperature gradients within the box, as heat leakage trough the wires did not influence the system. This also made the response of the system slower because of the increased termic resistance between the element and the temperature sensor.

The influence of the temperature control system on the analog signal conditioning could have been avoided by modifications on the circuit, but the MPT5000 was

available before this was done, and proved to have a better dynamic performance. Thus the MPT5000 unit was used during almost all of the experiments.

### 3.3 DAQ systems

Two different Data Acquisition (DAQ) setups were used. Both setup were based on a combination of a circuit card with analog signal conditioning, and a commercial DAQ card from Dspace or National Instruments.

The first setup used a card from National Instruments, the DAQcard-16-E-4, mounted in a portable computer. The second setup used a card manufactured by Dspace, DS1102A, mounted in a stationary computer.

The ELAB at the Physics Department, University of Oslo produced all the circuit cards containing the analog signal conditioning. A HP6236B triple output power supply provided  $\pm 15V$  and ground to the circuit cards.

#### National Instruments Setup

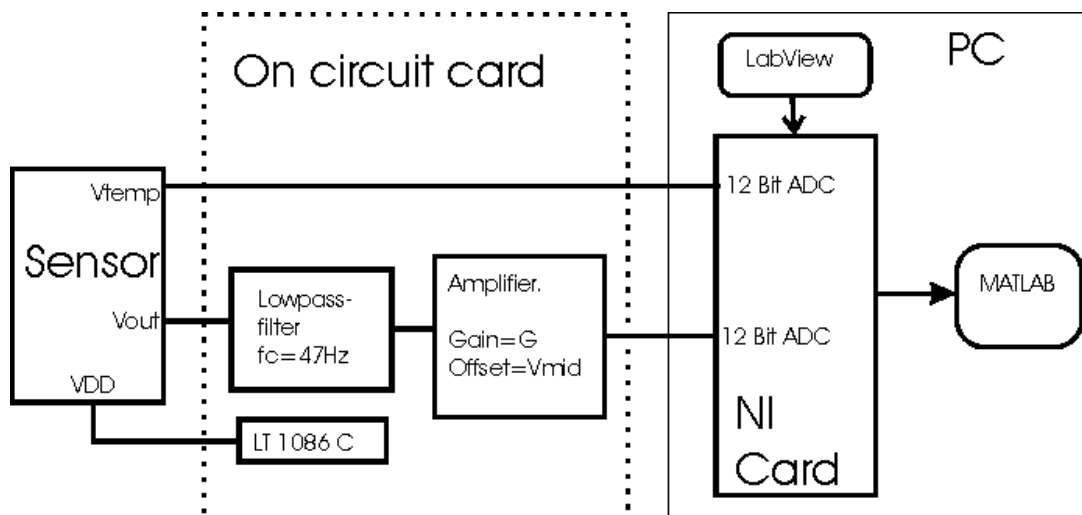


Figure (3.13): The DAQ system during the static experiments

A schematic drawing of the system is shown on figure 3.13.

The lowpass filter on figure 3.13 was a first order passive RC-filter, with  $R = 1000\Omega$  and  $C = 3.3\mu F$ , giving the cutoff frequency  $f_c = 1/(2\pi RC) = 47Hz$ . The component values do not match the values in the figure.

The signal was amplified to increase the signal-to-noise ratio, using an AD620 instrumental amplifier. The AD620 component is a high accuracy, low noise device from Analog Devices. The gain of the amplifier was set by the value of resistance R13 on the schematics showed on figure 3.14. A metal film resistor with a nominal error of 1% was used to prevent any long term changes in the gain. A bias  $V_{mid}$  was added to the signal in order to keep the signal to the NI-card within the ADC range, making the

output from the amplifier  $V_o = G \cdot (V_i - V_{mid})$ .  $V_{mid}$  was set by adjusting a potentiometer P2. The value of  $G$  and  $V_{mid}$  was calculated by using known input signals, measuring the outputs and using a line fit. The problems described in the next

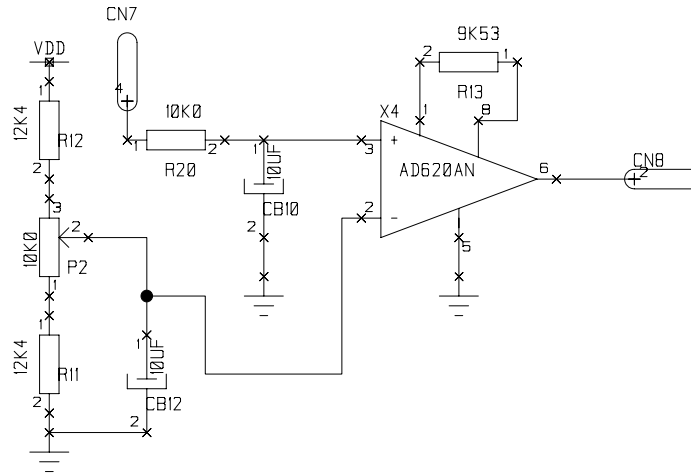


Figure (3.14): The amplifier block in figure 3.13 and 3.15, excerpt from circuit in appendix B

subsection, with long term variations in the gain and the  $V_{mid}$  had not yet been discovered when this setup was used in the experiments.

The offset adjustment input of the amplifier, pin number five, was connected to ground. The datasheets of the AD620 suggested to use the zero-output of the amplifier as the negative input of a differential ADC, but this setup was shown to give better results.

The DAQ card was of the type DAQcard-16-E-4, which has four 12 bits ADC's. The four 12 bits ADCs has individually adjustable gains, polarities and input areas. The noise floor was specified to be approximately 1/2 LSB RMS. The ADCs were configured as differential inputs, and the negative input were connected to signal ground.

The data acquisition were controlled with a LabView VI, and stored and later processed in MATLAB.

All cables between the accelerometer and the DAQ circuit, the DAQ circuit and the NI card were shielded and grounded.

The signal from the accelerometer's built-in temperature sensor was connected directly to the NI card without any buffering so that the signal picked relatively much electrical noise. The noise was removed by taking the mean over time, as only long time variations in the signal was of any importance.

### White noise tests

A way to verify the DAQ setup is to calculate the RMS value of the measured system. The datasheets of the ADXL105 specifies a noise floor between  $225 \mu g / (\sqrt{Hz})$  and

$325\mu\text{g}/(\sqrt{\text{Hz}})$ , which gives that the standard deviation of the output signal should be between 0.79mg and 1.1mg with a 12Hz bandwidth signal. Using a digital 20. order digital Butterworth filter with cutoff frequency 12Hz on 20s of a measured signal from the ADXL105 accelerometer number one, gave a signal with standard deviation 0.82mg, calculated with the nominal accelerometer scale factor from the ADXL105 datasheets, 0.25V/g.

The corresponding results with the Crossbow accelerometer was a measured to a standard deviation of  $107\mu\text{g}$ , calculated with the nominal scale factor 0.833V/g, compared to the nominal  $69\mu\text{g}$  calculated from the spectral density  $20\mu\text{g}/(\sqrt{\text{Hz}})$ .

The conclusion drawn from these tests was that the DAQ setup has sufficient accuracy to do experiments with the ADXL unit, while the results could be improved by choosing another setup when doing measurements on the Crossbow accelerometer.

### Dspace Setup

Figure 3.15 shows a simplified schematic drawing of this setup. The system shown on the figure is implemented on the same circuit card as the temperature control system described in section 3.2.2. Appendix B contains the complete schematics of the circuit.

The cutoff frequency of the lowpass filter was set to  $f = 1/(2\pi RC) = 159\text{Hz}$ . The frequency was set higher to decrease phase shifts in the dynamical area of hand movements, 0 to 12Hz.

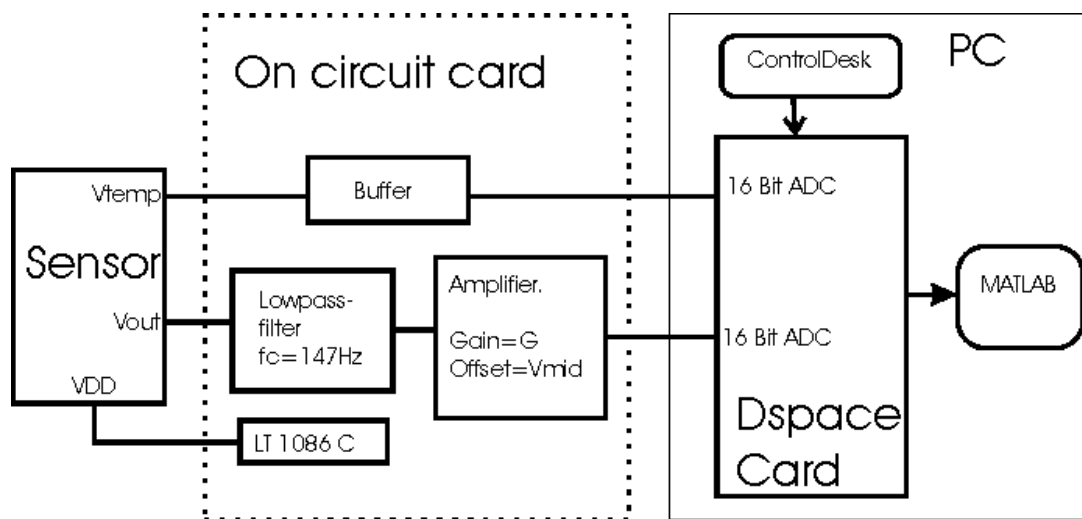


Figure (3.15): Principal drawing of the analog signal conditioning

The gain and the offset of the amplifier were configured the same way as on figure 3.14. When what at the time seemed like asymmetries in the DAQ setup were discovered, extensive testing on the linearity and symmetry of the DAQ system were performed, see the following subsection.

A buffer was inserted between the sensor's temperature output and the ADC, in order to increase the quality of the transfer, as the sensor showed unable to provide signal to both the ADC and the temperature control system.

As shown in figure 3.15, the analog signal was sampled with one of the ADCs of the DS1102A-card. The DS1102 contains two 16bits ADCs with adjustable sampling period down to 4us, and two 14 bit ADCs. The input signal range is constant -20 to 20V. It also contains four 14 bit DACs.

The DS1102 is a real-time card with a built in processor from Texas Instruments which can be programmed from libraries in Simulink via MATLAB, or directly with C-code compiled with the Texas Instruments C-compiler. Dspace provides a program, ControlDesk, which makes real time control over the data acquisition possible. The sampled data are streamed to disk and later converted into files which are processed in MATLAB. If real-time processing was needed, it could be done with the Dspace card, but it has limited computational resources.

All analog data cables were shielded, the accelerometer was kept in a metal box at ground potential, and the circuit card was shielded in a metal box kept at ground potential to shield it from electrical noise.

### Testing the Gain and the Offset of the Amplifier

When the amplifier block was thoroughly tested, quite late in the process, several irregularities were revealed. The most important discovery were that the offset to the instrumentation amplifier adjusted by the potentiometer P2 in figure 3.14,  $V_{mid}$ , had long time variations up to 3 mV.

To calculate the offset  $V_{mid}$  and the gain  $G$ , controlled inputs where applied to the amplifier input, and the output was measured. The output as a function of the input was then fitted to a straight line,  $ax + b$ , using the methods of least squares. The gain was then estimated as  $G=a$ , and  $V_{mid}$  as  $V_{mid} = -b/G$ . The standard deviation in the estimated  $V_{mid}$  was then given as (Squires, 1968)

$$\left(\frac{\Delta V_{mid}}{V_{mid}}\right)^2 = \left(\frac{\Delta a}{a}\right)^2 + \left(\frac{\Delta b}{b}\right)^2 \quad (3.3)$$

Several series were run, where the input of the amplifier block was excited by one of the DACs in the DS1102 card, and the output measured by one of the 16 bits ADCs. Testing was also done to verify that the ADCs were stable over time. Testing over 24 hours using a stable external voltage reference showed a long time stability for the ADC of the order 0.3 mV. The output of the DS1102 DACs were measured to be accurate within the accuracy of a Fluke 45 digital multimeter. Measurements of the output from the DAC output from the Dspace card directly with the ADC showed no systematic errors.

The results in table 3.1 and table were obtained by stepping the DAC from the lower input limit, 2.544V and increased by steps of  $20V/2^{12} = 4.88mV$  to the upper limit, 3.091V. The output on each step was constant for 1 second in the first series, first

column in table 3.1, and 2 seconds for the second and the third series. The input range was repeated twice during the last measurement.

The results in table 3.2 to table 3.4 were done after adjusting the gain, but leaving the offset unchanged. The input was then varied from 2.034V to 3.602V, with the same step size and the input constant 1 second on each step. The offset was adjusted to get the results in table 3.5, leaving the gain unchanged. Thus the input range was changed, but the span of the area was the same.

Table (3.1): Results from three consecutive measurements, 23st May 2003 t

Time	Total gain	Standard deviation gain	Vmid (V)	Standard deviation Vmid (V)	Gain for Vin<0 V	Gain for Vin>0 V
20:50	36.4075	1.7e-3	2.8171	3e-4	36.3975	36.3896
20:56	36.4047	2.3e-3	2.8170	3e-4	36.3950	36.3876
21:08	36.4036	1.7e-3	2.8170	2e-4	36.3936	36.3880

Table (3.2): Results from three measurements, 4th June 2003, 1. series immediately after power up.

Time	Total gain	Standard deviation gain	Vmid (V)	Standard deviation Vmid (V)	Gain for Vin<0 V	Gain for Vin>0 V
14.54	12.7335	3e-4	2.82010	7e-5	12.7267	12.7343
15.19	12.7282	2e-4	2.81971	5e-5	12.7201	12.7302
23.36	12.7286	2e-4	2.81800	5e-5	12.7218	12.7296

Table (3.3): Results from three consecutive measurements, 3rd June 2003

Time	Total gain	Standard deviation gain	Vmid (V)	Standard deviation Vmid (V)	Gain for Vin<0 V	Gain for Vin>0 V
20:18	12.7274	2e-4	2.81988	5.6e-5	12.7196	12.7294
20:37	12.7291	2e-4	2.82007	6.6e-5	12.7215	12.7312
20:55	12.7279	2e-4	2.81995	5.8e-5	12.7196	12.7299

Table (3.4): Results from measurements, 5th June 2003

Time	Total gain	Standard deviation gain	Vmid (V)	Standard deviation Vmid (V)	Gain for Vin<0 V	Gain for Vin>0 V
16:30	12.7306	2e-4	2.50967	5e-5	12.7298	12.7362

Table (3.5): Results from measurement, 10th June 2003

Time	Total gain	Standard deviation gain	Vmid (V)	Standard deviation Vmid (V)	Gain for Vin<0 V	Gain for Vin>0 V
Unknown	12.7286	2e-4	2.51021	5e-5	12.7277	12.73396

As can be seen from the results there are clear indications of a asymmetry in the gain, although the asymmetry is not the same for the two different gains. The variation in the calculated gain seems to be about ten times the size of the calculated standard deviation, indicating unmodelled effects which cause random variations in the estimations.

The variations in the Vmid are possibly correlated with temperature or humidity. The apparent asymmetry in the gain is probably due to non linearity in the output stage of the amplifier. During the measurement which results are listed in table 3.2, the offset varied with 2 mV, adding more evidence to the assumption that the variations in offset were correlated with temperature, as the room temperature in the lab decreased during the evening and night. The cause of the variations may be temperature dependent leakage current in the capacitance CB12 on figure 3.14, or temperature dependent offset in the amplifier.

The gain and offset was measured immediately before each experiment after the discovery of this phenomenon. The different gains for inputs larger than 0 and less than zeros were then used to calculate back to the input as a function of the output of the amplifier block.

#### White noise test

The signal from the ADXL105 number three in a static position filtered with a 7. order lowpass Butterworth filter with cutoff frequency of 12 Hz gave a noise floor of 0.72 mg. The order of the filter had to be decreased from 20 to 7 in order to cope with the increased sampling frequency, 1.667 kHz. The same filter applied with a cutoff frequency of 15 Hz gave a standard deviation of 0.78 mg. The results were calculated using the nominal scale factor of 0.25 V/g

Even though the Butterworth filter is the maximally flat filter, the decreased filter order probably causes significant damping of the signal below the cutoff frequency, such that the real value of the noise floor probably is around 0.8 mg. The conclusion is anyway that the DAQ setup passed the white noise test.

### 3.4 Experimental Environment

The ideal environment to conduct static experiments would be a room where all external vibrations from passing buses etc. is removed by a damping system. We did not have such an environment available, so the sensor most likely picked up vibrations from slamming doors, running fans and similar devices. These vibrations will be seen as noise, but they are probably zero mean and below the Nyquist frequency, and can be removed by averaging over time.



During the first experiments when no temperature control system was available, the most primitive form of temperature control was applied: choosing a room with a stable ambient temperature. A room in the cellar of the SINTEF building was shown to have a very stable temperature over time, although noisy with much vibrations.

Some experiments were done on an optics table with passive damping. The ambient temperature in this room varied with about 1 degree Celsius.

The dynamic experiments and the last static experiments were performed at ABB, Billingstad and the temperature control systems described in subsection 3.2.2 were used. Thus the temperature control was good, but the environment proved noisy with both vibrational and electrical noise.

## 3.5 The Static Experiments

### 3.5.1 Main Principle and Goals

The static experiments conducted was based on the ‘Multi point tumble test’ procedures described in the IEEE document ‘Guide and Test Procedure for Linear Single-Axis, Nongyroscopic Accelerometers’ (IEEE Std 1293-1999). The acquired data was analysed using the methods of linear least squares estimation described in section 2.5.4.

As described in section 2.2.4 the output of an accelerometer is given by the specific force,  $f = \partial^2 \underline{x} / \partial t^2 - g$ . When the acceleration relative to inertial space equals zero, the output is given as a function of the gravity vector projected onto the accelerometer input axis.

Using a rotation stage as described in following subsection, makes it possible to characterize the accelerometer output as a function of input acceleration in the input area  $[-g, g]$ , where  $g$  is the absolute value of the gravity vector  $g_0$  at the specific experiment location.

The first aim to build up an experimental setup and develop test series to be able to characterize an accelerometer. The experiments were run with 5 different Analog Devices ADXL105 units, and one Crossbow CLX02TG3 three axis accelerometer.

The goal was to model and characterize each individual accelerometer based on the IEEE accelerometer model equation listed in appendix D. How good results could be achieved with the available experimental equipment? How good could the parameters of a given accelerometer be characterized?

The next experiments aimed to use static four point tumble tests to look at temperature dependent variations in scale factor and offset. As described in section 2.4 one of the goals was to examine whether these changes are repeatable, and thus can be compensated for.

A number of experiments were also carried out to examine long-time variations, like low frequency noise, random walk and similar phenomena.

### 3.5.2 The Experimental Setup

#### Rotation Stage

A rotation stage was used to fix the accelerometer axis in a given static position. A schematic drawing of the device is shown in figure 3.16. Data sheets and the name of the manufacturer were not available, thus the resolution and accuracy had to be decided by testing.

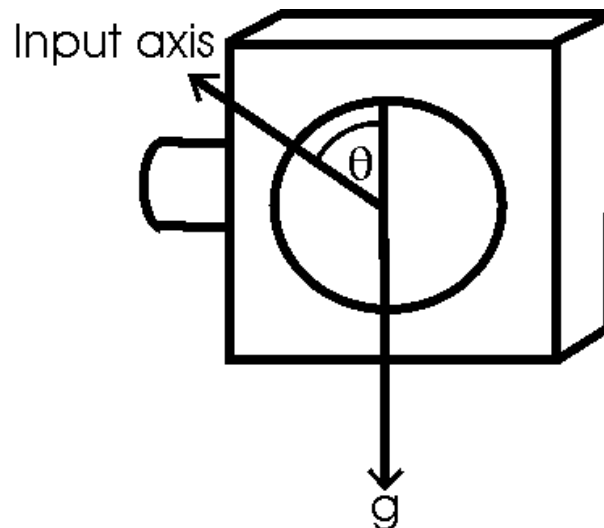


Figure (3.16): Schematical drawing of the rotation stage.

The rotation stage was mounted on a 3mm thick sheet of aluminium with a 90 degrees bend. As shown on the photograph in figure 3.17, the lower half of the aluminium sheet was attached to a tilt platform. The function of the tilt platform was to align the plane of rotation to the gravity vector. The tilt platform was attached to a relatively heavy slab of aluminium to keep the system stable and damp vibrations.

The input angle of the rotation stage was adjusted by hand, making the multipoint tumble test a time consuming process. The ideal instrument would be some kind of high accuracy automated device.

#### Specific force along the input axis

If the rotation axis is perfectly aligned normal to the gravity vector, and the angle  $\theta$  is defined as on figure 3.16, the static acceleration along the accelerometer input axis given by the specific force:

$$f_i = \ddot{x}_i - g_i = -g \cos(\theta) \quad (3.4)$$

The difference compared to the expression given in the IEEE document (IEEE Std 1293-1999),  $f = g \sin(\theta)$ , is due to different choice of the direction where  $\theta = 0$ .

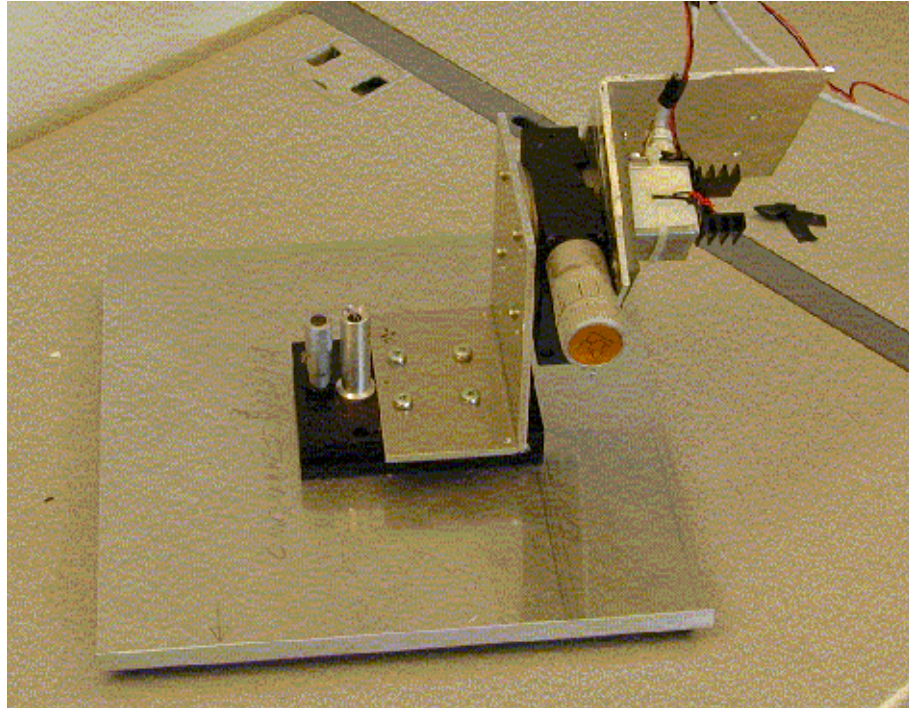


Figure (3.17): Experimental setup for the multipoint tumble tests

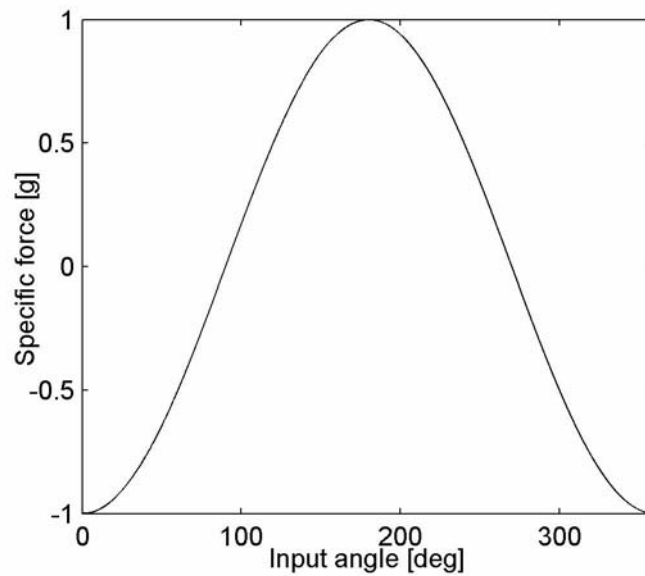


Figure (3.18): Specific acceleration as a function of input angle

### Specific force along the normal axis

As can be seen from the IEEE model equation, the output from the accelerometer can also depend on acceleration normal to the input axis. Thus the acceleration along the principle and the output axis also have to be calculated.

The acceleration along the axis normal to the input axis, depends on which axis the system is rotated about. With the input, output and principle axes, described in the IEEE standard, is defined as a right hand system as shown on figure 3.1, and the rotation axis directed into the rotation plane, the acceleration along the output axis for rotations about the principle axis is given by:

$$f_o = g \sin(\theta) \quad (3.5)$$

The acceleration along the principal axis when the system is rotated about the output axis is given by:

$$f_p = -g \sin(\theta) \quad (3.6)$$

### Measurement Resolution

The resolution of the rotation stage used in the experiments was  $0.05^\circ$ , giving an approximate accuracy of  $\Delta\theta = \pm 0.05^\circ$ . The output acceleration depends on the input as a cosine function. Using (Squires, 1968):

$$\Delta Z = \frac{\partial}{\partial X} f(X) \cdot \Delta X \quad (3.7)$$

the uncertainty in the output acceleration is given as

$\Delta f_i = g \Delta\theta |\sin(\theta)| = 0.8 mg \cdot |\sin(\theta)|$ . This expression gives no error at  $\theta = k\pi$ , which is unrealistic. Using  $\Delta Z = f(X + \Delta X) - f(X)$  gives  $\Delta f_i(k\pi) \approx g |\cos(\Delta\theta) - 1| = 38 \mu g$ .

Combining these expressions gives that the error in the calculated specific force varies with position approximately as

$$\Delta f(\theta) = 38 \mu g + 0.8 mg \cdot |\sin(\theta)| \quad (3.8)$$

### Gravity Constant

The value of the gravity constant has been measured at the SINTEF building, Oslo Norway to be  $g = (9.81897919 \pm 2 \cdot 10^{-7}) m/s^2$ . Effects of seasonal and lunar variations were considered to insignificant in these measurements, and ignored. In the absence of any measurement, the value of  $g_0$  at the ABB building at Billingstad, Norway, was considered to be of the same order. The formulas of Steiler and Winter (Titterton and Weston, 1997) could be used to obtain a value, but applied to the location of the SINTEF building ( $60^\circ N$ , 100m above sea level) the equation gives:

$$\begin{aligned} g &= 9.780318(1 + 5.3024 \cdot 10^{-3} \sin^2(60^\circ) - 5.9 \cdot 10^{-6} \sin^2(120^\circ)) m/s^2 \\ &\quad - 3.0877 \cdot 10^{-6} (1 - 1.39 \cdot 10^{-3} \sin^2(60^\circ)) \frac{m}{s^2} / m \cdot 100m \\ &= 9.81916601 \end{aligned} \quad (3.9)$$

This value deviates from the measured value with  $1.8 \cdot 10^{-4} m/s^2$  or 0.002%. Thus for very sensitive applications it would probably be wise to measure the gravity vector instead of relying on the Steiler and Winter formulas.

### Errors due to Misalignment to the Gravity Vector

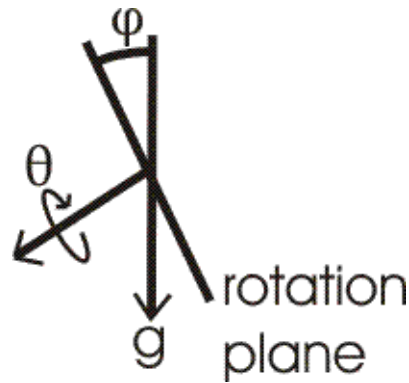


Figure (3.19): Misalignment to the gravity vector

If the rotational plane is misaligned to the gravity vector with an angle  $\varphi$  as shown in figure 3.19, the output as a function of the misalignment and the input angle is given by

$$f_i = -g \cos(\theta) \cos(\varphi) \quad (3.10)$$

which for small misalignment angles can be approximated to

$$f_i \approx -g \cdot \cos(\theta + \varphi) \quad (3.11)$$

This means that all estimations of the input acceleration are biased by a factor of  $\cos(\varphi)$ . This error will bias the estimation of several parameters, like the scale factor and higher order coefficients.

For instance, if  $\varphi$  is 1 degree, all the measurements get multiplied by the factor  $\cos(\varphi) = 0.99984$  which gives an error in the measurements of about 0.15 mg. If  $\varphi$  equals 0.1 degree, the error is about  $1.5 \mu g$  and 0.01 degrees gives about 16 ng.

Adjustments of the plane of rotation to the gravity vector were done with the tilt platform shown in figure 3.20. The tilt platform had a resolution of 0.01 mm on the micrometer screw from Starret. The length of the axis was measured to be 100 mm, giving the one mm adjustment to be equal to  $\text{atan}(1/100) = 0.01$  degrees, and a resolution of  $10^{-4}$  degrees. The input range was 13 mm.

Two different devices were used to calibrate the system. During the first experiments a spirit level device was used which had an accuracy of approximately 0.5 degrees, giving an error of about 0.04 mg. This error is insignificant compared to other measurement errors.

During later experiments a Sola Lasertronic inclinometer with laser projector with an accuracy of 0.05 degrees was used, making the error due to misalignment of the

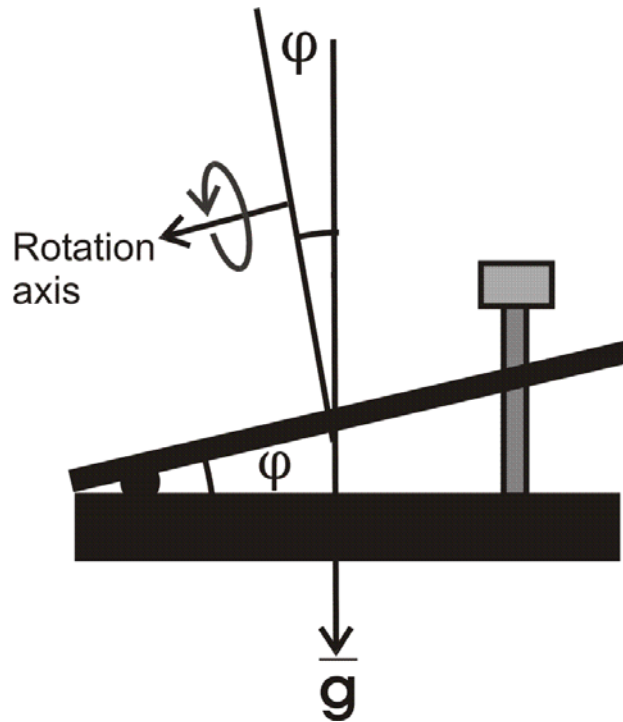


Figure (3.20): Schematical drawing of the tilt platform

rotation plane equal to  $0.38\mu g$ , which is of the same order as the accuracy in the measurement of the gravity constant.

The input axis of the accelerometer under testing had to be aligned to the plane of rotation, or the misalignment had to be estimated. The methods described in the following subsection were developed to do that.

### 3.5.3 Methods of Aligning the Input Axis to the Plane of Rotation.

As mentioned previously, aligning the accelerometer axis to the plane of rotation can be quite tricky. Even if a method of aligning the package to the plane was developed, there still could be up to one degree misalignment between the package and the input axis as specified in the ADXL105 datasheets.

The input axis may be misaligned to the rotation plane with an angle  $\varphi_o$ , but it may also be misaligned in the plane of rotation with an angle  $\theta_0$  to the vertical where  $\theta = 0$ . The challenge was to adjust these angles to be zeros, or to estimate and compensate for the misalignments.

- Until other methods were developed the angle  $\varphi_o$  where adjusted manually as seen on figure 3.21. After the plane of rotation had been aligned to the gravity vector using the inclinometer or a spirit level device, the screws where adjusted to the point where  $V_{out}$  from the accelerometer was minimized. The angle  $\theta_0$  was estimated by adjusting the rotation stage input angle around  $\theta = 0$  until the point with the minimum output where found.

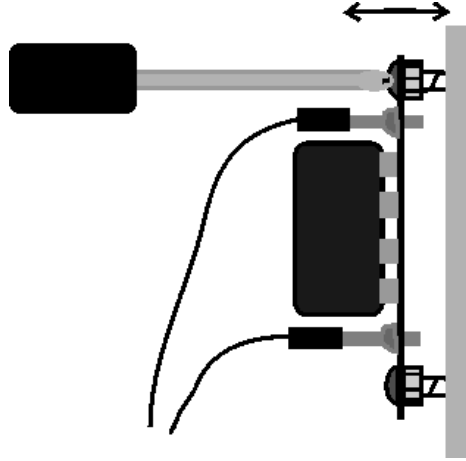


Figure (3.21): Manually adjusting the misalignment angle  $\varphi_0$

- The second method developed, used nonlinear least squares estimation with functions from the MATLAB statistics toolbox. The tilt platform was used to set the input angle  $\varphi$ , and the accelerometer output was logged for each input angle. The output could then be fitted to a cosine-function on the form

$$-V_0 \cos(\varphi - \varphi_0) \quad (3.12)$$

where  $\varphi_0$  indicates the maximum angle. The same method can be used to find the misalignment in the rotation plane  $\theta_0$  by nonlinear estimation to the function

$$-V_0 \cos(\theta - \theta_0) \quad (3.13)$$

The procedure has to be repeated at number of times to compensate for bias random drift. The disadvantages are that it is quite time-consuming, and has to be redone each time the accelerometer is remounted. In addition, it only estimates the angles  $\varphi_0$  and  $\theta_0$ , and has to be redone if adjustments are done on the screws.

- The last and preferred method, was to use least square estimation on the tumble test data, see section 3.5.5, to estimate the misalignment constant using the model equation given in equation (3.20). Then an angle  $\theta_0$  is subtracted from the data, such that

$$E(\theta_j) = O - Kg \cos(\theta_j - \theta_0) - \delta_o g \sin(\theta_j - \theta_0) \quad (3.14)$$

This expression is calculated until the point is found where  $|\delta_o|$  is minimized. This angle was then assumed to be the misalignment in the rotation plane. The experiments showed that this gave the same result as nonlinear estimation using equation (3.13). This method can only be done for the axis about which the accelerometer is rotated.

Mounting the accelerometer on a sheet of metal with a  $90^\circ$  corner as shown on figure 3.22, rotations around both the output and the principle axis are possible

without remounting the accelerometer on the metal. Thus the angle  $\delta_p$  can also be estimated, with the same method as for the output axis by attaching the other

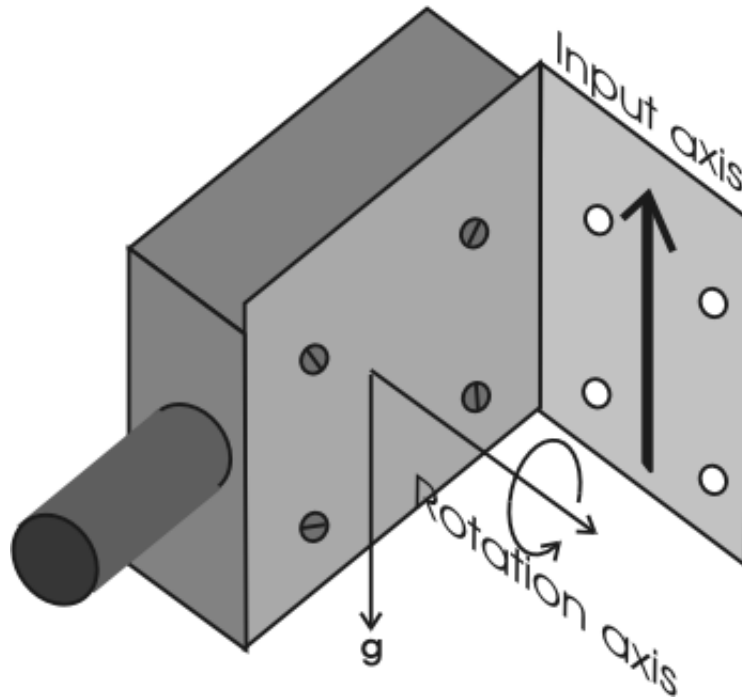


Figure (3.22): Setup used to rotate the accelerometers about both the principle- and the output- axis

surface of the metal sheet to the rotation stage.

In this way the equation (3.10) can be combined with the equations (3.18) and (3.19)

$$f_i(\theta) = -g \cos(\theta) \cos(\varphi_0) = -g_\theta \cos(\theta) \quad (3.15)$$

$$f_i(\varphi) = -g \cos(\theta_0) \cos(\varphi) = -g_\varphi \cos(\varphi) \quad (3.16)$$

If the  $g$  is replaced by  $g_\theta$  and  $\theta$  by  $\theta - \theta_0$  in equation (3.18), and  $g$  by  $g_\varphi$  and  $\varphi$  by  $\varphi - \varphi_0$  in equation (3.19), all the estimates are corrected for misalignments.

Since the estimations of  $g_\theta$  and  $g_\varphi$  depended on each other, a few repetitions were necessary where the updated values of each constant was inserted into the equations (3.18) and (3.19) until convergence was reached.

### 3.5.4 Data Analysis

As mentioned in section 3.5.1, the data gathered in the multi-position tumble tests was analysed using the method of linear least squares estimation.

#### With small Angle Assumption

The small axis assumption means that the misalignment of the plane of rotation relative to the vertical is estimated to be zero.



The IEEE equation for a multitumble test around the output axis with a small angle approximation is given by

$$\begin{aligned}
f(\theta_j, \alpha) = & K_0 + \frac{1}{2}K_0' \text{sign}(\cos(\theta_j)) + (1+k)\cos(\theta_j) + \delta_0 \sin(\theta_j) \\
& + \frac{1}{2}K_1'g \cdot \text{sign}(\sin(\theta_j)) + K_{oq} \sin(\theta_j)|\sin(\theta_j)| + K_2 \cos^2(\theta_j) + K_3 \cos^3(\theta_j) \\
& + K_{ip} \cos(\theta_j)\sin(\theta_j) + K_{pp} \sin^2(\theta_j)
\end{aligned} \tag{3.17}$$

As referred in section 2.4, the accelerometer output is modelled in Volts instead of acceleration units. The modified version of equation (3.17) becomes

$$\begin{aligned}
E(\varphi_j, \alpha) = & O + O_{ag} \cdot \text{sign}(\cos(\varphi_j)) - Kg \cos(\varphi_j) - \delta_o g \sin(\varphi_j) \\
& + K_{ag} \cdot \text{sign}(\cos(\varphi_j))\cos(\varphi_j) - K_{oq}g^2 \cos(\varphi_j)|\cos(\varphi_j)| \\
& + K_2g^2 \cos^2(\varphi_j) - K_3g^3 \cos^3(\varphi_j) + K_{ip}g^2 \cos(\varphi_j)\sin(\varphi_j) + K_{pp}g^2 \sin^2(\varphi_j)
\end{aligned} \tag{3.18}$$

For a tumble test about the principle axis the corresponding equation is given by

$$\begin{aligned}
E(\theta_j, \alpha) = & O + O_{ag} \cdot \text{sign}(\cos(\theta_j)) - Kg \cos(\theta_j) + \delta_p g \sin(\theta_j) \\
& + K_{ag} \cdot \text{sign}(\cos(\theta_j))\cos(\theta_j) + K_{oq}g^2 \cos(\theta_j)|\cos(\theta_j)| \\
& + K_2g^2 \cos^2(\theta_j) - K_3g^3 \cos^3(\theta_j) - K_{io}g^2 \cos(\theta_j)\sin(\theta_j) + K_{oo}g^2 \sin^2(\theta_j)
\end{aligned} \tag{3.19}$$

Estimation without the small axis assumption can be done by replacing  $g$  by  $g_\theta$  in equation (3.18) and  $g_\varphi$  in equation (3.19).  $g_\theta$  and  $g_\varphi$  are calculated as described in section 3.5.3.

### Correlated Parameters

Some of the parameters in the model equation are perfectly correlated in a Multi position tumble test. By doing a Fourier expansion on the model equation it can be shown (IEEE Std 1293-1999) that  $K_2$  and  $K_{pp}$  are perfectly correlated in a tumble about the output axis and  $K_2$  and  $K_{oo}$  are perfectly correlated in tumble test about the principle axis. As mentioned in the IEEE standard a centrifuge or a vibration test is needed to estimate these square coefficients individually.

The Fourier expansion of the model equation also shows that if  $K_2$  is not estimated, the offset is biased by a fraction of the value of the second order coefficient. Similarly if  $K_3$  is not estimated but set to 0, the scale factor is biased by one third of the third order coefficient.

### 3.5.5 Estimation models

Several different algorithms were implemented, where different selections of the parameters from the equations (3.18) and (3.19) were estimated.

Doing a tumble about the principle axis, and choosing to estimate the parameters scale factor, offset and misalignment simplifies equation (3.19) to

$$E(\theta) = O - K \cdot g \cos(\theta) - \delta_o \cdot g \sin(\theta) \quad (3.20)$$

This gives the least square estimation problem

$$J(O, K, \delta_o) = \sum_n (E(\theta_n) - O + K \cdot g \cos(\theta_n) + \delta_o \cdot g \sin(\theta_n))^2 \quad (3.21)$$

Solving the least squares estimation problem formulated in equation (2.58) for the parameter vector  $\alpha = [O, K, \delta_o]^T$

$$\frac{\partial J}{\partial O} = -2 \sum_n (E(\theta_n) - O - K \cdot g \cos(\theta_n) - \delta_o \cdot g \sin(\theta_n)) \quad (3.22)$$

$$\frac{\partial J}{\partial K} = 2 \sum_n \cos(\theta_n) (E(\theta_n) - O + K \cdot g \cos(\theta_n) + \delta_o \cdot g \sin(\theta_n))$$

$$\frac{\partial J}{\partial \delta} = 2 \sum_n \sin(\theta_n) (E(\theta_n) - O + K \cdot g \cos(\theta_n) + \delta_o \cdot g \sin(\theta_n))$$

gives the linear system of equations:

$$\begin{bmatrix} n & -g \sum_n \cos(\theta_n) & -g \sum_n \sin(\theta_n) \\ -g \sum_n \cos(\theta_n) & g^2 \sum_n \cos(\theta_n)^2 & g^2 \sum_n \cos(\theta_n) \sin(\theta_n) \\ -g \sum_n \sin(\theta_n) & g^2 \sum_n \cos(\theta_n) \sin(\theta_n) & g^2 \sum_n \sin(\theta_n)^2 \end{bmatrix} \begin{bmatrix} O \\ K \\ \delta_o \end{bmatrix} = \begin{bmatrix} \sum_n E(\theta_n) \\ -g \sum_n E(\theta_n) \cos(\theta_n) \\ -g \sum_n E(\theta_n) \sin(\theta_n) \end{bmatrix} \quad (3.23)$$

The solution of the normal equation to get the uncertainty in the estimated parameters as described in section 2.5.4 is also implemented in the function.

As shown in section 3.5.2, the accuracy of the results are not constant for all angles. The uncertainty varies as a function of angle because of the error input angle, not because of different accuracy in the measurement of the output signal. The least squares method can be implemented with a weighting vector giving

$$J(O, K, \delta) = \sum_n \frac{(E(\theta_n) - O + K \cdot g_0 \cos(\theta_n) + \delta \cdot g_0 \sin(\theta_n))^2}{w_n^2} \quad (3.24)$$

where the error as function of angle is given in equation 3.8. Normalizing the errors, and using  $w_n = 1/\Delta f_n$  gives the following expression for  $w_n$ :

$$w_n = \frac{38\mu g + |0,8mg \cdot \sin(\theta_n)|}{0,838mg} = \frac{1 + 2105 \cdot |\sin(\theta_n)|}{2106} \quad (3.25)$$

The equation (3.23) can then be reformulated as:

$$\begin{bmatrix} \sum_n \frac{1}{w_n^2} & -g \sum_n \frac{\cos(\theta_n)}{w_n^2} & -g \sum_n \frac{\sin(\theta_n)}{w_n^2} \\ -g \sum_n \frac{\cos(\theta_n)}{w_n^2} & g^2 \sum_n \frac{\cos(\theta_n)^2}{w_n^2} & g^2 \sum_n \frac{\cos(\theta_n)\sin(\theta_n)}{w_n^2} \\ -g \sum_n \frac{\sin(\theta_n)}{w_n^2} & g^2 \sum_n \frac{\cos(\theta_n)\sin(\theta_n)}{w_n^2} & g^2 \sum_n \frac{\sin(\theta_n)^2}{w_n^2} \end{bmatrix} \begin{bmatrix} O \\ K \\ \delta_o \end{bmatrix} \quad (3.26)$$

$$= \begin{bmatrix} \sum_n \frac{E(\theta_n)}{w_n^2} \\ -g \sum_n \frac{E(\theta_n)\cos(\theta_n)}{w_n^2} \\ -g \sum_n \frac{E(\theta_n)\sin(\theta_n)}{w_n^2} \end{bmatrix}$$

Similar least squares systems were also expressed and solved for the following models. All the solutions are implemented in the function `least_squares_estimate()` listed in appendix A.

- Cross axis sensitivity, scale factor and offset.

$$E(\theta) = O - K \cdot g \cos(\theta) + K_i \cdot g^2 \sin(\theta) \cos(\theta) \quad (3.27)$$

- Cross axis sensitivity, second order coefficient, scale factor and offset.

$$E(\theta) = O - K \cdot g \cos(\theta) + K_i \cdot g^2 \sin(\theta) \cos(\theta) + K_2 \cdot g^2 \sin^2(\theta) \quad (3.28)$$

- Second and third order coefficients, scale factor and offset.

$$E(\theta) = O - K \cdot g \cos(\theta) + K_2 \cdot g^2 \sin^2(\theta) - K_3 \cdot g^3 \sin^3(\theta) \quad (3.29)$$

- Scale factor asymmetry, scale factor and offset.

$$E(\theta) = O - K \cdot g \cos(\theta) + K_a \cdot \text{sign}(\cos(\theta)) \cos(\theta) \quad (3.30)$$

- Offset asymmetry, scale factor and offset.

$$E(\theta) = O + O_a \cdot \text{sign}(-\cos(\theta)) - K \cdot g \cos(\theta) \quad (3.31)$$

- In addition all parameters in the equation (3.19) were estimated at once.

All of these models were inserted into equation (2.58) and solved to form expressions like (3.23) and (3.26).

The same models and file can be applied to rotations about the output axis, but the sign of some of the parameters has to be inverted, given the different signs in equation (3.5) and (3.6).

### 3.5.6 Measurement Series

#### Noise and Start-up Tests

##### Start-up tests

The output from an accelerometer for a static position was logged for half an hour after power-up. The signal was then analysed to check for any settling periods, and variations in the noise pattern.

##### Noise and Random-walk tests

The output from the accelerometer under test several was logged over several minutes. The output were then analysed in the frequency domain using Power Spectral Density (PSD) analysis, to look for indications of random walk, trends and other phenomena described in section 2.4.2.

The output was also analysed in the time domain using digital filtering in order to examine the shape of the offset random walk curve over time.

#### Multipoint Tumble Tests about one Axis

This setup used the rotation stage as described in section 1.2.1.

The output from the accelerometer under testing, was measured at the following angles:

$$\theta = [0, 10, 20, 30, 40, 50, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110, 115, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 245, 250, 255, 260, 265, 270, 275, 280, 285, 290, 295, 300, 310, 320, 330, 340, 350]$$

The angles are not uniformly spaced in order to get an equal number of measurement points in each of the intervals  $[-g, -g/2]$ ,  $[-g/2, 0]$ ,  $[0, g/2]$  and  $[g/2, g]$ .

The arithmetic mean of the data was taken for the measurement period, and the data was analysed using the algorithms in section 3.5.4.

#### Four Point Tumble Tests with Temperature Variations

The heat boxes described in section 3.2.1 were used to control the temperature during these experiments.

For each temperature, the accelerometer was put in four different positions, positions,  $\theta = [0, 90, 180, 270]$ . The scale factor was then calculated as

$$K = (E(180) - E(0)) / (2g) \quad (3.32)$$

and the offset as

$$O = (E(90) + E(270)) / 2 \quad (3.33)$$

The input angle of the rotation stage had to be adjusted manually, which meant that the door to the heat box had to be opened. Opening the door meant that the inside temperature changed dramatically. Because of this, the accelerometer was heated to all the temperatures in the input range before adjusting the angle. This meant that the time span was quite long, and that the accelerometer went through four heating and cooling cycles for each experiment.

#### **Multipoint Tumble Tests about two Axes.**

A L-shaped sheet of aluminium with a  $90^\circ$  bend on it was attached to the rotation stage as previously described in section 1.2.2. The accelerometer could then be rotated about both the output and the principle axis without remounting it on the sheet.

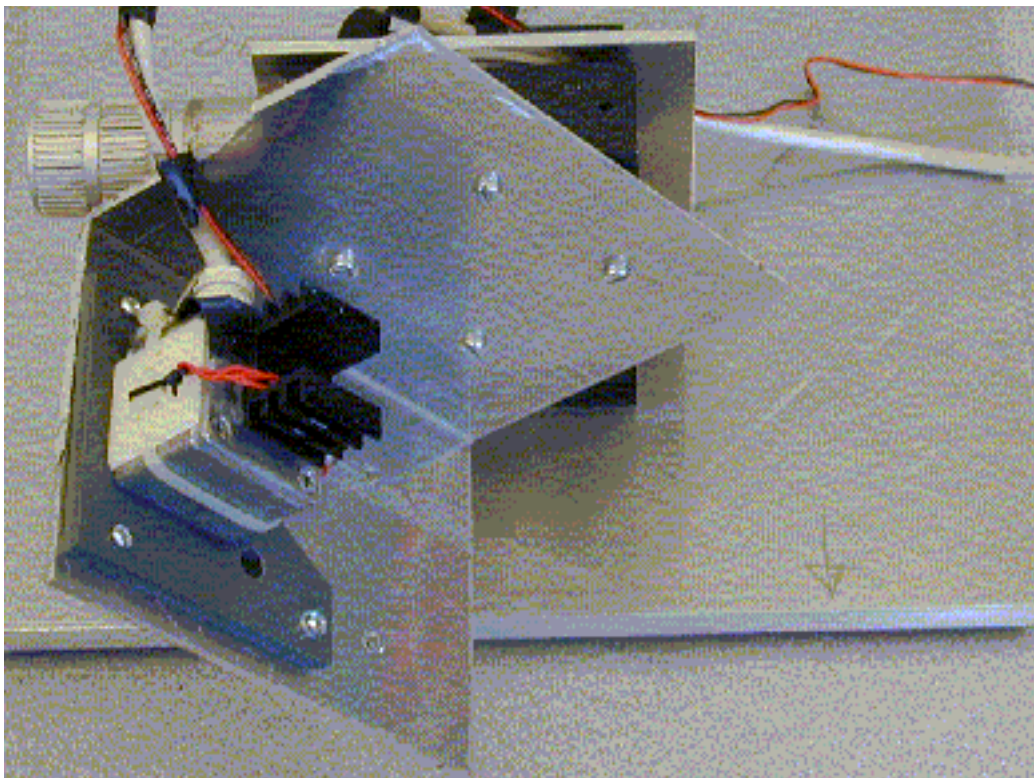


Figure (3.23): Photo of the setup, accelerometer rotated about the output axis.

Each side of the sheet can be attached to the rotation stage, and using the high accuracy inclinometer, the rotation plane can be aligned to previous mountings within  $0.05^\circ$ .

The same set of angles listed above was used for each axis.

## 3.6 Experiments using dynamic Accelerations

### 3.6.1 Main Principles

The aim of these experiments was to verify a new method to characterize accelerometers using a highly accurate capacitive position sensor, and if possible apply the method to gather information about the ADXL105 accelerometer.

When doing experiments to characterize accelerometers, the applied acceleration is often calculated through the gravity vector, with a centrifuge or through a setup using vibrations with known RMS amplitude. With this experimental setup, the acceleration

of the platform was obtained via the principle  $\frac{d^2}{dt^2}x(t) = a(t)$ . The measured position

was differentiated twice with respect to time using the Savitzky-Golay filter, described previously in section 2.5.5. No previous literature was found on accelerometer characterization using these methods.

The accelerometer under test was placed on a platform which was allowed to move in one direction in a plane carefully aligned normal to the gravity vector. The position of the platform was monitored with the capacitive position sensor.

By comparing the applied accelerations, which were calculated from the position data, to the output from the accelerometer, the goal was to obtain the output of the accelerometer as a function of accelerations at different bandwidths, amplitudes and temperatures.

### 3.6.2 Why Dynamic Testing?

If the accelerometer is perfectly aligned to the direction of movement, the modified model equation from appendix D can be reduced to:

$$E(a_i) = O + O_a + Ka_i + K_a a_i + K_2 a_i^2 + K_3 a_i^3 + K_{oq} a_i |a_i| \quad (3.34)$$

when all higher order nonlinear terms than third order are ignored. All of the misalignment terms are eliminated, and will not cause any random bias in the estimated parameters. The cross axis acceleration is constant, given by  $gK_{in}$ , and will bias the estimates as  $gK_{in} a_i$ , where  $n$  is either  $o$  or  $p$  depending on whether the output or the principle axis is directed along the gravity vector. The cross axis terms are small compared to other terms, so this error was ignored during these tests. This model equation has fewer parameters than the ones for multi-position tumble tests given in the equations (3.29) and (3.30), which is an advantage, as the expression for the output as a function of input acceleration is simpler. The output can be plotted as a function of input acceleration in a two-dimensional plot.

In addition to this, when doing a multi-position tumble position tests only a limited number of inputs are available for estimation, as for instance the 48 different positions

used during the experiments in section 3.5. Comparing the acceleration estimated from the measured position to the accelerometer output, will dramatically increase the number of points to use in the estimation, as for instance seen in figure 3.34. The input acceleration is approximately continuous instead of the 25 different input accelerations in the 48 position tumble test.

Another advantage is that the time span needed to make an estimation of the scale factor and the offset is decreased to approximately 10 seconds. As the multi position tumble procedure is relatively time-consuming, especially with a non-automated setup, estimating random variations in parameters is difficult, or impossible for variations with high bandwidths. If the estimation time can be limited to 10 seconds, the ability to track parameter changes would be increased. It is also possible to examine other dynamic phenomena like phase shifts in the accelerometer output.

### 3.6.3 The Experimental Setup

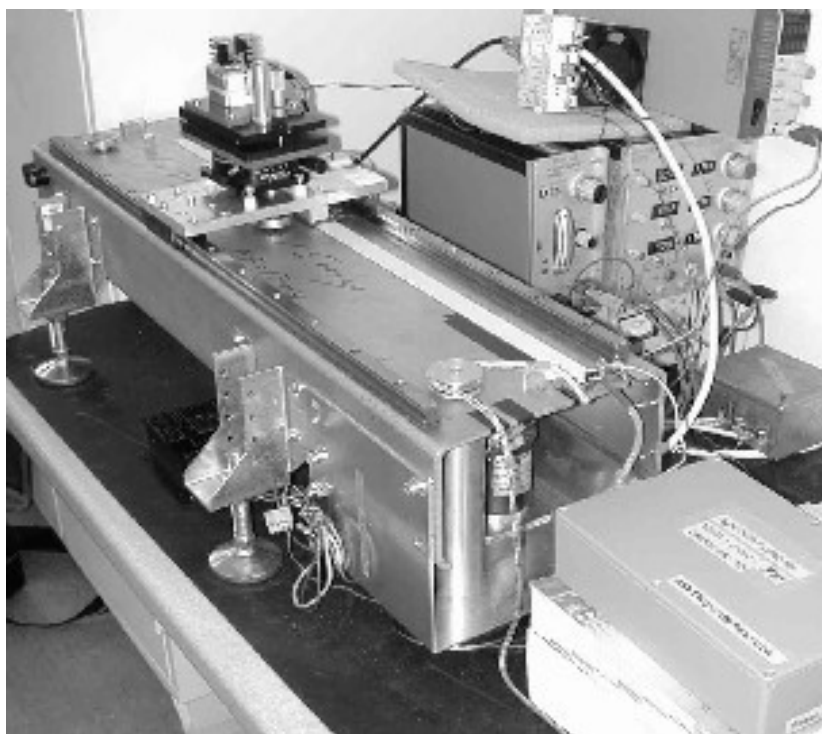


Figure (3.24): Overview of the experimental setup for the dynamic measurements

An overview of the experimental equipment can be seen in figure 3.24. The accelerometer was mounted on a platform which could move back and forth in one direction on a rig. The platform was attached to the rig with rigid ball bearings. The position of the platform was logged using a position sensor.

Equipment to align the input axis of the accelerometer to the direction of movement was mounted on the platform. This consisted of a rotation stage and a tilt platform. A detailed picture of this can be seen in figure 3.28.

The MPT5000 temperature controller was used to regulate the temperature of the accelerometers.

Both the accelerometer and the position sensor outputs were interfaced with a DAQ card from Dspace.

### The rig

The rig was a relatively heavy iron structure standing on metal feet. The feet had adjustable height, making it possible to level the plane of movement normal to the gravity vector. An accurate inclinometer, Sola Lasertronic, with an accuracy of 0.05 degrees was used to make the adjustments. Ideally the rig could have been attached to a heavy structure like a concrete block to keep it fixed in space, but it was considered to be heavy enough to withstand the movements of the platform.

The original plan was to run the platform with an electrical motor controlled by a Maxxon motor controller, but the motor was shown to induce too much vibrations, and possibly magnetic noise. The noise pattern generated by the motor can be seen in figure 3.25.

The transfer of force from the motor to the platform was done with nylon ropes and pulleys, making the system very elastic. This induced a lot of acceleration due to stick-slip-friction, seen as the peaks in figure 3.25. A more rigid mechanical transfer of force could have reduced these peaks.

The solution was to run the platform back and forth by hand. The platform was moved by hand during all the experiments described in this subsection, with approximately sinusoidal movements.

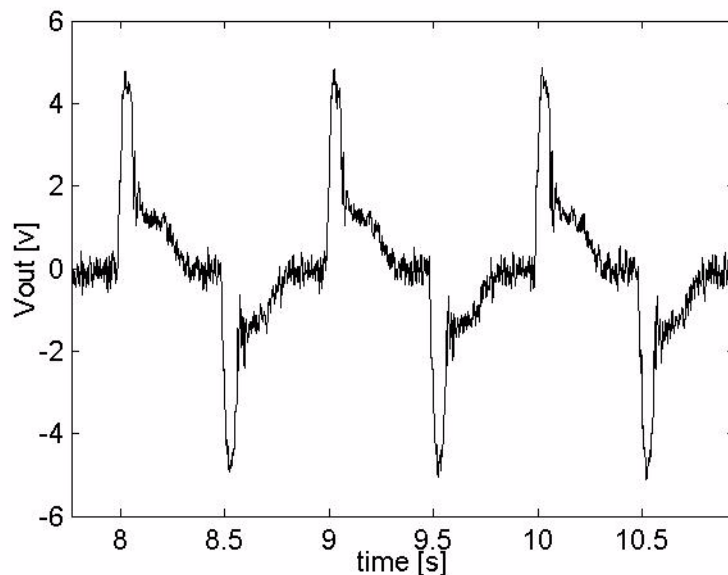


Figure (3.25): Amplified output from an accelerometer when the system was run by the motor.

### The position sensor

The capacitive position sensor used in the experiments was a prototype of a position sensor. The resolution of the sensor was 1  $\mu\text{m}$  (configurable), and the sampling



frequency was set to 833 Hz. The noise floor of the sensor was specified to be less than  $1 \mu\text{m}$ . The sensor had not been optimally calibrated, so the absolute error in position was specified to be maximum  $10 \mu\text{m}$ .

The sensor consisted of two parts, one permanently mounted on the rig, and one part attached the moving platform. The output was analysed with a prototype circuit card, which communicated with the computer through the digital interface of the Dspace card. The interrupt input of the Dspace card was used to signal that new data was available, and then four bytes of data were sent to the Dspace card with the least significant bit first.

Using the Dspace card as shown on figure 3.26, made it possible to synchronize the data from the position sensor and the accelerometer with good accuracy. This was necessary to avoid phase differences between the accelerometer and the position sensor output. The phase delay on the signal conditioning card was negligible in the input area below 12 Hz, and the transport delay in the circuit card analysing the position data was fixed 1.2 ms. The circuit card implementing the analog signal conditioning is described in section 3.3.

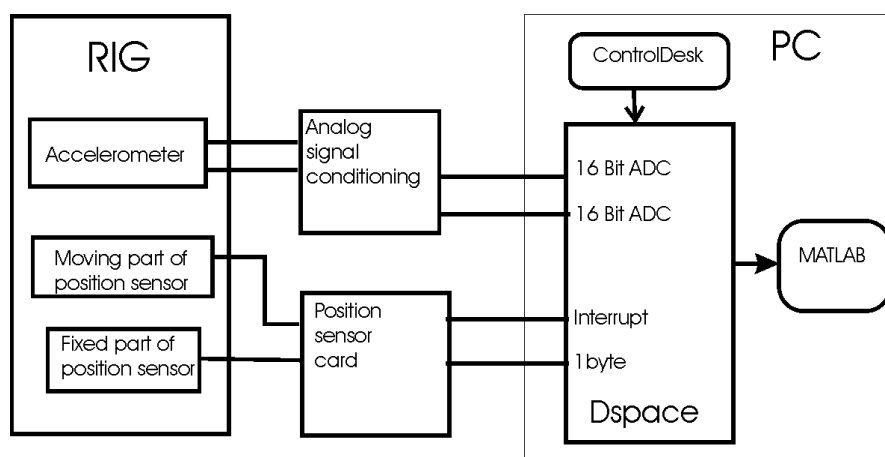


Figure (3.26): The full DAQ setup for the dynamic measurements

### Yaw and pitch adjustment equipment

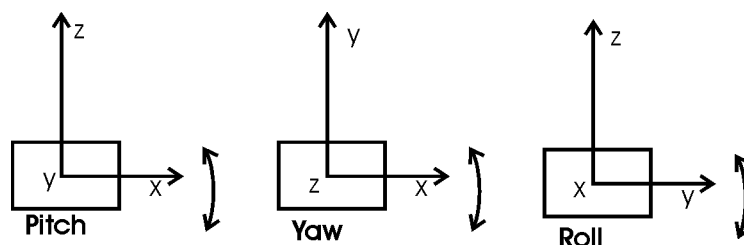


Figure (3.27): Definition of the Pitch- roll- and yaw-axis

In order to adjust the pitch, roll and yaw angles of the input axis of the accelerometer relative to the direction of movement, the setup shown in figure 3.28 was used. Yaw, pitch and roll rotation angles are defined in figure 3.27, where the axis of rotation

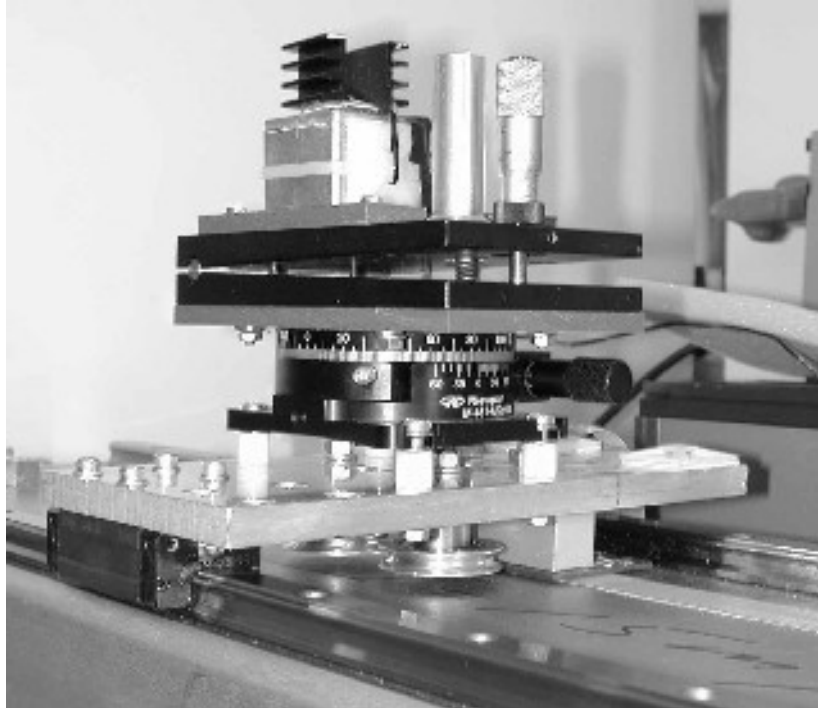


Figure (3.28): The equipment used to align the accelerometer input axis to the direction of movement

points of the plane. The x axis was defined along the direction of movement, while the z axis where defined upwards along the negative gravity vector. The yaw angle was adjusted using the Newport rotation stage 481-A. The absolute accuracy of the device was 1/12 degrees. The relative accuracy for small regions was 0.01 degree. The pitch angle was adjusted with the tilt platform described in section 3.5.2.

### 3.6.4 Vibrational Noise

As the setup was tested, it became clear that the accelerometer picked up some high frequency vibrations. The ball bearings were pointed out as the main source of the vibrations, as the frequency and amplitude of the vibrations increased with velocity. A deformed ball in the bearings may induce such vibrations. An example of the measured vibrations is shown in figure 3.29.

The vibrations during the run shown in figure 3.29 had amplitudes up to 40 mV and frequencies of approximately 200 Hz. This matches the Power Spectral Density (PSD) plot of this run seen in figure 3.30 well, which shows is a peak at approximately 200 Hz. As the vibrations have a frequency above the cutoff frequency, they had been damped by the passive RC filter in the analog signal conditioning card, so that the actual amplitude of the vibrations may have been up to

$$a_v = \frac{40mV\sqrt{1 + (\tau\omega)^2}}{K} = 40mV \cdot \sqrt{1 + \left(\frac{200}{159}\right)^2} \cdot \frac{4g}{V} = 0.26g \quad (3.35)$$

where the nominal scale factor 0.25 V/g was used in the calculation. PSD analysis showed that the vibrations lay in the frequency region between 125 Hz and 300 Hz.

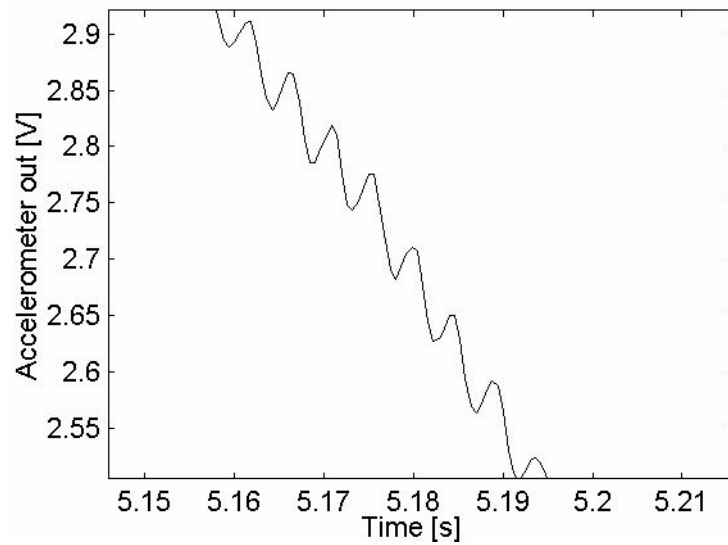


Figure (3.29): Vibrations in the output from the accelerometer

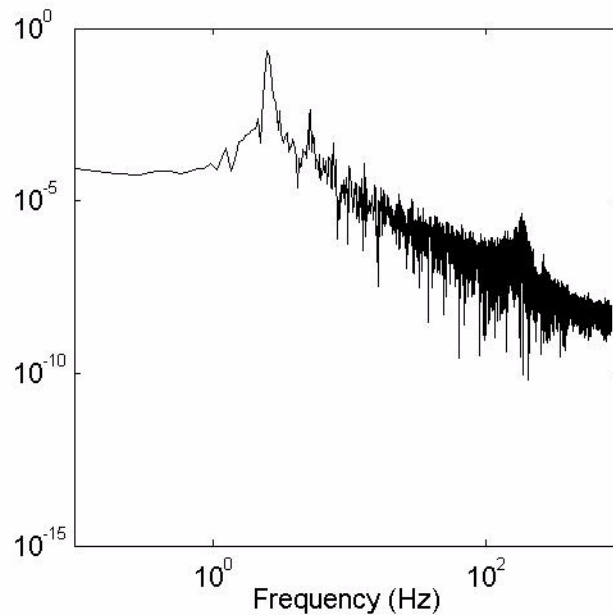


Figure (3.30): PSD of the vibrational patterns in a run with sinusoidal movements with frequency of approximately 2Hz

There were also irregularities at the peaks of the accelerometer output, as shown in figure 3.31. These were probably caused by stick-slip friction, by the small shock every time the platform started moving after stopping at a peak in the sinusoidal movement.

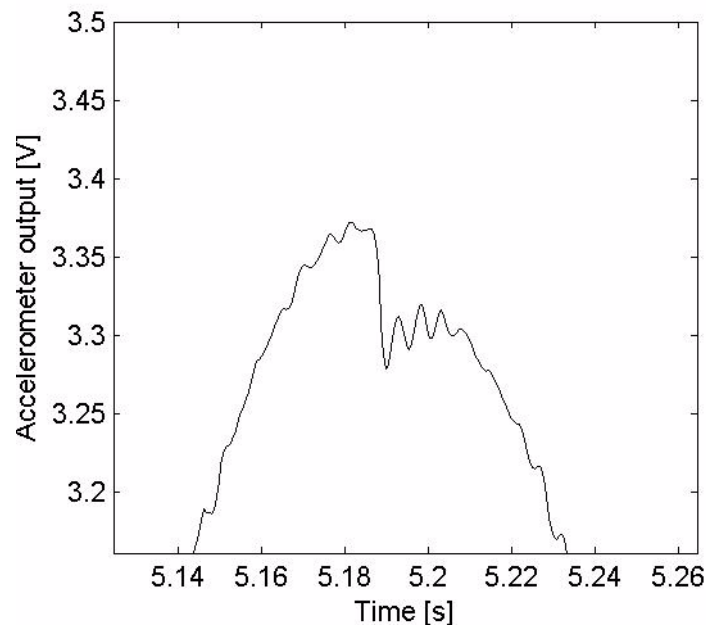


Figure (3.31): Irregularities caused by stick-slip friction.

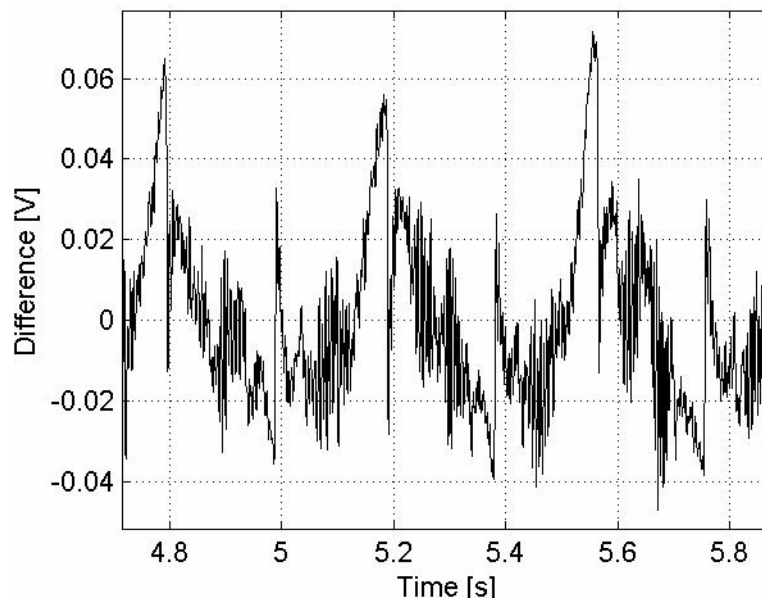


Figure (3.32): The difference between the smoothed output from the accelerometer and the real output

### 3.6.5 Data Analysis

#### Differentiating the position

The data from the position sensor was differentiated twice in order to calculate the acceleration of the system. In order to eliminate the effect of noise, the data had to be lowpass filtered either before or after the differentiation. This was done in two different ways. The first method used the Savitzky-Golay filter, which implements both

smoothing by the method of least squares and differentiation of the data. The second method used a digital lowpass Butterworth filter for smoothing, and numerical differentiation using a Taylor expansion.

The Savitzky-Golay method has previously been described in section 2.5.5. The function named `savgol()` was configured to fit the data to a polynomial of second degree and differentiate the polynomial twice, by passing the parameters `m=2` and `id=2` to the `savgol()` function. As described in section 2.5.5 the amplitude response of the filter depended on the width of the filter window. The tuning of the filter parameters is discussed in a following paragraph.

The second approach used a direct differentiation of the results. Using a Taylor expansion of  $x(t)$  about  $t=t_0 \pm h$ :

$$x(t) = x_{\pm h} = x(t_0) \pm h\dot{x}(t_0) + \frac{h^2\ddot{x}(t_0)}{2} \pm \frac{h^3\ddot{\ddot{x}}(t_0)}{6} + O(h^4) \quad (3.36)$$

Having measured  $x_{\pm h}$ , and reshaping (3.36) to  $x_h - 2x_0 + x_{-h} = h^2\ddot{x}(t_0) + O(h^4)$ , the expression for the second derivative of  $x(t_0)$  is given by:

$$\ddot{x}(t_0) = \frac{x_h - 2x_0 + x_{-h}}{h^2} + O(h^2) \quad (3.37)$$

The position data was first smoothed using a digital third order lowpass Butterworth filter, using the MATLAB filter function `filtfilt()` from the signal processing toolbox, which is a zero phase shift filter function.

### Smoothing the Accelerometer Output

Using lowpass filtering on the accelerometer improved the repeatability of the estimated accelerometer parameters. As given in the specifications in section 2.1.1 the bandwidth of hand movements is approximately 12Hz, thus all signals over this frequency could be considered as noise.

The Savitzky-Golay filter was used for smoothing, giving the parameter `id=0` to the `savgol()` function. A third order digital lowpass Butterworth filter was also applied, using the `filtfilt()` function again as described above.

### Matching the Smoothing and Differentiation filters

The choice of filter parameters influenced the estimations. Thus finding the filter parameters that gave the best estimates was a problem that had to be solved. Tuning the Savitzky-Golay filter meant adjusting the width of the filter window both for the differentiating filter and for the smoother. The cutoff and the order of the lowpass Butterworth filters were also varied to find the best parameters, using the methods described below.

The first method tried was to use the algorithms to find the optimal yaw and pitch angles described in section 3.6.6. The residuals of the cosine-fit were calculated, and

the parameters that minimized the sum of the residuals were chosen. This method was shown to give large variations in the estimated optimal filter parameters.

The second method was to use the methods described below in this subsection to estimate the offset and scale factor. The algorithm was then used on a 5 minutes run, and the parameters that gave the lowest standard deviation in estimated offset and scale factor over the 5 minutes were chosen. Using this method on one set of data gave a window size of 103 points for the differentiating Savitzky Golay filter, and 255 points for the smoothing filter. The frequency responses of the two filters are shown in figure 3.33. Other experiments gave different values. The cutoff frequency of the

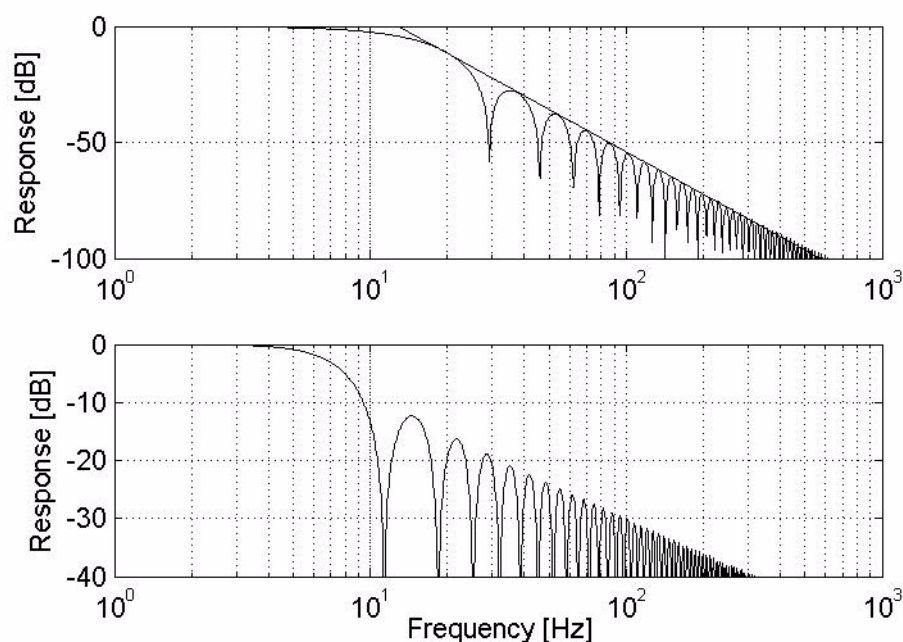


Figure (3.33): Frequency response of the a Savitzky-Golay differentiating filter (top) and a Savitzky-Golay smoother (bottom)

differentiating filter calculated with this method was 13 Hz, which matches the estimated bandwidth of the hand very well. The filter had a -60 dB/decade decay beyond the cutoff frequency. The smoother had the minus 3 dB point at 6.7 Hz and a -20 dB/decade decay.

Combining the Savitzky-Golay differentiating filter and the Butterworth lowpass filter was also tried in order to match the frequency response of the smoother and the differentiator.

The last approach was to combine static and dynamic measurements, where the static results were considered accurate. Knowing that the parameters of the accelerometer should be given values allowed the filter parameters to be adjusted until the values were reached.

## AV-plotting, the AV-plane and line-fitting

Having filtered the position data and the accelerometer output, the next step was to plot the output as a function of acceleration, getting plots like the one shown in figure 3.34.

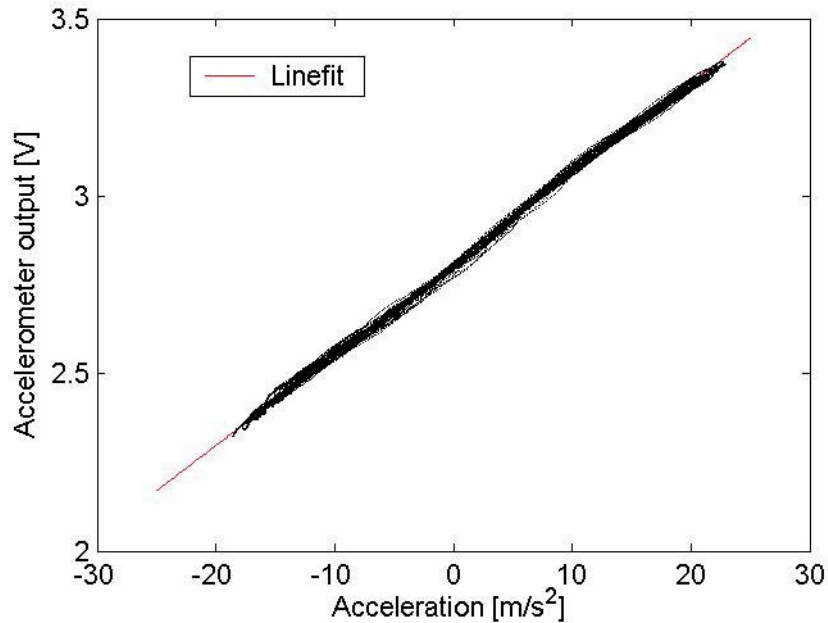


Figure (3.34): Accelerometer output plotted as a function of input acceleration, fitted to a straight line.

The output from the accelerometer was then fitted to a straight line using the method of least squares. The algorithm is implemented in the function `line_test_lsq()` in appendix A, and is based on the methods given by Squires (1968). The scale factor was estimated as  $a$  in the line fit,  $ax + b$ , and the offset as  $b$ . The line fit can be seen as the straight red line in figure 3.34.

The plane on which the graphs in figure 3.34 are plotted will from now on be referred to as the AV-plane, and a line or a function in the plane will be called an AV-plot.

### Effects of stick-slip friction on the line-fit

The irregularities caused by stick-slip friction shown in figure 3.31 were shown to cause errors in the estimations at the peaks of the accelerometer output. The accelerometer output smoothed with the Savitzky-Golay filter and a third order Butterworth filter is shown in figure 3.35. As seen from the graphs, there are differences in how the filters estimate the peak. The differences varied with the selected filter parameters. The consequences of these error are discussed in the results chapter. One possible solution to the problem was to ignore all input data outside a given range, for instance  $a=[-10, 10]$  m/s<sup>2</sup> in figure 3.34.

### Effects of nonlinearities and asymmetries on the line-fit

The estimated parameters in an AV-plot like the one in figure 3.36 can be biased if there are significant nonlinearities or scale factor asymmetries present.

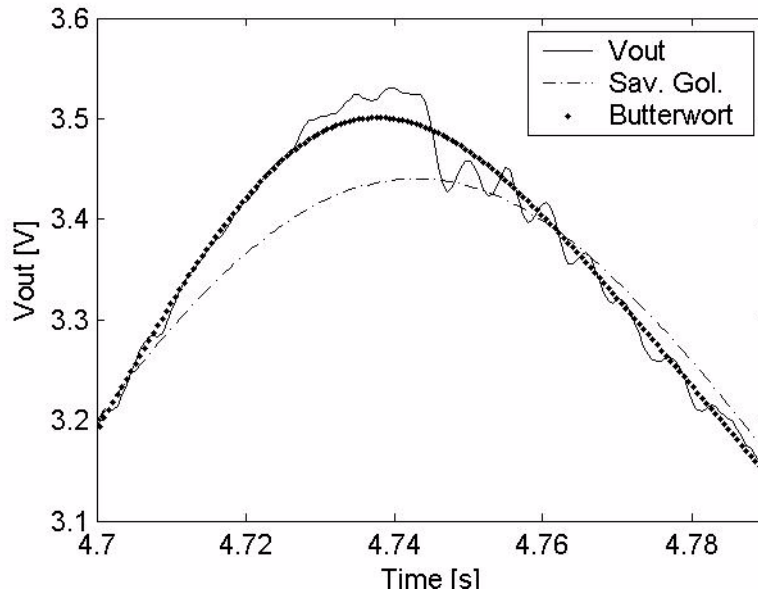


Figure (3.35): The peak of the output from an accelerometer filtered by the Savitzky-Golay filter and a third order lowpass Butterworth filter.

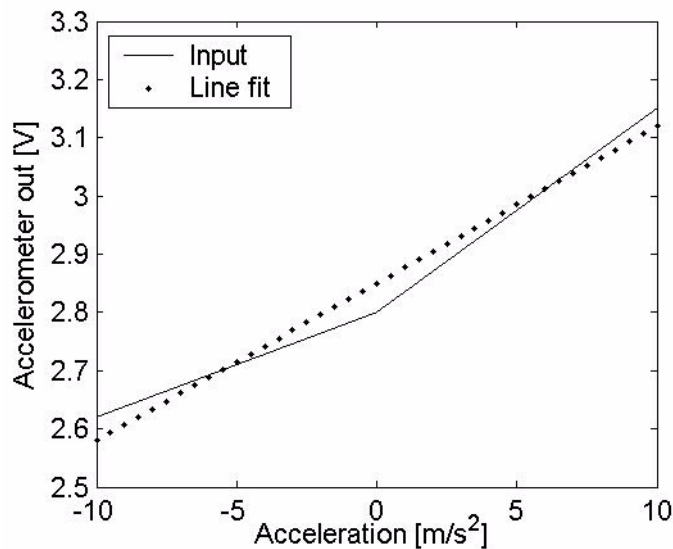


Figure (3.36): Illustration of the effect of scale factor asymmetry on the line-fit

As can be seen from figure 3.36 a scale factor asymmetry would add a bias to the estimated offset. The scale factor would be the average of the two scale factors. If there are more points in the plot on one side of  $a=0$ , the estimated scale factor would also be biased, as all points are weighted equally in the linear fit. Thus either the negative or the scale factor will dominate the estimation. For a repeatable estimation of the offset and scale factor, an equal number of points on each side of  $a=0$  should be used. This can be achieved by using an integer number of periods in the sinusoidal movements.

Nonlinear terms may bias the estimated scale factor, as seen in figure 3.37. In order to get a repeatable estimation of the scale factor, the same input area  $a_i = [-a_0, a_0]$



should be used in every fit. Second order nonlinearities or an asymmetric input area around  $a=0$  can bias the estimated offset as well.

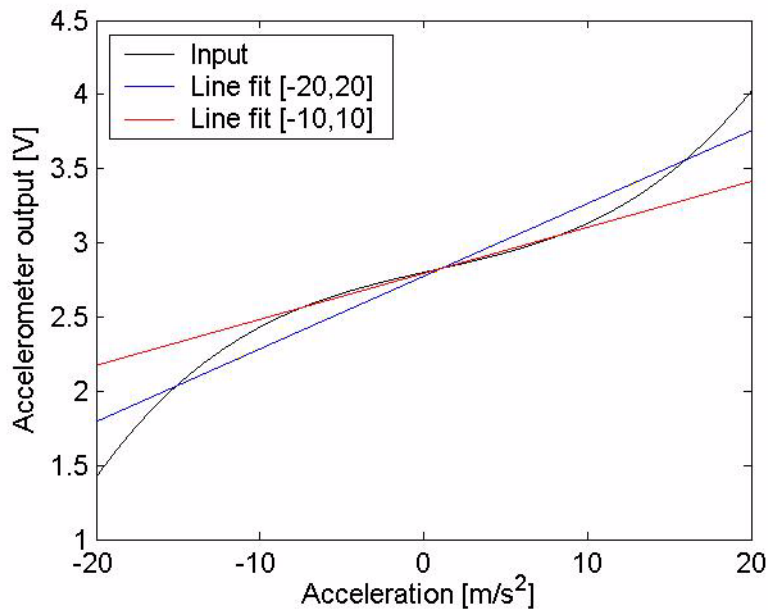


Figure (3.37): Illustration of the effects of nonlinear terms on the line fit

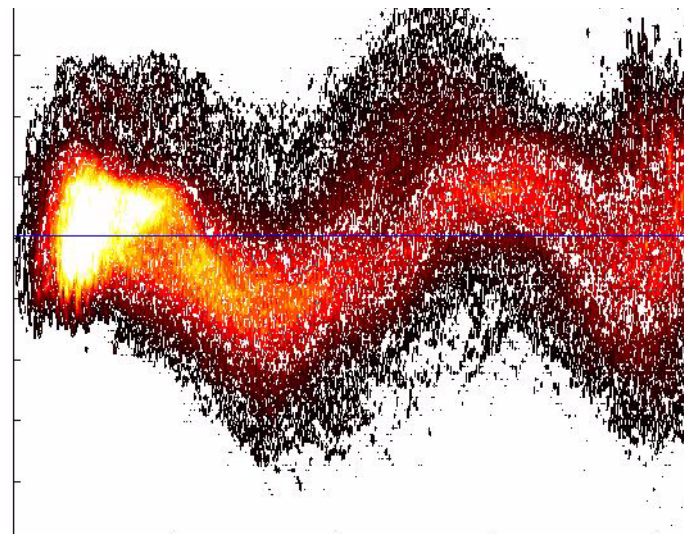


Figure (3.38): The density of points in the AV-plane around the line fit for a 5 minutes run. The plot is rotated  $\theta = -\text{atan}(K)$  in the AV-plane in order to better visualize the results

### Density Plots

In order to make a plot like the one shown in figure 3.38, the AV-plane was divided into a grid with size  $\delta a$  and  $\delta v$ , and the number of points in each cell was counted. The two dimensional array was then plotted as a contour plot as shown in figure 3.38 in order to see the density of points around the line fit. These plots could be seen as a representation of the probability density function, although not normalized. In order to

make the graph more readable, the density function was smoothed in the direction of the V-axis with an equally weighted running average filter for each acceleration step,

The data has been rotated with an angle  $\theta = -\text{atan}(K)$  in the AV-plane around the point  $(a,v)=(0,Of)$ , where  $Of$  is the estimated offset and  $K$  is the scale factor, to get a better visualization of the results.

The line fit is seen as the blue line on the plot. As can clearly be seen from this and other plots, there were indications that the residuals deviated from the line-fit in a systematic manner.

### 3.6.6 Measurements Series

#### Stability of parameters over time

To test the stability of the method over time, experiments lasting several minutes were done. The filtering methods were used as a running window to estimate the development of the offset and scale factor as a function of time. The random walk of both bias and scale factor could be logged over time with this method.

#### Optimal Yaw and Pitch adjustments

A nonlinear cosine fit was used to align the input axis to the direction of movement. The `lsqcurvefit()` function from the statistics toolbox in MATLAB was used to do the fit, and the `regdata()` function listed in appendix A written by Arthur Jutan was used to calculate the standard deviations of the estimated parameters.

#### Yaw adjustments

The pitch angle was set to a constant position. Several runs were done around the initial estimation of the angle, by adjusting the input angle of the rotation stage shown in figure 3.28. Ignoring all cross-axis terms, the estimated scale factor is then given as, and estimating the pitch misalignment to be zero

$$K \cos(\theta - \theta_0) \quad (3.38)$$

where  $\theta_0$  is the ideal angle. Thus fitting the estimated scale factors at each angle to the function given in equation (3.38) gives an estimate of the angle, and the scale factor.

Changing the yaw with an angle  $\theta$  from the ideal angle is the same as a rotation of the AV-plot in the AV plane, with an angle

$$\alpha = \text{atan}(K \cos(\theta)) - \text{atan}(K) \quad (3.39)$$

#### Pitch adjustments

The pitch angle  $\varphi$  was adjusted with the tilt platform, also seen in figure 3.28. The same method, with a nonlinear cosine fit was used to find an estimate of the ideal angle  $\varphi_0$  and the scale factor.

$$K \cos(\phi - \phi_0) \quad (3.40)$$

Having estimated the offset for all angles, and using the estimated scale factor, it was also possible to get an estimate of the gravity constant.

The estimated offset of the accelerometer at an angle  $\varphi$  is given as a function of the offset at zero input,  $O_0$ , and the scale factor as:

$$O(\varphi) = O_0 - Kg \sin(\varphi) \quad (3.41)$$

Using that  $O(\varphi_0) = O_0$ , the gravity constant was estimated fitting the estimated offsets as functions of angle to the function given in equation (3.41), as  $K$  and  $O_0$  were known. This estimation was compared to the value calculated in section 3.5.2, and worked as a test whether the results were sensible or not.

### Effects of nonlinearities and scale factor asymmetry on the estimated pitch angle

When the pitch angle is set to an angle different from the ideal angle, the sensor will measure some component of specific force from the gravity vector. Thus input accelerations in the region  $[-b, b]$  will be measured by the accelerometer as accelerations in the region  $[-b-a, b-a]$  where  $a$  is the projection of the gravity vector onto the accelerometer input axis.

The effect of adjusting the pitch angle  $\phi$  is the same as a linear rotation of the AV-plot in the AV-plane  $\alpha = \text{atan}(K \cos(\phi)) - \text{atan}(K)$  plus a sideways linear transform

$$\alpha' = \alpha - \cos(\phi - \phi_0)g \quad (3.42)$$

In the presence of significant nonlinearities, the input area may be moved to another nonlinear section on the A-V curve, with different dynamic properties. As seen in figure 3.39 the accelerometer output for the adjusted curve in the input area  $a = [-10, 10]$   $\text{m/s}^2$ , has different dynamic properties than the original curve.

Thus an inverse transform of the AV-plot given by  $\alpha = \alpha' + \cos(\phi - \phi_0)g$  may have to be applied before the line-fit is done, which is hard to do when  $\phi_0$  is the constant sought to find. An initial estimation of  $\phi_0$  could be used and then adjusted after a new value of  $\phi_0$  has been estimated by using equation (3.40) and nonlinear estimation by the method of least squares. The procedure then has to be repeated until the estimated value of  $\phi_0$  changes less than a specified number.

The effect of scale factor asymmetry is somewhat like the effect of nonlinearities. The sideways transform of the A-V curve given by equation (3.42) will bias the estimated offset and scale factors if not compensated for.

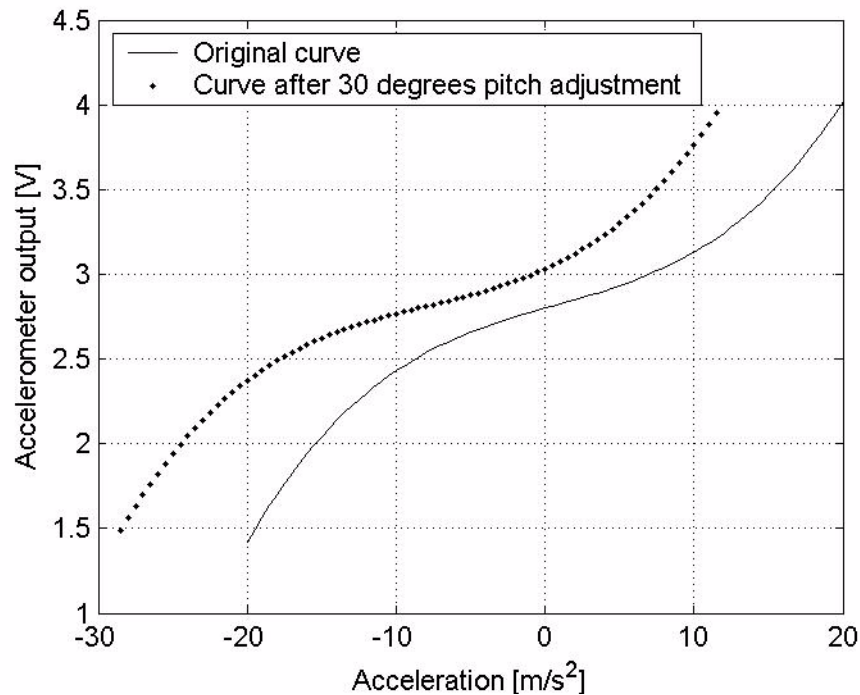


Figure (3.39): Effects of pitch adjustments of the accelerometer input axis on the A-V plot.

### Temperature dependence of Bias and Scale Factor

Some experiments were conducted to check the connection between the accelerometer parameters and temperature using these methods. The MPT5000 was used to control the temperature, while the methods described previously in this section was used to calculate the scale factor and the offset. The accelerometer was kept at constant pitch and yaw angles during the experiments.

## 3.7 Simulations of a Three-dimensional Inertial Tracking System

### 3.7.1 Main principles

A virtual environment was created where the performance of a modelled Inertial Measurement Unit (IMU) could be measured for random movements defined in a local reference frame. The solution to the navigation equations in section 2.2 was implemented numerically and the output from the IMU was used as input to the navigation equations, and the position and the velocity as a function of time was estimated.

The movement of the IMU was defined in the local geographical frame while the rotation and heading was defined in the IMU body frame. In order to calculate the acceleration measured by the IMU due to Coriolis force, and the measured rotation due to the rotation of the earth, the vector transformations described in section 3.7.2 were applied.

The three accelerometers of the IMU was modelled based on results from the static experiments described in section 3.5. The three gyroscopes were mostly modelled as perfect devices as no testing of the performance of gyros had been done.

The aim of the simulations was to see how errors in the estimated accelerometer parameters influenced the performance of the system. In addition it was possible to look at the effects on the performance caused by nonlinear terms and cross-axis sensitivity. Different ways to compensate for these terms were tested.

As the error of an inertial system to some extent depends on the movement of the system, using random movements with bandwidths up to 12 Hz was a mean to get a statistical estimation of the system error as a function of each of the IMU errors.

Any accelerometer model could be used in the simulations, making it a way to test performance of an IMU containing accelerometers with given specifications.

### 3.7.2 Transformation of Vectors

To calculate the rotation and the acceleration measured by the IMU for a given movement, the coordinate transformations shown in figure 3.40 was used

The three orthogonal axes of the IMU make up the *body frame* given the subscript  $b$ , which is defined by the unit vectors  $\underline{i}$ ,  $\underline{j}$ , and  $\underline{k}$  in figure 3.40. The local reference frame was defined as the local geographical frame see section 2.2.8, from now on called the fixed navigation frame or the navigation frame. Vectors and points in this frame were given the subscript  $n$ , and the unit vectors were defined by the vectors  $\underline{ii}$ ,  $\underline{jj}$ , and  $\underline{kk}$  in figure 3.40.  $\underline{ii}$  is fixed pointing in direction north,  $\underline{jj}$  to the east and  $\underline{kk}$  down along the  $\underline{g}_0$  vector. This coordinate system is not well defined on the geographical poles of the earth, but can be by fixing the  $\underline{jj}$  vector along a chosen meridian. The inertial frame,  $i$ , was placed at the centre of the earth and is defined by the unit vectors  $\underline{I}$ ,  $\underline{J}$ , and  $\underline{K}$ .

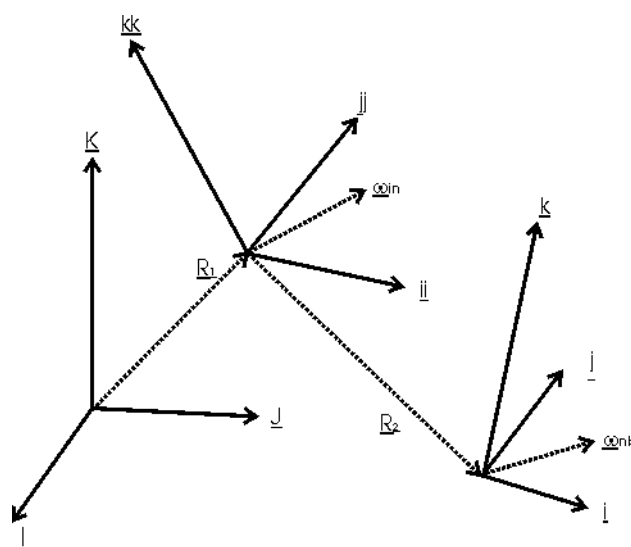


Figure 3.40: The three frames used in the simulations, the inertial frame, the navigation frame and the body frame.

In order to calculate the IMU-output for a given movement in the navigation frame, the movements had to be transformed to the inertial frame, and then back to the body frame, adding coriolis acceleration, specific force due to gravity and other terms.

As seen from figure 3.40, the relative velocity, position and acceleration of the IMU in the navigation frame is defined by the position of the origo of the body frame which by using Coriolis equation defined in equation (2.6) gives equation (3.43).

$$\underline{r}_n = \underline{R}_2, \underline{v}_n = \dot{\underline{R}}_2, \underline{a}_n = \ddot{\underline{R}}_2 \quad (3.43)$$

Applying the Coriolis equation and transforming the vectors to the inertial frame gives

$$\underline{a}_i = \underline{\omega}_{ni} \times \underline{\omega}_{ni} \times \underline{R}_1 + D_n^i \underline{a}_n + 2D_n^i (\underline{\omega}_{in} \times \underline{v}_n) + D_n^i (\underline{\omega}_{in} \times (\underline{\omega}_{in} \times \underline{r}_n)) \quad (3.44)$$

where  $\underline{R}_1$  is the distance from the centre of the earth to origo of the navigation frame.

The rotation vector for a given latitude L is defined by

$$\underline{\omega}_{in} = \omega_0([\cos(L), 0, -\sin(L)])^T \quad (3.45)$$

Using the equations  $\underline{f}_i = \underline{a}_i - \underline{G}(\underline{R}_i)$  and  $\underline{g}_i = \underline{G}(\underline{R}_i) - \underline{\omega}_{ni} \times \underline{\omega}_{ni} \times \underline{R}_1$  gives the expression for the specific force measured by the IMU expressed in inertial coordinates:

$$\underline{f}_i = D_n^i \underline{a}_n + 2D_n^i (\underline{\omega}_{in} \times \underline{v}_n) + D_n^i (\underline{\omega}_{in} \times (\underline{\omega}_{in} \times \underline{r}_n)) - \underline{g}_i \quad (3.46)$$

Thus the specific force expressed in the navigation frame is given by

$$\underline{f}_n = \underline{a}_n + 2(\underline{\omega}_{in} \times \underline{v}_n) + (\underline{\omega}_{in} \times (\underline{\omega}_{in} \times \underline{r}_n)) - \underline{g}_n \quad (3.47)$$

The output from the accelerometers in the IMU is then defined by

$$\underline{f}_b = D_n^b \underline{f}_n \quad (3.48)$$

The size of the term  $\underline{\omega}_{in} \times (\underline{\omega}_{in} \times \underline{r}_n)$  is very small for small values of  $\underline{R}_1$ . For instance  $\underline{R}_1=10$  m when the navigation frame is located at the equator, gives the maximum norm of the term  $(7.3e-5 \text{ rad/s})^2 * 10 \text{ m} = 5e-8 \text{ m/s}^2$ . Thus this term was ignored in all the calculations, as it is also ignored when Titterton and Weston (1997) derives the navigation equations.

Using the results from chapter 2.2.8, the rotation measured by the IMU expressed in inertial coordinates can be given as

$$\underline{\omega}_{bi} = D_n^i (D_b^n \underline{\omega}_{nb} + \underline{\omega}_{in}) \quad (3.49)$$

thus the output measured by the IMU in body coordinates is given as

$$\underline{\omega}_{ib} = D_b^{nT} (D_b^n \underline{\omega}_{nb} + \underline{\omega}_{in}) = \underline{\omega}_{nb} + D_n^b \underline{\omega}_{in} \quad (3.50)$$

### 3.7.3 Generating Random Movements

In order to generate random movements, the position in the navigation frame were calculated as the sum of several different cosine and sine-functions with random amplitudes and phase-shifts as shown in the following equation.

$$\underline{R}_2(t) = \left[ \sum_{k=1}^N a_{(k,i)} \cos(\omega_k t - \phi(a)_{(k,i)}) + b_{(k,i)} \cos(\omega_k t - \phi(b)_{(k,i)}) \right] \quad (3.51)$$

The velocity and acceleration were then calculated by differentiating equation (3.51) with respect to time once and twice. The amplitudes  $a_{(k,i)}$  and  $b_{(k,i)}$  were random constants with the amplitudes limited such that  $x_{(k,i)}(\omega_k)^2 < \alpha \cdot 10 \text{ m/s}^2$ . The constant  $\alpha$  were tuned such that the acceleration stayed within  $10 \text{ m/s}^2$  most of the time, to keep the output from each of the accelerometer in the IMU less than 2 g most of the time. The phase-shifts  $\phi(a)_{(k,j)}$  and  $\phi(b)_{(k,j)}$  where generated as random variables with uniform probability density on the interval  $[0, 2\pi]$ . The frequencies  $\omega_k$  were chosen in the interval  $[0, 12 \text{ Hz}]$ . Care was taken to choose the lowest frequency to be lower than  $2\pi/T$ , where  $T$  was the total time of the interval of the movements. This was done to avoid periodic repetitions of the acceleration during the period.

The heading of the body frame was generated as given in the following equation.

$$\underline{\theta}_{nb}(t) = \left[ \sum_{k=1}^N c_{(k,i)} \cos(\omega_k t - \phi(c)_{(k,i)}) + d_{(k,i)} \cos(\omega_k t - \phi(d)_{(k,i)}) \right] \quad (3.52)$$

The differentiated with respect to time of this heading vector was then the rotation vector  $\underline{\omega}_{nb}(t)$ . The amplitudes  $c_{(k,i)}$  and  $d_{(k,i)}$  where now random constants with the amplitude limited by  $x_{(k,i)}\omega_k < \alpha \cdot \pi/s$  where  $\alpha$  was tuned such that the total rotation stayed within  $\pi/s$  most of the time, as given in the specifications from chapter 2.1.1. The frequencies  $\omega_k$  of the functions were chosen in the interval  $[0, \pi/s]$ .

The direction cosine matrix could then be generated from equation (3.52) as

$$D_n^b(t) = \left( \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \sin(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_3) \\ 0 & \sin(\theta_3) & \cos(\theta_1) \end{bmatrix} \right)^T \quad (3.53)$$

where  $\underline{\theta}_{nb}(t) = [\theta_1, \theta_2, \theta_3]^T$ .

### 3.7.4 Adding Error and Integrating

As the input to the IMU had been calculated by the equations (3.43) to (3.48), the output from the accelerometers were calculated with different error models, as described in the following subsection.

When the algorithm in equation (2.27) was implemented it appeared that the solution was unable to handle large changes in the direction of the rotation vector. Applying the rotation correction term given in equation (2.28) improved the results, but it still gave too much errors. The solution was to solve the equation

$$\dot{C}_b^n = C_b^n \Omega_{ib} - \Omega_{in} C_b^n \quad (3.54)$$

numerically, using a fourth order Runge-Kutta algorithm.

Equation (2.30) was solved using the trapezoidal integration algorithm giving

$$\begin{aligned} \mathbf{v}_{k+1} &= \int_{t_k}^{t_{k+1}} C_b^n(t) f_b(t) dt + \int_{t_k}^{t_{k+1}} \mathbf{g}_0 dt \\ &= \mathbf{v}_k + dt \cdot [C_k^n f_k + C_{k+1}^n f_{k+1} + 2\mathbf{g}_0] / 2 \end{aligned} \quad (3.55)$$

where  $\mathbf{v}_k$ ,  $C_k$  and  $f_k$  are  $\mathbf{v}_n$ ,  $C_b^n$  and  $f_b$  at the time  $t=t_0+k \cdot dt$ .

The correction for the Coriolis term was then done as

$$\mathbf{v}_{k+1}(\text{corrected}) = \mathbf{v}_{k+1} - 2 \cdot dt \cdot \underline{\omega}_{in} \times \mathbf{v}_k \quad (3.56)$$

The calculated velocity was then integrated to get the position, using the trapezoidal algorithm again.

All of these algorithms are implemented in the MATLAB function `inertial_algorithm()` listed in appendix A.

### 3.7.5 Error Models

The accelerometer output as function of the input, was varied to look at the effects of different errors. For instance white noise was added, giving the output from the x-axis of the IMU to be

$$E_x = O^x + K^x f_b^x + \tilde{v}^x \quad (3.57)$$

where  $\tilde{v}$  was white noise defined by its standard deviation. The superscript (<sup>x</sup>) here means the parameter of the accelerometer along the x-axis in the IMU. Offset error was added as  $E_x = (O^x + \delta O^x) + K^x f_b^x$ , and scale factor error as  $E_x = O^x + (K^x + \delta K^x) f_b^x$ .



The complete model used for the x-axis is given in equation (3.58), and similar models were used for the y- and z-axis.

$$E_x = (O^x + \delta O^x) + (K^x + \delta K^x)f_b^x + (K_2^x + \delta K_2^x)(f_b^x)^2 + (K_3^x + \delta K_3^x)(f_b^x)^3 + (K_{io}^x + \delta K_{io}^x)f_b^x f_b^y + (K_{ip}^x + \delta K_{ip}^x)f_b^x f_b^z + \tilde{v}^x \quad (3.58)$$

The principle axis of the accelerometer directed along the  $i$ -axis in figure 3.40, the x-axis accelerometer, was directed along the IMU's z-axis and the output axis along the y-axis. For the accelerometer directed along the  $j$ -axis in figure 3.40, the y-axis accelerometer, the output axis was directed along the IMU's z-axis and the principle axis along the x-axis. The corresponding headings of the IMU's z-axis accelerometer was the principle axis along the IMU's y-axis and the output axis along the x-axis.

The input specific force along a given axis was calculated as

$$f = (E - O)/K \quad (3.59)$$

or as the solution of the third order polynomial given by

$$E = O + Kf + K_2f^2 + K_3f^3 \quad (3.60)$$

This equation has three different solutions. Which solution that gives the non-imaginary correct answer depends on the sign of  $K_2$  and  $K_3$ .

In addition a numerical solution to the equation (3.61) for the input vector

$f_b = [f_b^x, f_b^y, f_b^z]^T$  was obtained using nonlinear least squares estimation. Using this solution was shown to increase the computation time of the algorithms by a factor of 5.

$$\begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix} = \begin{bmatrix} O^x + K^x f_b^x + K_2^x (f_b^x)^2 + K_3^x (f_b^x)^3 + K_{io}^x f_b^x f_b^y + K_{ip}^x f_b^x f_b^z \\ O^y + K^y f_b^y + K_2^y (f_b^y)^2 + K_3^y (f_b^y)^3 + K_{ip}^y f_b^y f_b^x + K_{io}^y f_b^y f_b^z \\ O^z + K^z f_b^z + K_2^z (f_b^z)^2 + K_3^z (f_b^z)^3 + K_{io}^z f_b^z f_b^x + K_{ip}^z f_b^z f_b^y \end{bmatrix} \quad (3.61)$$

### 3.7.6 Simulations

#### Position error as function of accelerometer model errors

Different errors were inserted into (3.61), and the position and the velocity of the IMU was estimated using the equations from section 3.7.4. The output from the simulations was the difference between the estimated position  $\tilde{p}(t)$  and the actual position  $p(t)$ ,

calculated as  $e(t) = \sqrt{(p - \tilde{p})^T (p - \tilde{p})}$ . Starting off with zero, the error typically drifted off like as  $\sim t^2$ , giving that the error at the  $t=T$ , where  $T$  was the length off the simulation, most likely was the maximum error or close to the maximum error. The maximum of  $e(t)$  was used as the output from the simulations. The movements during the simulations was random movements generated by the methods from section 3.7.4, and 10 or 20 different simulations were done for each error model, in order to calculate

the average of the results. The maximum error seen during the 10 or 20 simulations was also logged.

The effects of white noise, offset error and scale factor errors were simulated by setting all other parameters than  $O$ ,  $K$  and  $\tilde{v}$  to be zeros in equation (3.58) and its equivalents for the IMU's y-axis and z-axis. The IMU output was then calculated using equation (3.59).

To look at the effect of uncompensated nonlinear terms, these errors were added to the accelerometer output, but equation (3.59) was used to calculate the IMU output. The output from the IMU was also calculated using equation (3.61), to examine whether it is possible to compensate for nonlinearities if they are exactly known.

The last errors to be added were cross-axis sensitivities. Both the effect when not compensated for and the effect of the errors when compensated for using equation (3.61) was tested.

## **3.8 Simulations of a Local Positioning System (LPS)**

### **3.8.1 Main Principles**

The synchronized data from the position sensor and the accelerometer obtained during the experiments described in section 3.6 made simulations of a LPS possible. The simulated LPS consisted of a primary system, for instance the ultrasound system from Sonitor, and an IMU. A less accurate primary system than the capacitive position sensor was simulated by downsampling the position data and adding white noise. Linear Kalman filters were used to combine measurements from the accelerometers and the position sensor, as described below.

Using real accelerometer output and hand movements in the simulations, made the simulations closer to reality than when using generated paths and accelerometer models, even though limited to one dimension. Only the primary system data had to be generated, and they were based on the position measured by the capacitive sensor.

Three one-dimensional runs were put together to simulate movements in three dimensions. This is in principle the same as a gimbaled system, see section 2.3, where the measured acceleration in the three directions are independent of each other. All errors induced by misalignments and gyro errors are ignored, thus the error in a system with 6 degrees of freedom will be several orders larger. The open source VTK visualization toolkit was used to make three-dimensional visualizations of the results.

The stand-alone accuracy of the inertial system during periods where the primary system was unavailable was also simulated in these experiments.

### **3.8.2 Simulations of an Ultrasonic System**

As described in the previous subsection an ultrasonic system was simulated by using data from the position sensor. The simple model given in equation 2.1.5 was used, where the measurement error was modelled as white Gaussian noise, with the same standard deviation for all positions. The noise was generated using the built-in

MATLAB function `randn()`, which generates normalized white noise. The accuracy and the sampling frequency were changed by changing the standard deviation of the noise and by changing the downsampling interval of the position data.

An specific ultrasonic system from Sonitor has a sampling frequency of 4 Hz and an error of about 30 mm. This is considered a relatively low accuracy system, and the given specifications were used in the simulations as a low accuracy system. The error was decreased down to 1mm and the sampling frequency increased up to 50 Hz to simulate high accuracy systems.

The simulations were based on six different experiments, where the sensor outputs looked similar to the plot shown in figure 3.41. The first 10 seconds the platform was fixed in a constant position, the next 10 seconds with random movements and the last 10 seconds at rest. The purpose of the 10 seconds at rest before and after the movement of the platform was to simulate that the inertial measurement unit was kept in a fixed known position before and after the system is used. The raw position data from the capacitive sensor was fed to the Kalman filter to simulate that the initial and final states of the system were almost exactly known.

The offset and scale factor for the given pitch and yaw angle were calculated using a combination of static and dynamic estimations. The offset was calculated as the mean of the 20 second where the sensors was at rest. The scale factor was calculated with the methods described in section 3.6 from the 10 seconds where the sensor was moving.

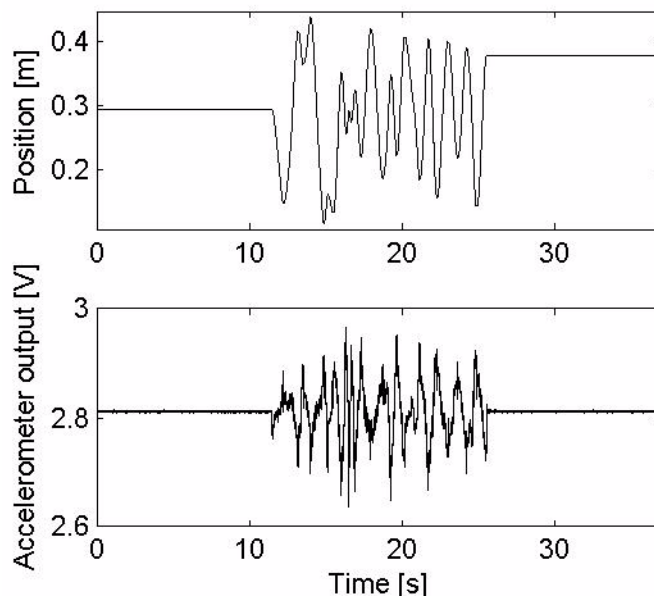


Figure (3.41): Position sensor and accelerometer output as function of time, experiment number two.

### 3.8.3 The Kalman filter and the one-dimensional linear Smoother

The acceleration input as function of accelerometer output was calculated both as the solution of the linear model given in equation (3.59) or the nonlinear model from equation (3.60). The calculated acceleration was fed to a Kalman filter as a scalar control input,  $a$ .

Several different system models were tested. The first model is given in a continuous form in equation (3.62). The scalar variables  $p(t)$  and  $v(t)$  are the position and the velocity of the system, while  $a(t)$  is the input acceleration obtained from the accelerometer.  $q(t)$  is the scalar process noise, defined by the normal probability distribution  $q \sim N(0, \sigma_q^2)$  or the expectation value of the square of the scalar

$$E\{q(t)q^T(t)\} = \delta(t - \tau)\tilde{Q}(t) . \tilde{Q}(t) \text{ is the spectral density of the white noise.}$$

In equation (3.62) the process noise is modelled as white noise.  $z_k$  and  $\underline{x}_k$  are respectively the measurement update and the state vector at the time  $t=k$ .  $w$  is the zero mean white measurement noise, defined by  $w \sim N(0, \sigma_w^2)$  . The measurement noise of the capacitive position sensor was considered small compared to the process noise added to simulate an ultrasonic system, and ignored.

$$\underline{x}(t) = \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} \quad \dot{\underline{x}}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} q(t) \quad (3.62)$$

$$z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \underline{x}_k + w_k$$

Equation (3.63) shows a system model where the accelerometer noise is modelled as a first order Markov process, defined by  $\dot{m}(t) = -\beta m(t) + q(t) = -m(t)/T + q(t)$  where  $q(t)$  is white noise, and  $T$  is the correlation time of the Markov process.

$$\underline{x}(t) = \begin{bmatrix} p(t) \\ v(t) \\ m(t) \end{bmatrix} \quad \dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\beta \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} a(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} q(t) \quad (3.63)$$

$$z_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \underline{x}_k + w_k$$

In equation (3.64) the accelerometer noise is modelled as the sum of a first order Markov process  $m(t)$  and white noise. The noise of the system is now a vector consisting of two different independent stocastic variables,  $q_1$  and  $q_2$ . The stocastic variables are zero mean white noise, defined by their standard deviations. The measurement update of the system is the equal to equation (3.63) and is left out of the equation.

$$\underline{x}(t) = \begin{bmatrix} p(t) \\ v(t) \\ m(t) \end{bmatrix}, \quad \dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\beta \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} a(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \underline{q}(t), \quad \underline{q}(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} \quad (3.64)$$

The expectation value of the squared noise vector is now given by

$$E\{\underline{q}(t)\underline{q}^T(t)\} = \delta(t - \tau)\tilde{Q}(t) \quad (3.65)$$

where  $\tilde{Q}(t)$  is a spectral density matrix.

The continuous-discrete equations given in the equations (3.62), (3.63) and (3.64) were discretized by the MATLAB function `diskretiser()` listed in appendix A to obtain systems on the form

$$\begin{aligned}x_{k+1} &= \Phi x_k + \Lambda u_k + \Gamma v_k \\z_k &= H x_k + w_k\end{aligned}\tag{3.66}$$

and thus the discrete formulation of the Kalman filter listed in equation (2.53) could be used. The `diskretiser()` function is based on algorithms from Bar-Shalom et al. (2001) and Gelb (1974).

As given in the system specifications listed in section 2.1.1, there was no need for real-time knowledge of system state, thus a fixed interval ideal linear smoother was used to increase the accuracy of the filter.

The one dimensional linear discrete Kalman filter from section and the Rauch-Tung-Striebel formulation of the smoother defined in equation are implemented in the MATLAB function `Kalman_Markov()` listed in appendix A.

### 3.8.4 Residual Minimization

The methods of residual minimization previously mentioned in section 2.4.3 was used to model the accelerometer noise. The ideal smoother was used to estimate the position, and the residuals between the estimated and the position measured with the capacitive sensor were calculated. Different noise models and filter parameters could then be tested and adjusted until the residuals reached a minimum.

In addition the method was used to obtain estimation of the higher order nonlinear coefficients in equation (3.60). It turned out that the nonlinear model gave better results than the linear model in equation (3.59). The estimations of the nonlinear coefficients from the static measurements were used as initial guesses, and the parameters were adjusted to minimize the calculated residuals. This could be an other way to estimate the accelerometer parameters. The estimated nonlinear coefficients were probably biased by the initial estimations of the scale factor and offset, in addition to the selected noise models. Thus the higher order nonlinear coefficients estimated in this manner, may just compensate for errors in the estimated linear coefficients.

### 3.8.5 Visualization of the Results

One way to visualize the result of the estimations, was to plot the difference between the position measured with the capacitive sensor, and the position estimated with the Kalman filters. This error as function of the primary system sampling frequency or other parameters, was plotted in two-dimensional plots.

Libraries from the VTK toolkit for C++ was also used for visualization. The toolkit was used to make three dimensional visualizations of the performance of the filters. The results from three one-dimensional estimations was put together to make a three-dimensional run. The visualizations were a useful way to represent the filter

performance in addition to the two-dimensional plots. A screenshot from a run is seen in figure 3.42. The animations could be stored as videos in MPEG-format.

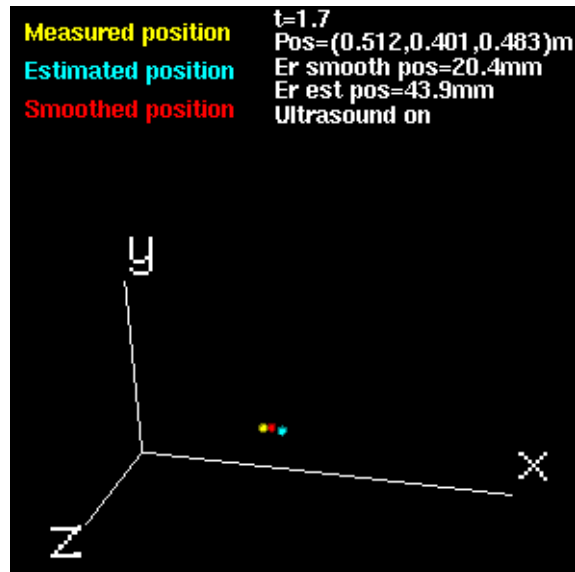


Figure (3.42): Screenshot from an animation

### 3.8.6 Simulations

#### Ultrasonic combined with an inertial system

The ultrasonic system was combined with the inertial measurements with the linear Kalman filter. The different system models were tested for varying primary system accuracies and sampling frequencies to check the performance. Each of the models in the equations (3.62), (3.63) and (3.64) were tested on several different runs, and the parameters in the models were adjusted using the methods from section 3.8.4.

To get an estimate of the performance of each model, the standard deviation of the error defined by

$$e(t) = z(t) - p(t) \quad \sigma_e = \sqrt{\frac{1}{N} \sum_{k=1}^N (e_k - \bar{e})^2} \quad (3.67)$$

and the maximum value of  $e(t)$  was estimated as the ultrasonic system accuracy was varied. These values were logged both as function of the standard deviation of the noise added to simulate the ultrasonic system, and the sampling frequency of the ultrasonic system

Situations where the final and initial state of the system were known with high accuracy were also simulated to see the effects on the performance of the filters.

#### Missing primary system data

During periods where data from the primary system is unavailable, for instance because of obstacles in the line of sight between the ultrasound transmitter and

receivers, the LPS system is totally dependent on the data from the inertial system. This was simulated for short and long periods, by running the Kalman filter without measurement updates for periods of different lengths.

The performance of the filter during these intervals were shown to depend on the movement during the interval, and the state of the system at the start and end of the interval. Thus the starting and ending points were varied, as well as the length of the interval.

The primary system performance was also varied to look at the connection between primary system accuracy and the filter performance during the periods without data from the primary system.

### Stand-alone ultrasonic system

To compare the performance of an stand-alone ultrasonic system and a combined ultrasonic and inertial system, a Kalman filter without input from the accelerometers were constructed. The input acceleration was then modelled as a first order Markov process, and the system model is given by equation (3.68).

The performance of the filter was tested in the same manner as the combined system.

$$\underline{x}(t) = \begin{bmatrix} p(t) \\ v(t) \\ m(t) \end{bmatrix} \quad \dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\beta \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} q(t) \quad (3.68)$$

$$z_k = [1 \ 0 \ 0] x_k + w_k$$

## 4 Results

### 4.1 Results from Experiments using Static Positions

#### 4.1.1 Noise and Start-up Tests

##### Start-up tests

The results of a start-up test conducted with an ADXL105 unit can be seen in figure 4.1. The temperature of the system, the accelerometer inside the aluminium box as seen on figure 3.6, was controlled with the MPT5000 unit to be constant within 0.1 °C. The temperature of the system was stabilized before the power to the accelerometer was turned on. The data was sampled with the DAQ system shown on figure 3.15 at 1.667 kHz.

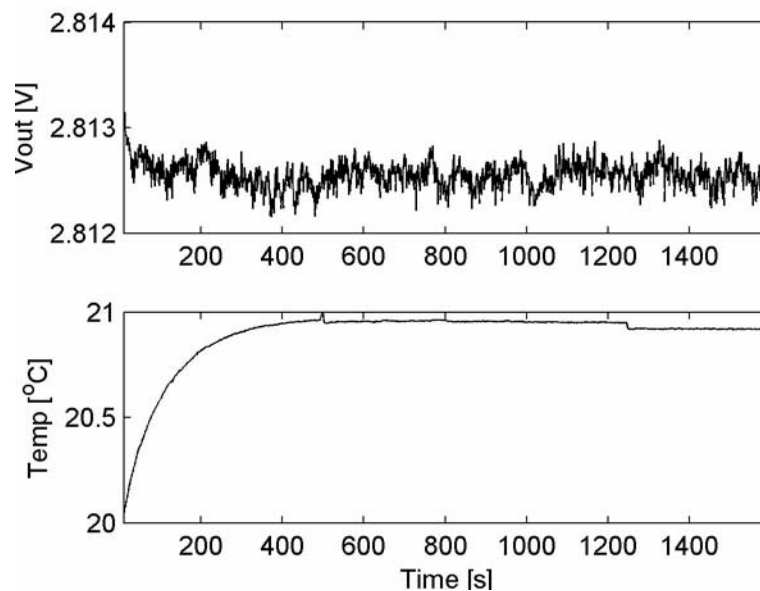


Figure (4.1): Start-up period ADXL105 number three

As can clearly be seen from the graph, there is a transient period. The apparent increase in measured temperature could be because of heat generated by the accelerometer, or because the temperature sensor needs some time to display a stable correct temperature. It may also be a combination of both.

The sensor should probably be turned on about 1000 seconds before the output can be regarded as stable.

##### Random Walk test

The output of the several accelerometers were logged for half an hour in a static position, sampling the data with a frequency of 120 Hz using the DAQ system seen on figure 3.13.



The PSD analysis of the output from the ADXL can be seen in figure 4.2 while a PSD plot of half an hour from the x-axis of the Crossbow CLX02TG3 accelerometer can be seen in figure 4.3. .

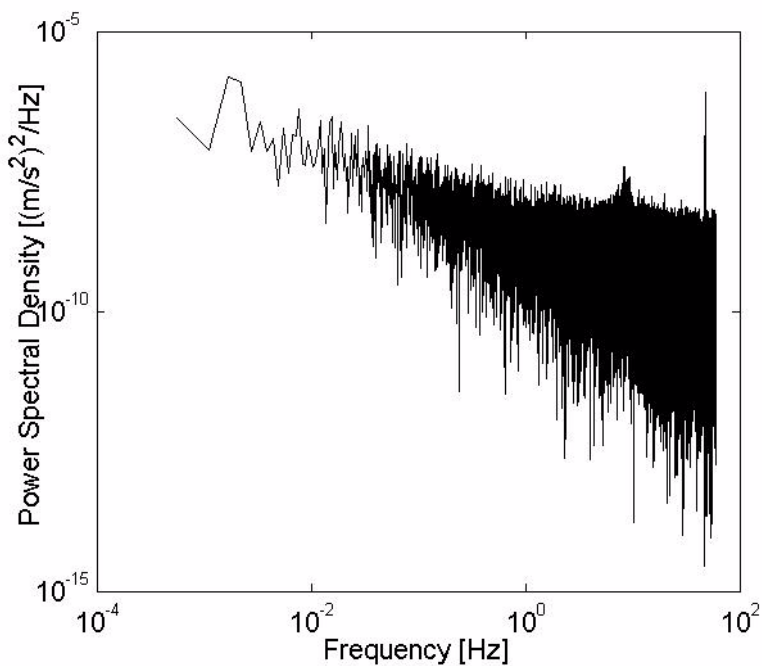


Figure (4.2): PSD plot of half an hour of output from an ADXL105 unit at a static position.

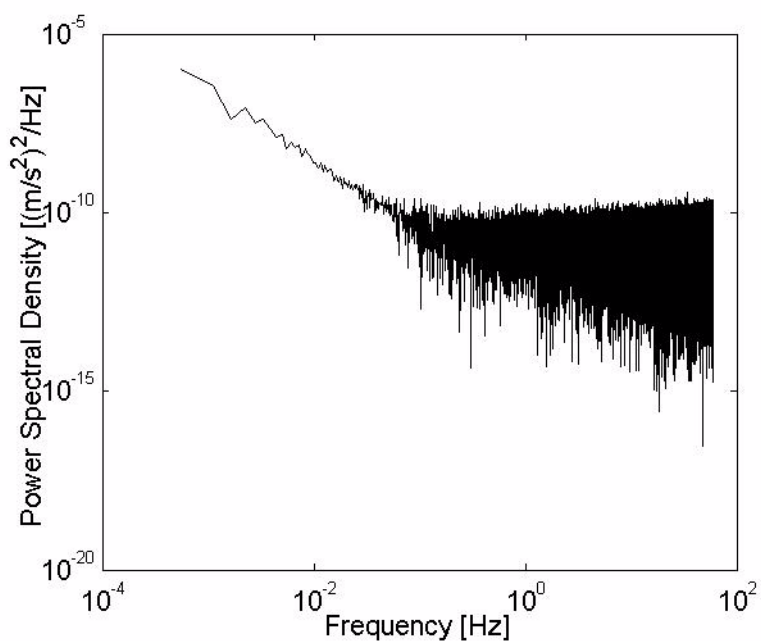


Figure (4.3): PSD plot of half an hour from x-axis of a Crossbow CLX02TG3 accelerometer at a static position.

The result from a random walk test in the time domain can be seen in the following figures, 4.4 and 4.5. The output from the accelerometers in static positions was logged for 5 minutes, at a sample rate of 1.667 kHz using the DAQ system on figure 3.15. An equally weighted running average was calculated for a period of 0.5 seconds and 5 seconds on each side of each point, and plotted as a function of time.

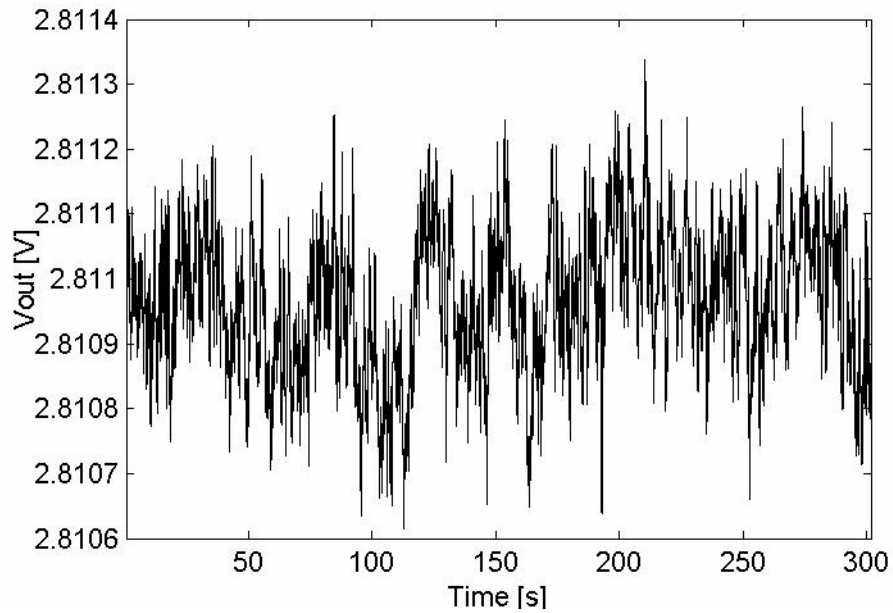


Figure (4.4): Offset /bias random walk for an ADXL105 unit at a static position. Data filtered with 1 second running average.

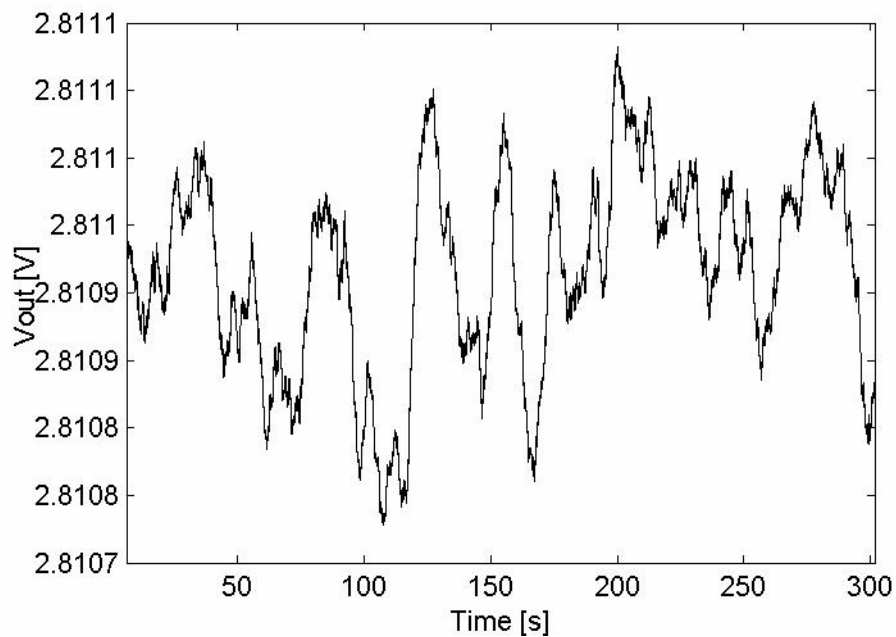


Figure (4.5): Offset /bias random walk for an ADXL105 unit at a static position. Data filtered with 10 seconds running average.

## 4.1.2 Multi Position Tumble Tests

### Multitumble tests about one axis, ADXL105

A multitumble test were done with the ADXL given the number one. The experiment was performed in a room with relatively stable temperature, and measured by the accelerometer temperature sensor to lie between 24 and 25 °C. The data was sampled at 120 Hz, and filtered with a passive lowpass filter with cutoff frequency 47Hz with the DAQ system shown on figure 3.13. The system was rotated about the principle axis, where the principle axis was defined on figure 3.1.

The following tables show the results of the estimation algorithms described in section 3.5.4..

Table (4.1): Offset (O), scale factor (K) and misalignment factor ( $\delta_o$ ) of the input axis relative to  $\theta = 0$  estimated with equal weights on each measurement.

Estimated parameters	O (V)	K (V/m/s <sup>2</sup> )	$\delta_o$ (V/m/s <sup>2</sup> )
Value	2.892	2.5893e-2	-1.9e-4
Standard deviation	3e-3	4e-4	3e-4

Table (4.2): Offset (O), scale factor (K) and misalignment factor ( $\delta_o$ ) of the input axis relative to  $\theta = 0$  estimated with the weighting function given in equation (1.24)

Estimated parameters:	O (V)	K (V/m/s <sup>2</sup> )	$\delta_o$ (V/m/s <sup>2</sup> )
Value	2.8921	2.589e-2	2.1e-4
Standard deviation	7e-4	7e-5	3e-4

The estimated standard deviation is substantially smaller when weighting is applied. The total sum of the residuals increased from 4.5428e-6 to 5.1838e-6, but the sum of the residuals where each residual is weighted, decreased. The rest of the results in this subsection are obtained with least squares algorithms using weighted measurements.

The method of estimating the misalignment angle from section 3.5.3 gave the result  $\theta = -0.43762^\circ$ . This angle has been subtracted from the input angles to eliminate the misalignment in the estimations which results are listed in the tables 4.3-4.8.

Table (4.3): Estimation of offset (O), scale factor (K) and cross-axis sensitivity ( $K_{ip}$ )

Estimated parameters	O (V)	K (V/m/s <sup>2</sup> )	$K_{ip}$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )
Value	2.8921	2.5891e-2	1.2e-7
Standard deviation	1e-4	1e-5	5e-5

Table (4.4): Estimation of offset (O), scale factor (K) and second order nonlinear coefficient ( $K_2$ )

Estimated parameters	O (V)	K (V/m/s <sup>2</sup> )	$K_2$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )
Value	2.8920	2.5891e-2	1.9e-6
Standard deviation	1e-2	4e-5	1e-4

Table (4.5): Estimation of offset (O), scale factor (K), 2. and 3. order nonlinear coefficients ( $K_2$ ,  $K_3$ )

Estimated parameters	O (V)	K (V/m/s <sup>2</sup> )	$K_2$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	$K_3$
Value	2.8920	2.5901e-2	1.9e-6	-1.0e-7
Standard deviation	1e-2	3e-3	1e-4	3e-5

Table (4.6): Estimation of offset (O), scale factor (K), 2. order nonlinear coefficient ( $K_2$ ) and cross-axis sensitivity ( $K_{ip}$ ).

Estimated parameters	O (V)	K (V/m/s <sup>2</sup> )	$K_2$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	$K_{ip}$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )
Value	2.892	2.5891e-2	1.9e-6	7.2e-8
Standard deviation	1.2e-2	3.8e-5	1e-4	2e-4

Table (4.7): Estimation of offset (O), offset asymmetry ( $O_a$ ) and scale factor (K)

Estimated parameters	O (V)	$O_a$	K (V/m/s <sup>2</sup> )
Value	2.8921	-1.0e-4	2.590e-2
Standard deviation	1.2e-4	5.1e-3	5.2e-4

Table (4.8): Estimation of offset (O), scale factor (K) and scale factor asymmetry ( $K_a$ ).

Estimated parameters	O (V)	K (V/m/s <sup>2</sup> )	$K_a$ (V/m/s <sup>2</sup> )
Value	2.8919	2.589e-2	2.2e-5
Standard deviation	5.0e-3	1.2e-5	5e-4

In addition all the parameters, except  $K_{pp}$ , were estimated at once. As mentioned previously in section 1.3.2,  $K_2$  and  $K_{pp}$  are perfectly correlated in a tumble test about the principle axis. The misalignment factor was also estimated.

Table (4.9): Estimation of all parameters.  $K_{oq}$  is the odd second order nonlinear coefficient.

Estimated parameter	Value	Standard deviation
O (V)	2.8921	8e-3
$O_a$ (V)	-1.9177e-5	1e-2

Table (4.9): Estimation of all parameters.  $K_{oq}$  is the odd second order nonlinear coefficient.

$K$ (V/m/s <sup>2</sup> )	2.57e-2	8e-3
$K_a$ (V/m/s <sup>2</sup> )	-7.0e-5	3e-3
$K_2$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	7.3e-6	2e-4
$K_3$ (V/(m/s <sup>2</sup> ) <sup>3</sup> )	-3.4e-6	9e-5
$\delta_o$ (V/m/s <sup>2</sup> )	5.0e-8	3e-4
$K_{ip}$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	5.8e-8	5e-5
$K_{oq}$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	5.1e-5	2e-3

Notice how the estimated standard deviation for each parameter increases as the number of estimated parameters increases.

### One Axis Multipoint tumble test, Crossbow CLX02TG3

Two similar series were performed with the high accuracy Crossbow accelerometer. The DAQ system shown on figure 3.13 was used at a sampling frequency of 120Hz.

The location of the experiment was on an optics table, damping much of the environment vibrations. The temperature in the room was between 22.5 and 24.5 °C. The series were taken without remounting the accelerometer on the rotation stage.

The input axis was the y-axis, while the system was rotated about the z-axis. The misalignment relative to  $\theta = 0$  was estimated to be -0.18871 degrees. The difference in the estimated misalignment between the two series was about 0.01 degrees.

Table (4.10): Estimation of offset, scale factor and misalignment factor.

Measurement series:	Estimated parameters:	O (V)	K (V/m/s <sup>2</sup> )	$\delta_o$ (V/m/s <sup>2</sup> )
SERIES 1	Value	2.51790	8.5893e-2	-2.83e-4
	Standard deviation	1e-5	1e-6	5e-4
SERIES 2	Value	2.51780	8.5898e-2	-2.79e-4
	Standard deviation	1e-5	1e-6	4e-4

Table (4.11): Estimation of offset, scale factor 2. and 3. order nonlinear coefficient. Input compensated for misalignment

Measurement series:	Estimated parameters	O (V)	K (V/m/s <sup>2</sup> )	$K_2$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	$K_3$ (V/(m/s <sup>2</sup> ) <sup>3</sup> )
SERIES 1	Value	2.5181	8.573e-2	-2.4e-6	1.6e-6
	Standard deviation	6e-2	1e-2	6e-4	1e-4

Table (4.11): Estimation of offset, scale factor 2. and 3. order nonlinear coefficient. Input compensated for misalignment

SERIES 2	Value	2.5177	8.583e-2	1.0e-6	6.9e-7
	Standard deviation	4e-2	8e-3	4e-4	1e-4

### Tumble tests about two axes

These tests were performed with the ADXL105 number 3. The DAQ setup in figure 3.15 was used, with a sampling frequency of 1.667 kHz. The temperature was controlled using the MPT5000 temperature controller. The temperature was measured with the accelerometer temperature sensor to lie between 20.78 and 20.92 °C during the experiments.

Table (4.12): Estimation of misalignment of the input axis relative to  $\theta = 0$  e

Measurement series:	Estimated parameters:	$\theta_0$ (°)
SERIES 1	Value	0.283067
SERIES 2	Value	0.73851

Table (4.13): Estimation of misalignment of the input axis relative to  $\varphi = 0$

Measurement series:	Estimated parameters:	$\varphi_0$ (°)
SERIES 1	Value	-0.01337
SERIES 2	Value	-0.12015

Measurements with the Sola inclinometer gave that the two remounts of the sheet were skewed in relation to each other with an angle of 7/20 degrees and 0.1 degrees. Thus the difference between the two estimations of the misalignment was 0.1 for  $\theta_0$  and 0.007 degrees for  $\varphi_0$ .

### Second test

The second test was performed with the same DAQ setup as the first series, but the gain had been adjusted. The gain and  $V_{mid}$  was measured and calculated immediately before the experiment to be  $GP=12.730$ ,  $GN=12.720$  and  $V_{mid}=2.8197$ .  $GN$  here means the gain for input voltages higher than zero, and  $GN$  the gain for input voltages lower than zero. An asymmetry in the gain of the AD620 amplifier was discovered that had not been seen before this point. The measurements were repeated later and the same results were found.

The temperature was measured to lie between 21.04 and 21.22 °C. The corresponding output from the AD592 temperature sensor glued to the lid of the aluminium bow was 23.7°C.

Table (4.14): Estimation of misalignment of the input axis relative to  $\varphi = 0$

Measurement series:	Estimated parameters:	$\theta_0$ (°)
SERIES 1	Value	0.89332
SERIES 2	Value	1.27371

Table (4.15): Estimation of misalignment of the input axis relative to  $\varphi = 0$

Measurement series:	Estimated parameters:	$\varphi_0$ (°)
SERIES 1	Value	0.68593
SERIES 2	Value	0.86465

Tanglement measurements with the inclinometer gave a difference between the two remounts of 1/10 degrees for the principle measurements, and 3/20 degrees for the output experiments. Thus the difference in the estimated misalignments was 0.28 degrees for the estimation of  $\theta_0$  and 0.02 degrees for  $\varphi_0$ .

Table (4.16): Results from all series. O, K,  $K_2$  and  $K_3$   $K_{i0}$  and  $K_{ip}$

Measurement series:	Estimated parameters:	O (V)	K (V/m/s <sup>2</sup> )	$K_2$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	$K_3$ (V/(m/s <sup>2</sup> ) <sup>3</sup> )	$K_{i0}$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )	$K_{ip}$ (V/(m/s <sup>2</sup> ) <sup>2</sup> )
1ST TEST, SERIES 1	Value	2.8125	2.515e-2	1.2e-6	-1.8e-7	-9.9e-7	
	Standard deviation	2e-4	2e-5	2e-4	4e-5	6e-5	
	Value	2.81260	2.5498	-7.8e-7	-1.8e-7		6.6e-7
	Standard deviation	2e-5	6e-6	3e-4	9e-5		6e-5
1ST TEST, SERIES 2	Value	2.8125	2.550	-1.8e-6	2.0e-7	2.5e-7	
	Standard deviation	1e-4	2e-5	1e-4	3e-5	4e-5	
	Value	2.81260	2.5490	-5.4e-7	9.5e-9		-1.1e-6
	Standard deviation	7e-5	8e-6	1e-4	3e-5		1e-4
2ND TEST, SERIES 1	Value	2.8149	2.553e.2	-2.8e-6	1.4e-8	1.4e-6	
	Standard deviation	2e-4	2e-5	1e-4	3e-5	5e-5	
	Value	2.8152	2.551e-2	-3.3e-7	-7.2e-8		-2.7e-6
	Standard deviation	2e-4	2e-5	8e-5	2e-5		6e-5

Table (4.16): Results from all series. O, K, K<sub>2</sub> and K<sub>3</sub> K<sub>io</sub> and K<sub>ip</sub>

2ND TEST, SERIES 2	Value	2.8150	2.553e-2	-3.4e-6	8.1e-9	1.1e-6	
	Standard deviation	3e-4	3e-5	1e-4	2e-5	5e-5	
	Value	2.8150	2.552e-2	-2.4e-6	-1.2e-8		-2.6e-8
	Standard deviation	3e-4	3e-5	1e-4	2e-5		3e-5

#### Four-points tumble tests with temperature variations

The accelerometers were placed inside heat boxes, connected to the DAQ system with a shielded wire.

The results displayed in figure 4.6 was performed with the ADXL105 accelerometer number 2. Fitting the offset as a function of temperature to a straight line by the method of least squares, gave the relationship between offset and temperature was  $-3.24e-4 \text{ V}/^\circ\text{C}$ . The corresponding value for the scale factor was  $1.59e-7 \text{ V}/(\text{m}/\text{s}^2)/^\circ\text{C}$ .

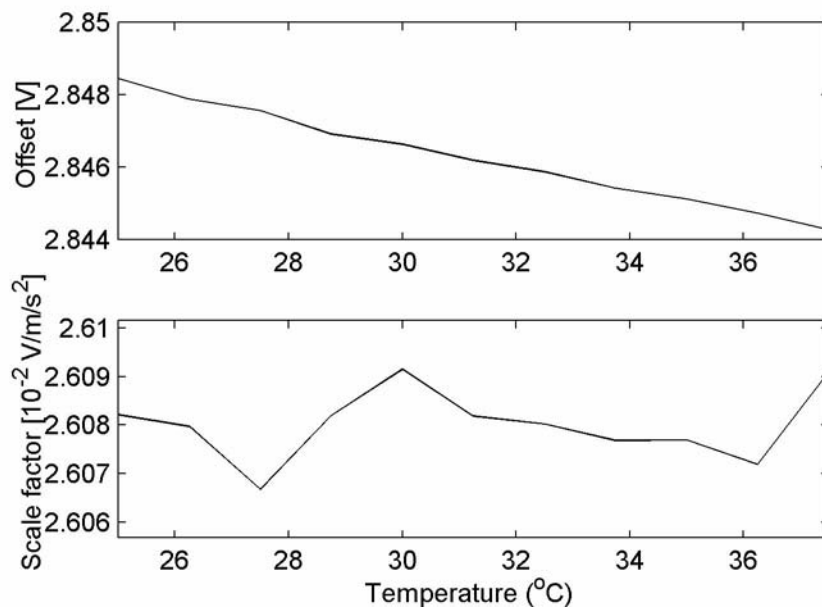


Figure (4.6): Scale factor and offset as function of temperature as results of a 4 point tumble test performed in a heat box

The results displayed in the figures 4.7 and 4.8 were done with the ADXL105 unit number 3.

The first series fitted to straight lines gave that the offset depended on temperature as  $-4.38e-4 \text{ V}/^\circ\text{C}$  and the scale factor as  $-3.18e-6 \text{ V}/(\text{m}/\text{s}^2)/^\circ\text{C}$ . The last series were done in a heat box with cooling functionality. The results was that the offset varied with temperature as  $-2.64e-4 \text{ V}/^\circ\text{C}$  and scale factor as  $2.99e-6 \text{ V}/(\text{m}/\text{s}^2)/^\circ\text{C}$



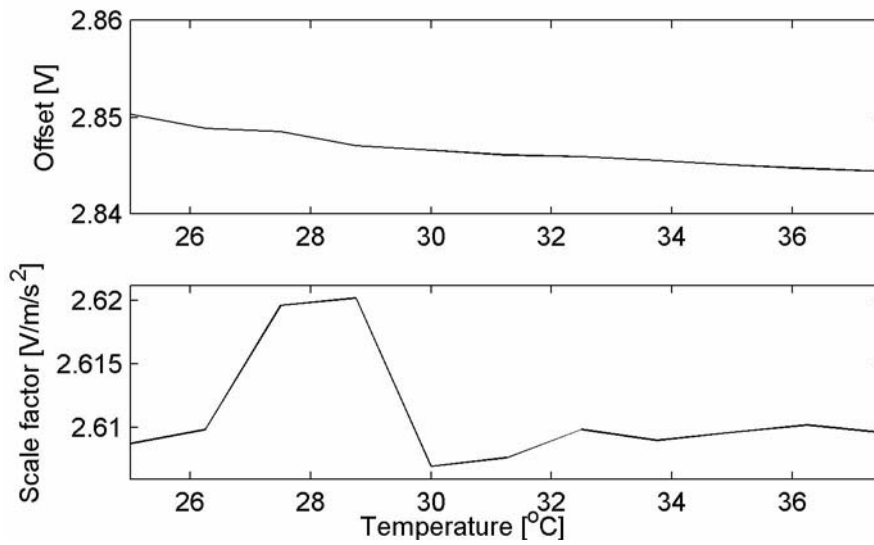


Figure (4.7): Scale factor and offset as function of temperature as results of a 4 point tumble test performed in a heat box

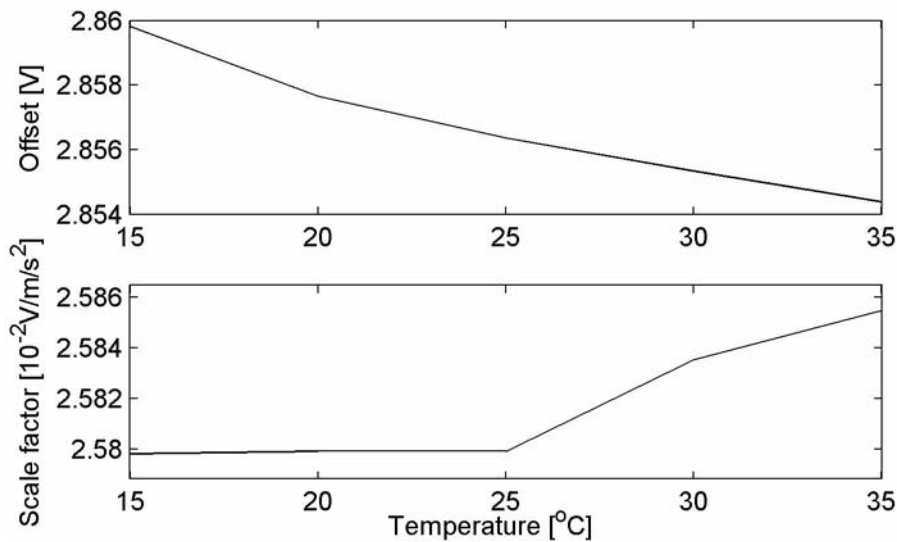


Figure (4.8): Scale factor and offset as function of temperature as results of a 4 point tumble test performed in a heat box

## 4.2 Results of Dynamic Experiments

### 4.2.1 Repeatability Tests

Data from several long experiments with the ADXL105 number three on the rig was used to tune the filter parameters. The offset and scale factor were estimated for subintervals of a fixed length, and the standard deviation of the estimated parameters were then calculated. The goal was to minimize the variation over the total period, and the random walk of the offset and scale factor were considered to be small.

The filter parameters of the differentiating filter and the smoothing filter, and the width of the input area were tuned as described in section 3.6.5.

### Savitzky-Golay differentiator, accelerometer output smoothed with a Savitzky-Golay filter

The least standard deviation in the estimated parameters was found using wide input areas. With the input data in the area  $a=[-27, 27]$   $\text{m/s}^2$ , and a window size for the differentiating filter of 85 points or 51 ms the results was as shown in the following figure. The window size of the smoothing filter was 232 points, or 139.2 ms. The standard deviation in the offset was 0.51 mV and the standard deviation in the scale factor was  $3.4\text{e-}5$   $\text{V/m/s}^2$

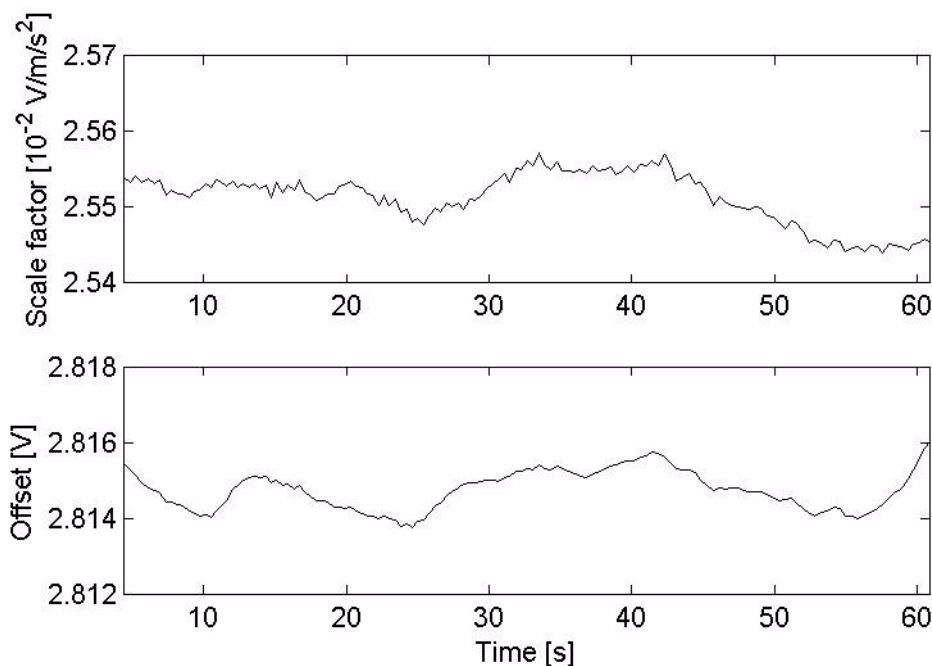


Figure (4.9): Acceleration calculated with the Savitzky-Golay filter, window size 51ms. The accelerometer output filtered with Savitzky-Golay smoother, window size 139.2ms.

The values of the estimated offset and scale factor when using these filter parameters deviated from the parameters obtained from the static measurements, see section 4.1. The mean values of the offset was 2.8147V, and the mean value of the scale factor was  $0.025512$   $\text{V/m/s}^2$ . The corresponding values taken from table 4.15 with static experiments is offset=2.8152V and scale factor= $2.553\text{e-}2$   $\text{V/m/s}^2$ .

This was combined with a static measurement of the accelerometer output for the same pitch and yaw angle, which was 2.8156V. The yaw angle was set to be  $90^\circ$ , while the optimum yaw angle was estimated to be about  $87^\circ$ , see table 4.21. In addition to seeking the lowest possible standard deviation in the estimated parameters over the one minute period, the estimated parameters should also be close to  $O=2.8156\text{V}$  and  $K=0.025512$   $\text{V}/(\text{m/s}^2) * \cos(3^\circ)=0.02548$   $\text{V/m/s}^2$ .

The offset and scale factor as a function of different filter parameters are shown in the following table.

Table (4.17): Standard variation and mean of the estimated accelerometer offset and scale factor as function of filter parameters of the Savitzky-Golay differentiating filter, and the Savitzky-Golay smoother.

Differentiating filter window size (ms)	Smoothing filter window size (ms)	Acceleration input region (m/s <sup>2</sup> )	Mean Offset (V)	Standard deviation Offset (V)	Mean Scale factor (V/m/s <sup>2</sup> )	Standard deviation scale factor (V/m/s <sup>2</sup> )
57	155.6	[-12.5, 12.5]	2.8155	9.e-4	2.626e-2	2e-4
27	137.4	[-15, 15]	2.8150	1e-3	2.551e-2	2e-4
51	160.2	[-25, 25]	2.8155	5e-4	2.547e-2	3e-5
21	21	[-27, 27]	2.8147	2e-4	2.559e-2	3e-5

#### Savitzky-Golay differentiator without smoothing the accelerometer output

As seen from the results in table 4.18 no set of parameters was found that gave an offset close to 2.8156V

Table (4.18): Standard variation and mean of the estimated accelerometer offset and scale factor as function of filter parameters of the Savitzky-Golay differentiating filter. Accelerometer output not smoothed.

Differentiating filter window size (ms)	Acceleration input region (m/s <sup>2</sup> )	Mean Offset (V)	Standard deviation Offset (V)	Mean Scale factor (V/m/s <sup>2</sup> )	Standard deviation scale factor (V/m/s <sup>2</sup> )
24	[-10, 10]	2.8084	1e-3	2.514e-2	2e-4
18	[-15, 15]	2.8096	1e-3	2.521e-2	1e-4
15	[-25, 25]	2.8128	5e-4	2.564e-2	3e-5
12	[-27, 27]	2.8130	4e-4	2.533e-2	5e-5

#### Savitzky-Golay filtering and Butterworth lowpass filtering

Using a window size of 19.8 ms for the differentiating Savitzky-Golay filter, and a cutoff frequency of 11Hz for the smoothing Butterworth filter, did not give the desired offset value, but gave a very good repeatability for the scale factor, 2.7e-5V/m/s<sup>2</sup>. The input area was chosen to be [-25, 25] m/s<sup>2</sup>.

#### Differentiation using a Taylor series expansion combined with Butterworth lowpass filtering of the accelerometer output

The estimated ideal cutoff frequencies to minimize the variation in the scale factor and offset were 12 Hz for the filter that lowpass filtered the position data, and 9 Hz for the

filter that lowpass filtered the accelerometer output. The results are shown in figure 4.10. The standard deviation in the calculated offset was 0.49 mV and the standard deviation in the estimated scale factor was  $2.5 \times 10^{-5} \text{ V/m/s}^2$ . Choosing the cutoff frequencies to be 16 Hz and 7 Hz gave a mean estimated offset 2.8140 V scale factor of  $2.540 \times 10^{-2} \text{ V/m/s}^2$ , which are close to the desired values.

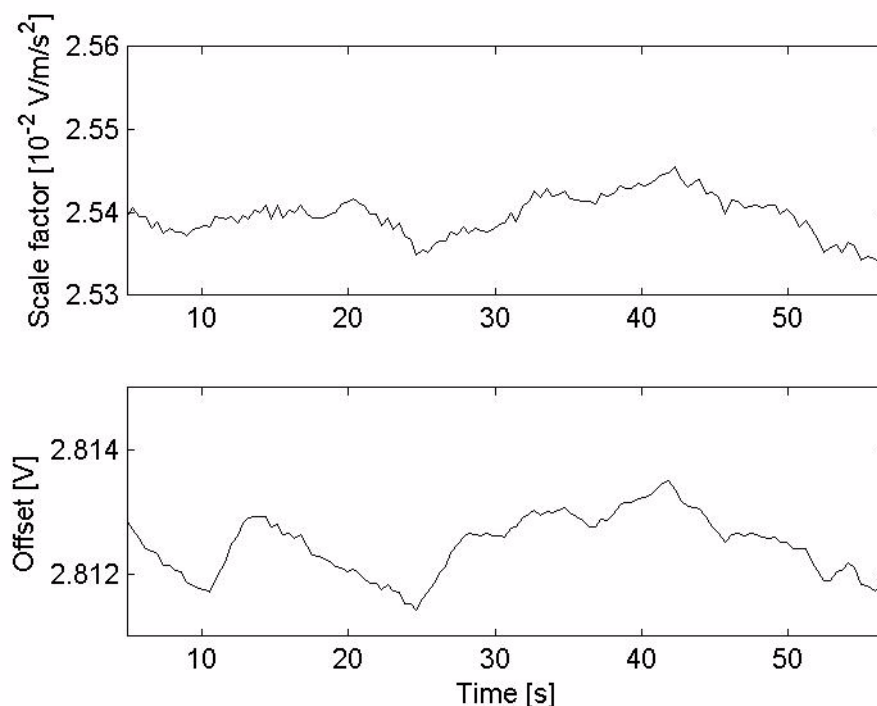


Figure (4.10): Direct differentiation using a Taylor series expansion of position data, lowpass filtered with a Butterworth filter with cutoff frequency of 12 Hz. Accelerometer output lowpass filtered with cutoff frequency 9 Hz.

## 4.2.2 Results from Combined Static and Dynamic Experiments

### 4.2.3 Yaw and Pitch Alignment

#### Results from a series done with gain 36.4 and input interval [-g, g]

The yaw alignments is shown in figure 4.11. The data was differentiated with the Savitzky-Golay filter with a window size of 105 points, or 62.6ms. The data was smoothed with a Savitzky-Golay smoother with window size 255 points or 153ms.

The three different series gave the results shown in table 4.19

The pitch alignment series were done with the same filter parameters as the yaw series, and are shown in figure 4.12. The results are displayed in table 4.20

Table (4.19): Results from estimations of the ideal yaw angle, input accelerations in the region [-g, g]

Series number	Maximum angle	Estimated standard deviation of the maximum angle	Scale factor	Estimated standard deviation of the scale factor
1	84.38	0.1	2.5709 e-2	5e-06
2	84.41	0.1	2.5728 e-2	6e-06
3	84.43	0.1	2.5707 e-2	4e-06

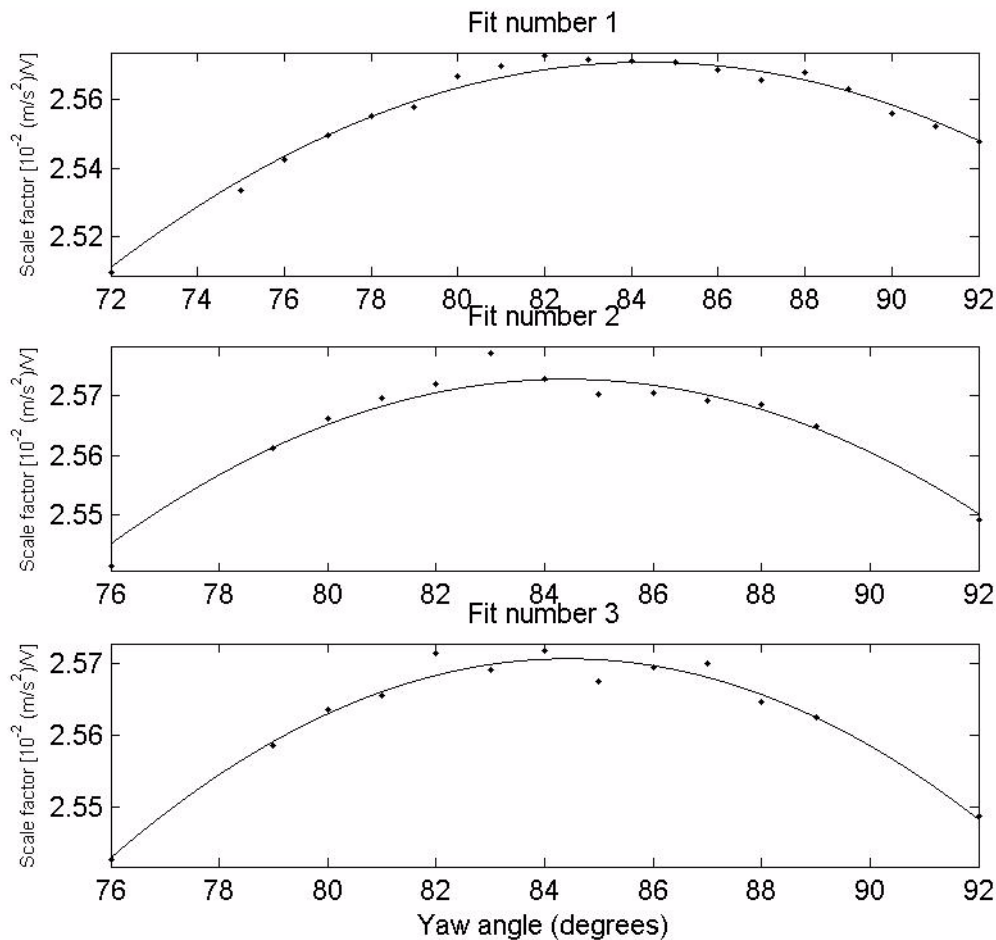


Figure (4.11): Results from three different estimations of the optimal yaw angle. Input accelerations in the region [-g, g].

### Results from series with input accelerations up to [-2.5g, 2.5g]

The results shown in the tables 4.21 and 4.22 were obtained using differentiation with Taylor expansion, as it was shown to give the best repeatability for the scale factor. The accelerometer was remounted on the rig. The angles therefore have no connection to the ideal angles in tables 4.19 and 4.20.

Table (4.20): Results from estimations of the ideal pitch angle. Input accelerations in the region [-g, g].

Series number	Maximum height (mm)	Estimated standard deviation, maximum height (mm)	Scale factor (V/m/s <sup>2</sup> )	Estimated standard deviation, scale factor (V/m/s <sup>2</sup> )	g (m/s <sup>2</sup> )
1	6.4393	0.39623	0.025681	4.2138e-06	10.303
2	8.086	0.23303	0.025728	2.8542e-06	9.9617

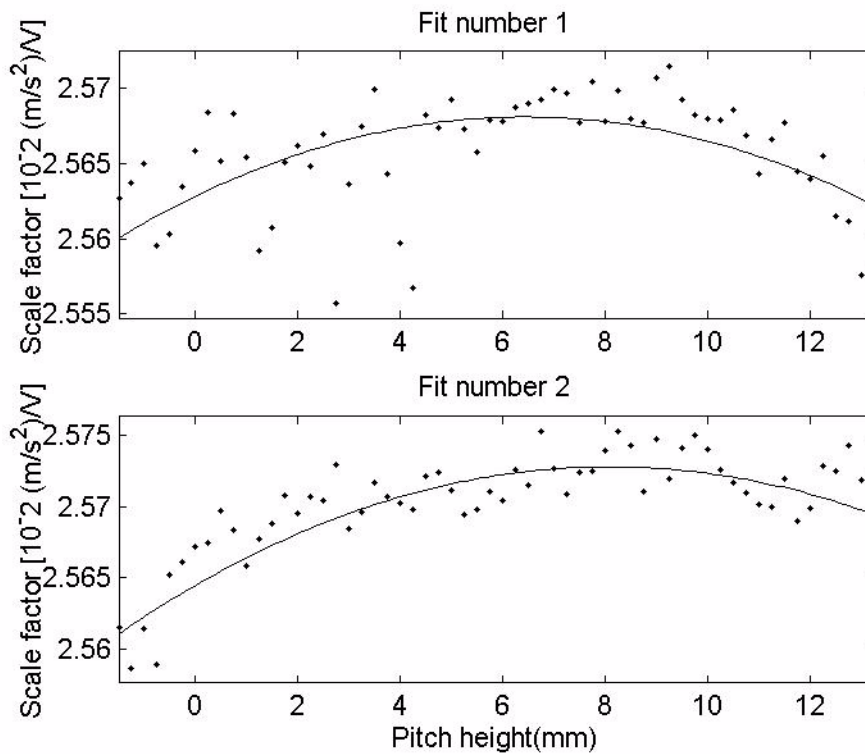


Figure (4.12): Results from two different estimations of the ideal pitch angle. Input accelerations in the region [-g, g]

Table (4.21): Results from estimations of the ideal yaw angle. Input accelerations in the region [-2.5g, 2.5g]

Series	Maximum angle	Standard deviation maximum angle	Scale factor	Standard deviation scale factor
1	87.05	4e-2	2.5449 e-2	5e-06
2	86.65	0.1	2.5430 e-2	1e-05
3	87.16	9e-2	2.5455 e-2	9e-06

Table (4.22): Results from estimations of the ideal pitch angle. Input accelerations in the region [-2.5g, 2.5g].

Series	Maximum height (mm)	Standard deviation maximum height (mm)	Scale factor (V/m/s <sup>2</sup> )	Standard deviation scale factor (V/m/s <sup>2</sup> )	g (m/s <sup>2</sup> )
1	6.1	0.3	2.5373e-2	3e-06	9.7497
2	7.6	0.3	2.5348e-2	3e-06	10.2519

#### 4.2.4 Results from Combined static and dynamic Measurements

During the pitch alignments presented in the previous subsection, the measurements were combined with static measurements. At each input angle, the output from the accelerometer was logged for a period of 15 seconds. Ideally, the average of this measurement should be the same as the offset estimated from the dynamic measurements. This was not the case, the offset estimated from the line fit was in general about 3 mV lower than the offset estimated from the static measurement.

As described in section 3.6.5 both second order nonlinearities and scale factor asymmetry can bias the estimated offset. In both cases, the scale factor will be asymmetric about  $a=0$ . As shown in the following figure, this was the case. There was a significant gap between the scale factor estimated from positive and negative accelerations.

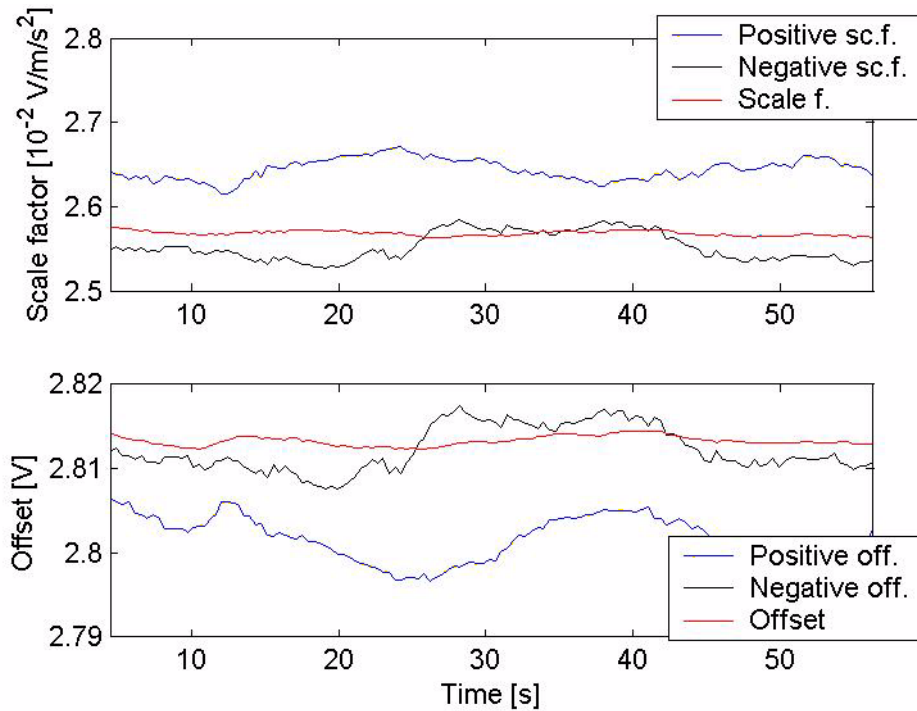


Figure (4.13): Difference between parameters estimated from accelerations between [-25, 0] m/s<sup>2</sup> and [0, 25] m/s<sup>2</sup>.

This asymmetry vanished if the input area was set to be  $a=[-10, 10] \text{ m/s}^2$ , indicating that the asymmetry originated from the higher accelerations and not from scale factor asymmetry. The apparent asymmetry may be a real asymmetry caused by second order nonlinearities, or the results of the stick-slip friction described in section 3.6.5.

Thus the output ideal pitch angle can be calculated from the static measurements and the value of the offset calculated in the static measurements. The ideal pitch height was then calculated to be 6.11mm.

#### 4.2.5 Density Plots

Ideally, the contours on the plot in figure 4.14 should have followed the blue line, which represents the line fit. They do not, and cause the effects discussed in section 4.2.4. As seen in figure 4.15, which is a cross-section of the density plot parallel to axis number two at the line 0 on axis number one, the data are centred about the correct value for  $a=0$ , and the calculated mean is also 2.815 V, as it should be. Also seen in figure 4.14 is the fact that the concentration of points is highest at the peaks, which implies that cutting the input area, means cutting a lot of the data and thus increasing the uncertainty in the plots.

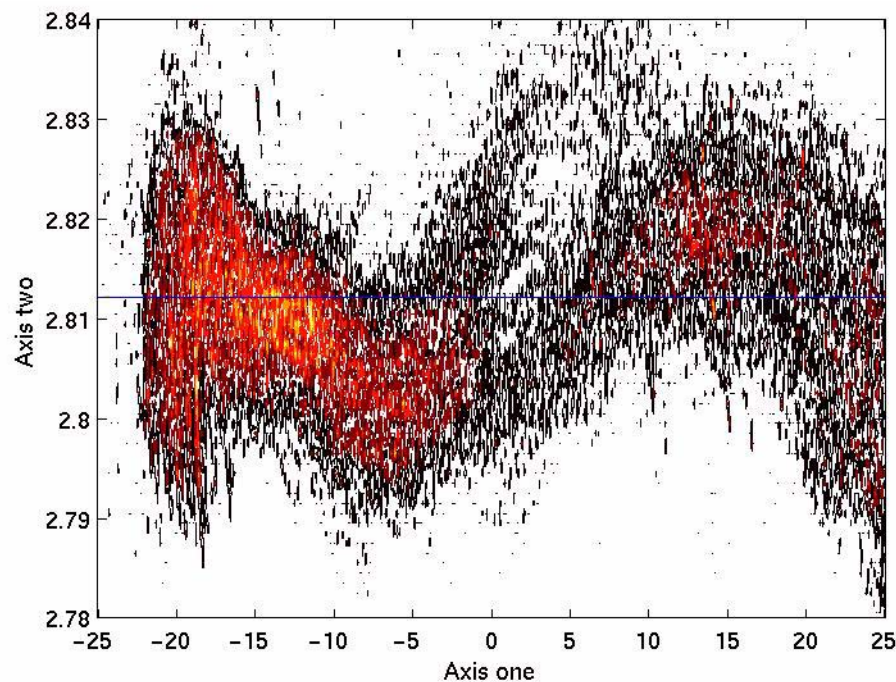


Figure (4.14): Density plot of a 1 minute long experiment. The blue line is the linear fit. Units of the axes is  $V+m/s^2$  and is without any physical interpretation.

The plot in figure 4.16 is a corresponding density plot from the Crossbow accelerometer. As seen from the figure, there are indications of hysteresis due to phase delay or other phenomena.



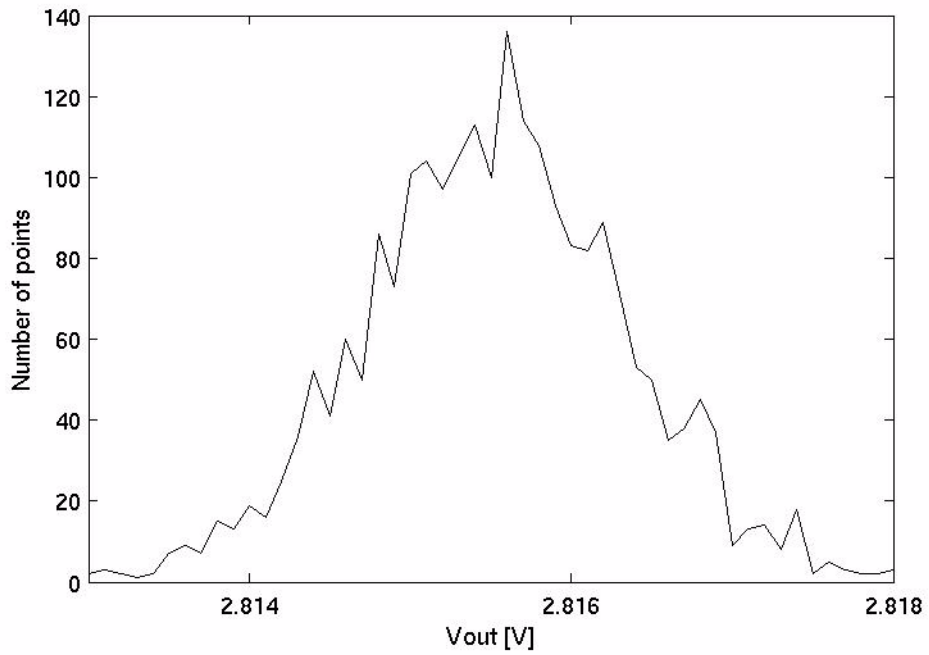


Figure (4.15): section from the plot figure 4.14 at axis1=0.

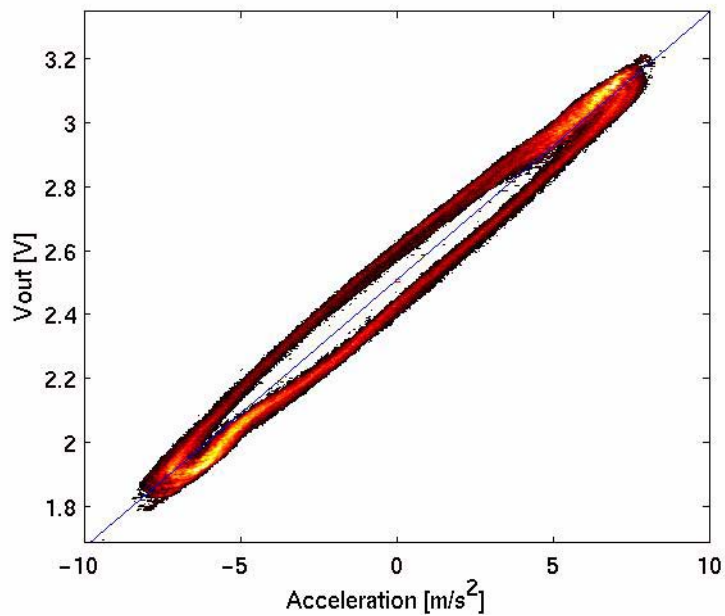


Figure (4.16): Density plot for a five minutes long experiment with the Crossbow accelerometer.

#### 4.2.6 Temperature Experiment

Two series were done with the ADXL105 unit number three. The results are shown in the figures 4.17 and 4.18.

The *first* series gave the following results:

Scale factor dependency on temperature= $(3.4e-6 \pm 2e-6) \text{ V}/(\text{m}/\text{s}^2)/^\circ\text{C}$

Offset dependency on temperature= $(-2.3e-4 \pm 4e-5) \text{ V}/^\circ\text{C}$

The *second* series gave the results:

Scale factor dependency on temperature: $(1.6e-6 \pm 6e-7) \text{ V}/(\text{m}/\text{s}^2)/^\circ\text{C}$

Offset dependency on temperature:  $(2.5e-4 \pm 2e-5) \text{ V}/^\circ\text{C}$

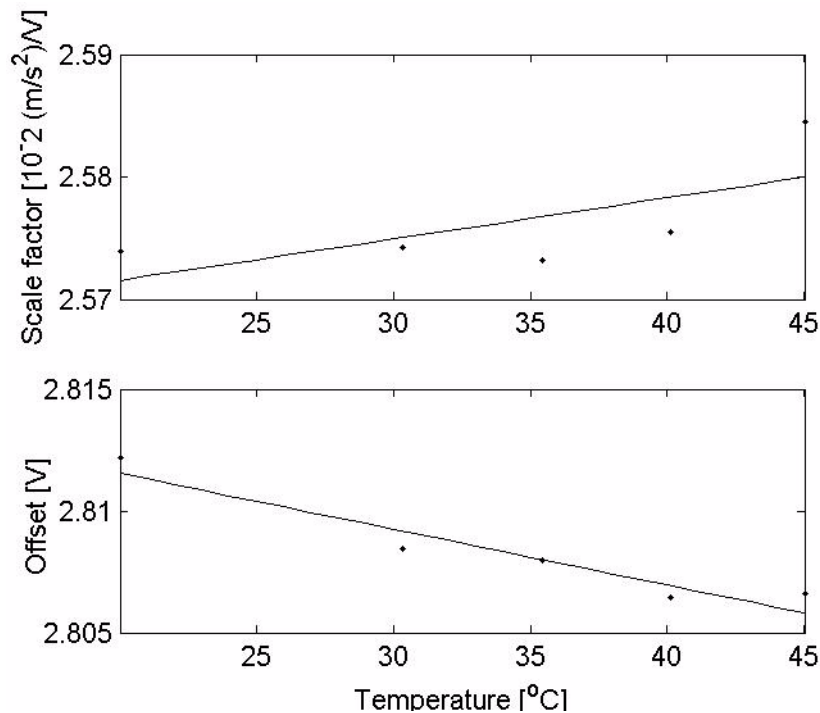


Figure (4.17): Scale factor and offset as a function of temperature, series one.

## 4.3 Results from Simulations of an Inertial Tracking System

### 4.3.1 Simulations with White Noise, Offset and Scale Factor Errors

#### Inertial system error with white noise

The results shown in figure 4.19 are from simulations done with all accelerometer parameters modelled as ideal, and white noise with different standard deviation added to the accelerometer outputs. The simulation time was 20 seconds starting with zero error. The maximum measured error measured during 10 simulations is plotted as the non-stapled line in the figure, while the stapled line is the average maximum error from the 10 simulations. The sampling frequency of the system was set to 2 kHz.

#### Effects of scale factor errors

The results in figure 4.20 is the error as function of scale factor error. The same error was applied to all the accelerometers. The simulation time was 20 seconds, with a 2 kHz sampling frequency. Each of the results is a mean over 10 simulations. The

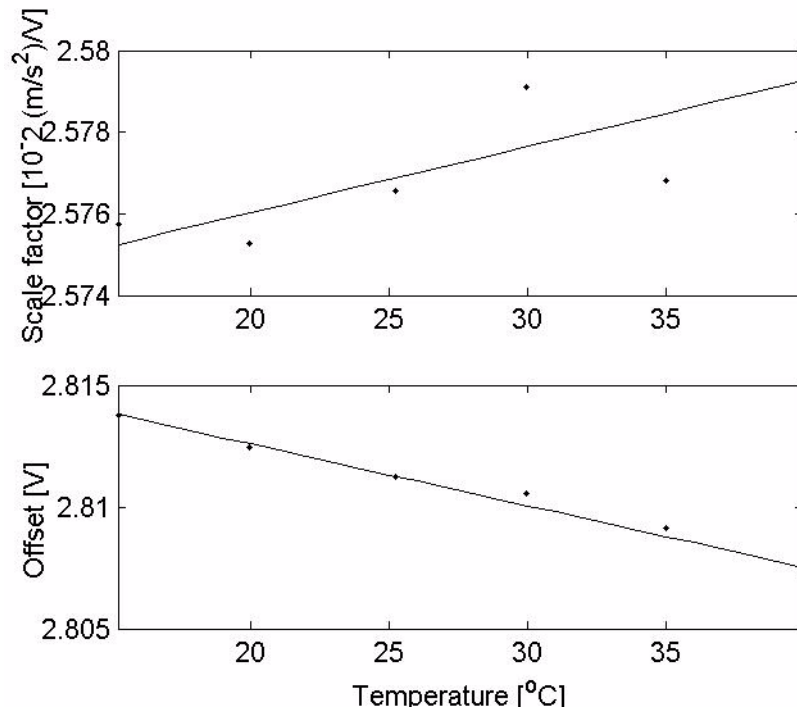


Figure (4.18): Scale factor and offset as a function of temperature, series two

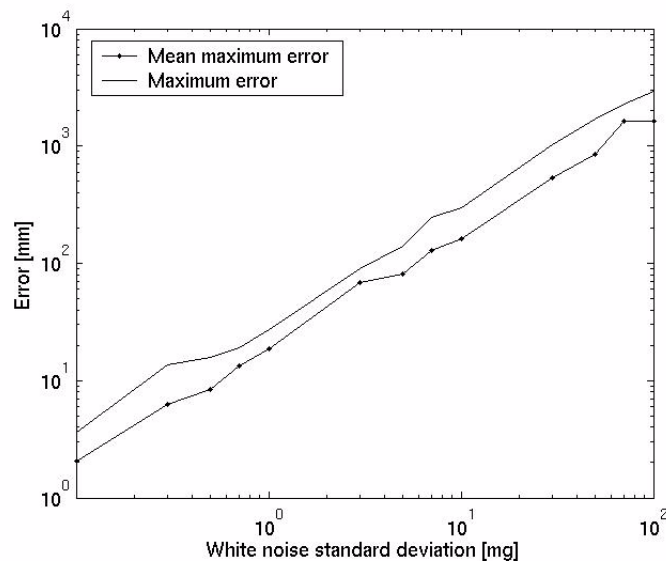


Figure (4.19): System error as function of added white noise. Mean over 10 separate simulations with random movements.

nominal offset was set to 2.8149 V for all the accelerometers and the scale factor to  $2.553e-2 V/(m/s^2)$ .

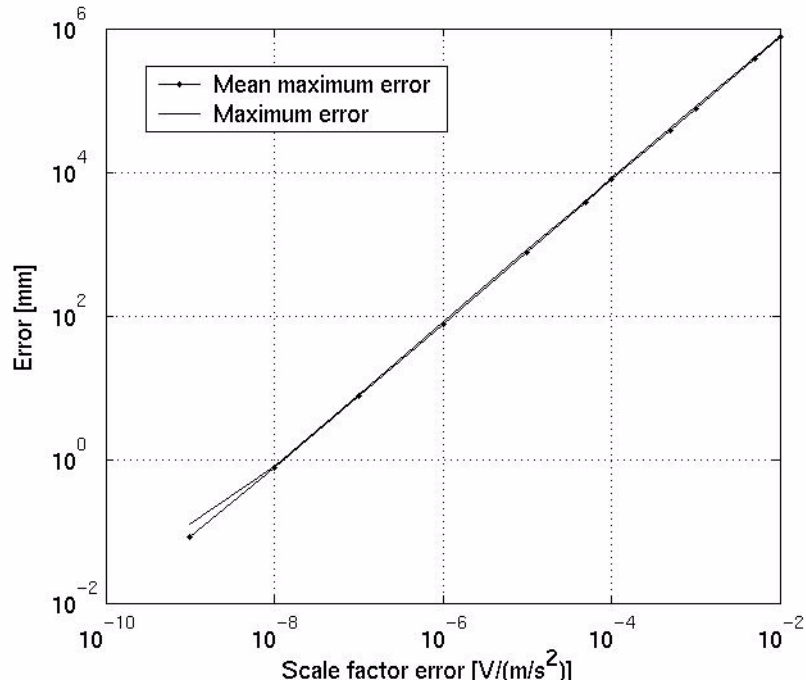


Figure (4.20): Error as function of scale factor error.

#### Estimation error as function of offset error

The plot in figure 4.21 is the system error as a function of offset error. The same error was applied to all the accelerometers, and the error was taken as the mean over 10 simulations for each offset error. The simulation time was 20 seconds. The values of the offset and the scale factor was again set to  $O=2.8149$  V and  $K=2.553e-2$  V/(m/s<sup>2</sup>).

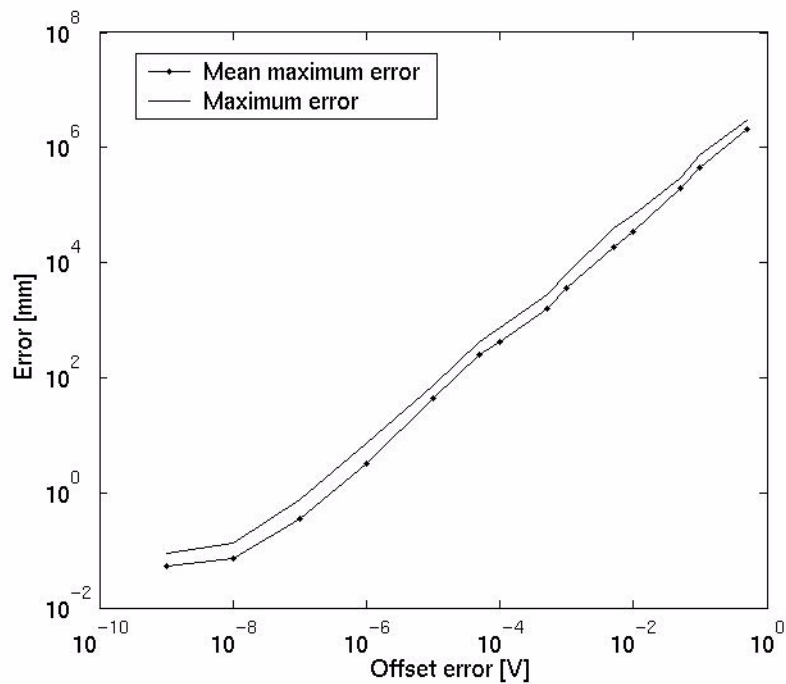


Figure (4.21): System error as function of offset error.

### Combined effect of white noise, offset error and scale factor error

In order to look at the combined effect of offset errors, scale factor error and white noise, several simulations were done where these parameters were varied. The nominal scale factor and offset were the same as in the preceding simulations. The error was taken as the mean of 10 simulations.

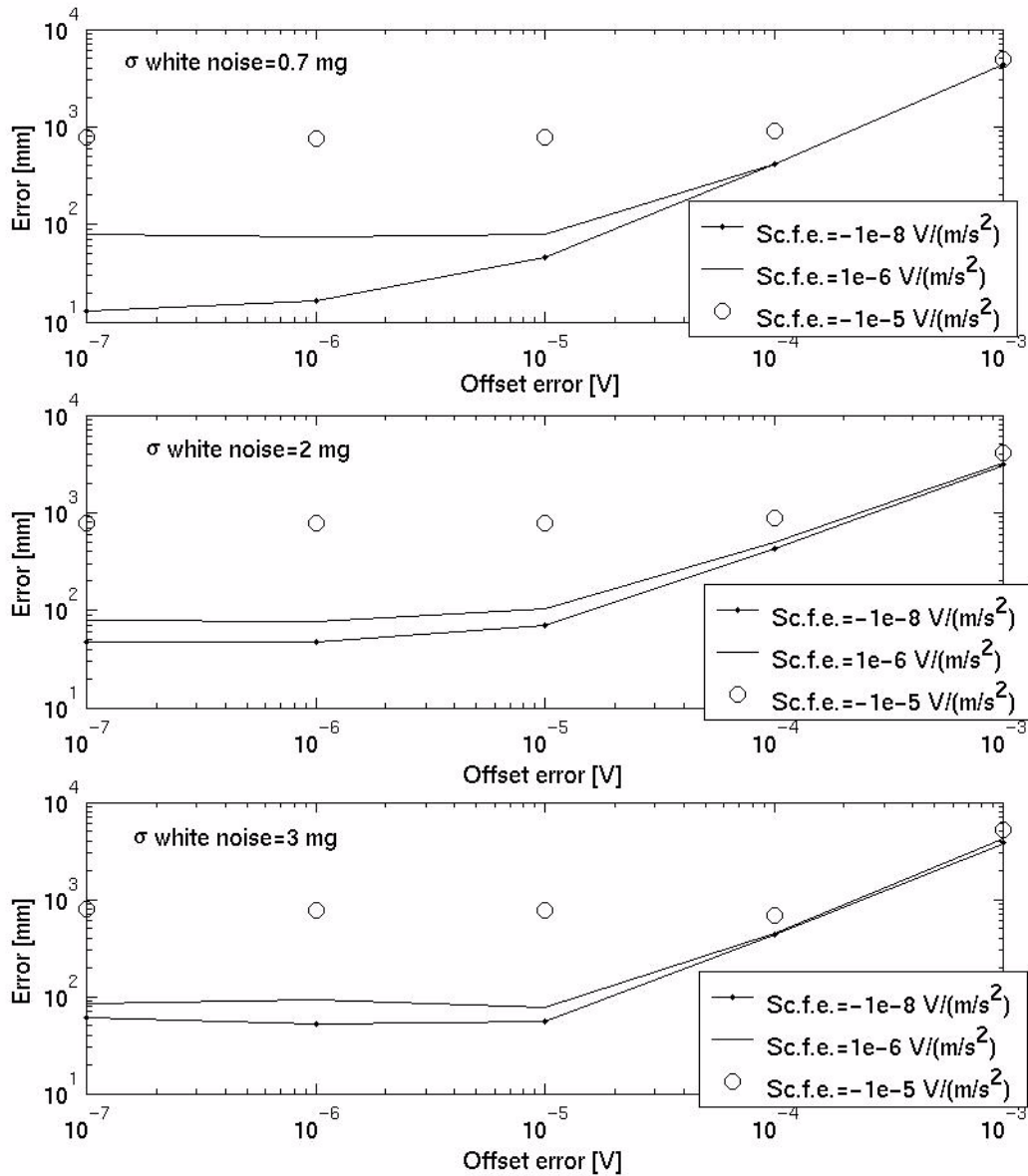


Figure (4.22): System error by combining scale factor errors, offset errors and white noise. Mean error over 10 simulations.

### 4.3.2 Simulations with non-linear Accelerometer Models

When adding higher order non-linear terms to the accelerometer model, the results were the ones displayed in table 4.23. The nonlinear terms were compensated for by using the solution of the polynomial given in equation (3.60).

Table (4.23): System error over 20 seconds. Standard deviation white noise=2 mg, offset=2.8149 V, scale factor =2.553e-2 V/(m/s<sup>2</sup>)

Second order coefficient (V/(m/s <sup>2</sup> ) <sup>2</sup> )	Third order coefficient (V/(m/s <sup>2</sup> ) <sup>3</sup> )	Maximum error, mean 10 simulations, uncompensated (mm)	Maximum error, mean 10 simulations, compensated (mm)
-1e-7	1e-6	8.8e3	53.9
-5e-7	1e-8	0.16e3	47.0
-2e-6,	1e-7	0.79e3	58.4
-6e-6	1e-9	1.1e3	64.0
-1e-5	1e-5	94e3	50.4

#### With cross-axis sensitivity

The next simulations included cross-axis sensitivities. The compensation was now done using equation (3.61) and nonlinear least squares estimation.

Table (4.24): System error over 20 seconds. Standard deviation white noise=3 mg, offset=2.8149 V, scale factor =2.553e-2 V/(m/s<sup>2</sup>)

Kip (V/(m/s <sup>2</sup> ) <sup>2</sup> )	Kio (V/(m/s <sup>2</sup> ) <sup>2</sup> )	Maximum error, mean 10 simulations, uncompensated (mm)	Maximum error, mean 10 simulations, compensated (mm)
-1e-8	1e-8	45.6	50.0
-1e-6	1e-8	164	47.7
1e-4	1e-8	9.69e3	40.8
1e-2	1e-8	1.16e6	57.3
-1e-8	-1e-7	57.2	49.7
-1e-6	-1e-7	132	64.5
1e-4	-1e-7	10.7e3	31.9
1e-2	-1e-7	1.24e6	60.5

#### Simulation using accelerometer models from table 4.15

A mean of the estimated Kip, Kio and K2 and K3 from the table 4.15 were used, while the results from measurement number two were used as the scale factor and offset. The errors

were then calculated as the difference between the highest and the lowest estimate for the nonlinear factors, and the estimated standard deviation for the scale factor and the offset.

Table (4.25): Accelerometer model used in simulation which results are shown in table 4.26

Parameter	Offset	K	K2	K3	Kip	Kio
Value	2.8150	2.552e-2	-1.3562e-6	-2.6612e-8	-7.9150e-7	4.40e-7
Error	2e-4	2e-5	3.6e-6	7.38e-6	3.36e-6	2.39e-6

Table (4.26): System error over 20 seconds. Standard deviation white noise=3 mg. 20 simulations.

	Without parameter errors		With parameter errors	
	Maximum error mean (mm)	Maximum error (mm)	Maximum error mean (mm)	Maximum error (mm)
Linear calculations	526	954	63.5e3	87.3e3
Compensated for non-linear terms	81.2	132	39.0e3	43.1e3
Compensated for all terms	59	138	122	183

## 4.4 Results from Simulations of a Local Positioning System

### 4.4.1 Simulations of Ultrasonic System and Inertial System

#### Accelerometer noise modelled as white noise

Using the model from equation (3.65), applying the ideal smoother and minimizing the residuals with a sampling frequency of 10Hz for the simulated ultrasonic system, gave an ideal spectral density for the white noise of  $1800\mu g\sqrt{f}/(Hz)^{-1/2}$ . The spectral density suggested in the ADXL105 datasheets is  $225\mu g\sqrt{f}/(Hz)^{-1/2}$ . This main reason behind this difference was probably the vibrations induced by the ball bearings, described in section 3.6.4, which are added to the white noise generated by the accelerometer. Experiments were done where the accelerometer output was filtered with a digital lowpass filter to remove some of the vibrations, but this did not improve the performance of the estimations.

The results shown in figure 4.23 is an example from a simulation. The estimation obtained from the smoother is shown as the dashed line, and advantages of applying smoothing can clearly be seen from the plot.

The results shown in figure 4.24 is the results from a single run, where the data from the position sensor has been added noise, and sampled at different frequencies. The error is the difference between the smoothed output and the position measured with the capacitive position sensor.

Figure 4.25 displays the results from the same run, but the standard deviation of the noise added to the position to simulate the ultrasonic system is now varied. The standard deviation of the error and the maximum measured error is taken as a mean over five different sets of generated ultrasonic position data for each different value of the standard deviation of the added noise.

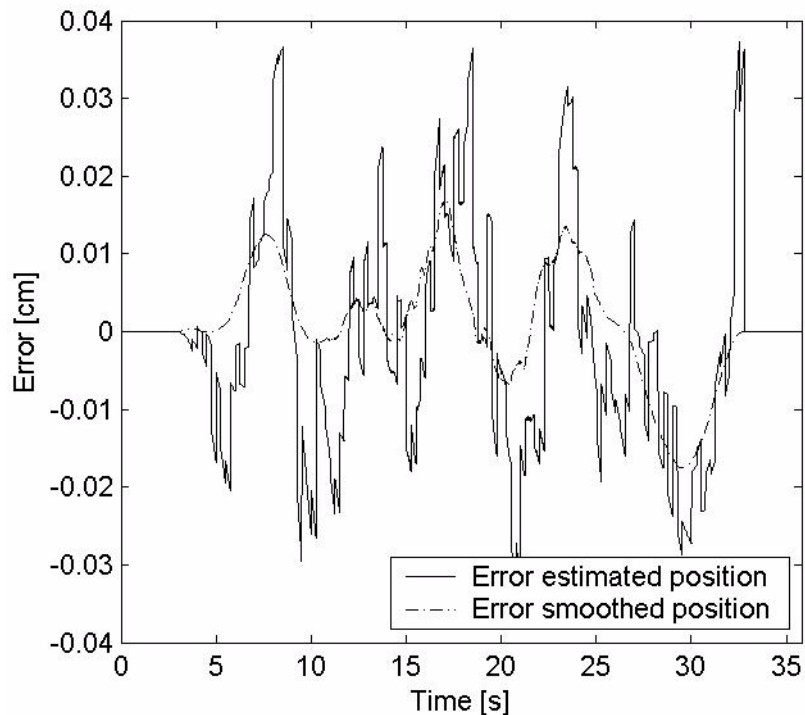


Figure 4.23: Example of simulation of a low accuracy ultrasonic system combined with an inertial system. Sampling frequency 4 Hz, error of primary system:  $\sigma = 30\text{mm}$

#### Accelerometer noise modelled as first order Markov process

The optimal time coefficient for the Markov process,  $\dot{x} = -x/T + w$  was shown to vary as a function of the sampling frequency of the primary system. The spectral density was determined by minimizing the residuals for with an ideal smoother to be  $0.26g\sqrt{f}/(\text{Hz})^{-1/2}$ . The ideal cutoff frequency,  $\beta = 1/T$ , was estimated by minimizing the residuals as described in section 3.8.4 to be about 300 radians/s or 48Hz for a sampling frequency of 10 Hz for the simulated ultrasonic system.

Figure 4.26 and 4.27 shows performance of the smoothing filter when the primary system parameters were varied. The results in figure 4.27 are obtained from averaging over 5 simulations for each frequency instead of using one set of measurement data as it was done for the plots in figure 4.25. The  $1\sigma$  line in the plot was calculated as given in equation (3.70).



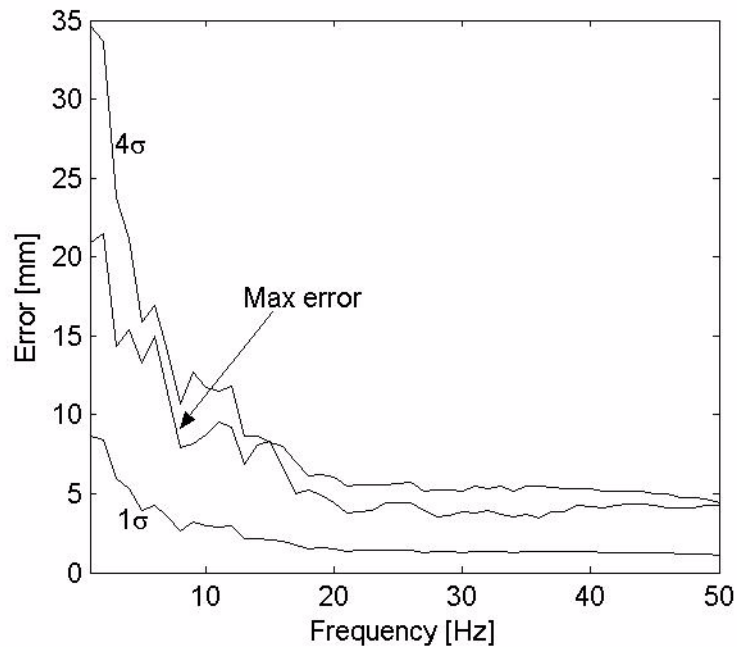


Figure 4.24: Smoothing filter performance as function of primary system sampling frequency. The standard deviation ultrasonic system error  $\sigma = 10\text{mm}$ . Accelerometer noise modelled as white noise

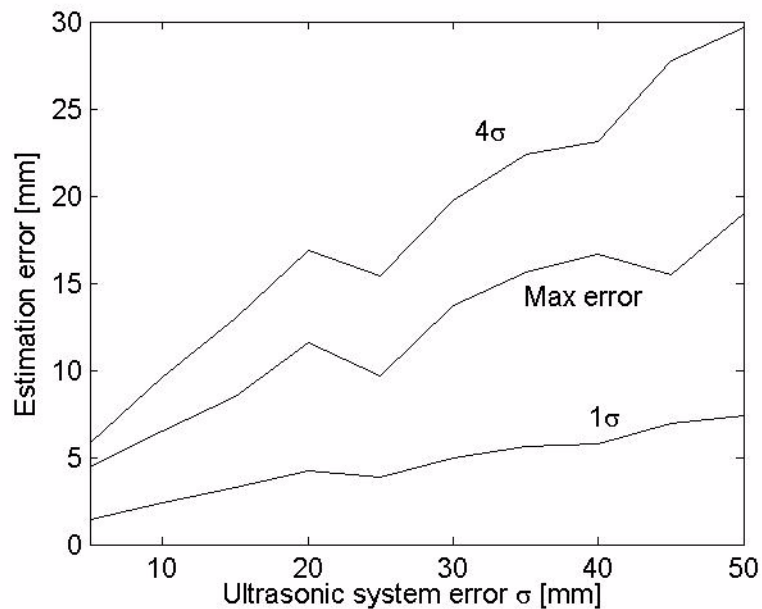


Figure 4.25: Smoothing filter performance as function of ultrasonic system error. Sampling frequency of ultrasonic system 10Hz. Accelerometer noise modelled as white noise.

#### Accelerometer noise modelled as white noise and first order Markov process

The performance of the ideal smoother when the system was modelled as defined in equation (3.67) can be seen in figure 4.28. The ideal cutoff frequency of the Markov process was estimated to be 144Hz, which makes sense considering that the input from

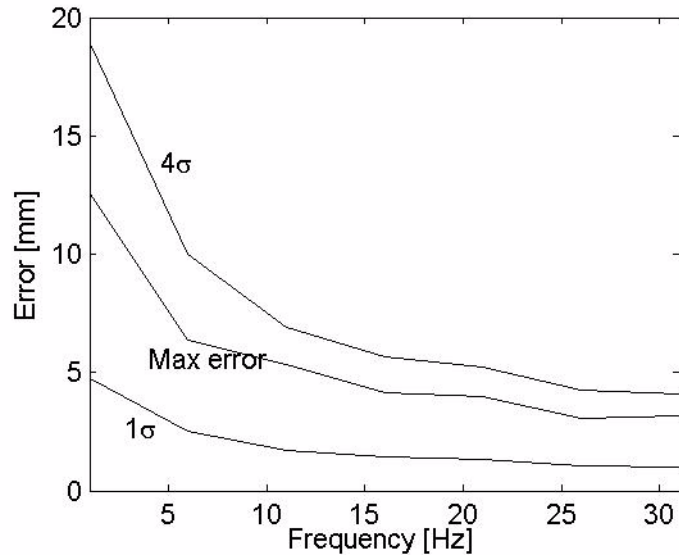


Figure 4.26: Performance of smoothing filter as function of primary system sampling frequency. Standard deviation of ultrasonic system noise:  $\sigma = 10\text{mm}$ . Accelerometer noise modelled as first order Markov process.

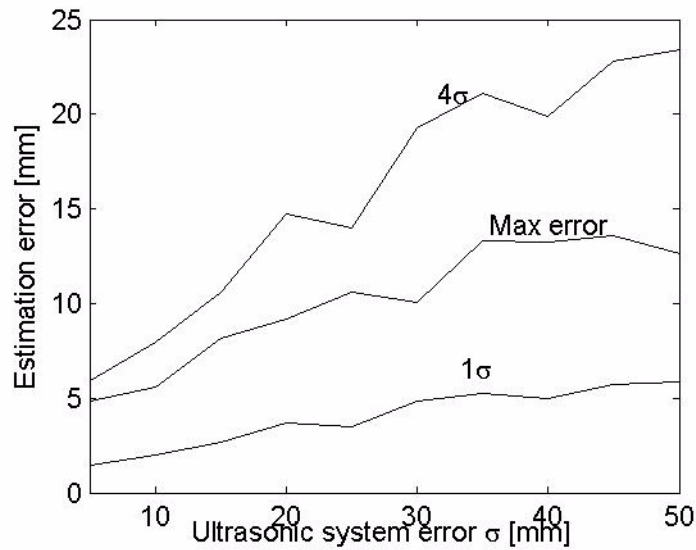


Figure 4.27: Smoothing filter performance as function of ultrasonic system error. Sampling frequency of ultrasonic system 10Hz. Accelerometer noise modelled as a first order Markov process.

the accelerometer has been lowpass filtered by the analog RC filter seen in figure 3.14 with a cutoff frequency of 159Hz. The spectral density matrix defined in equation (3.68) was estimated to be

$$\tilde{Q} = \begin{bmatrix} 122\mu\text{g}\sqrt{f/\text{Hz}^{1/2}} & 0 \\ 0 & 0.202\text{g}\sqrt{f/\text{Hz}^{1/2}} \end{bmatrix} \quad (4.1)$$

for  $f=10\text{Hz}$  and  $\sigma=1\text{cm}$ . The ideal parameters was now shown to vary to some extent with the standard deviation of the noise added to the position to simulate the ultrasonic system.

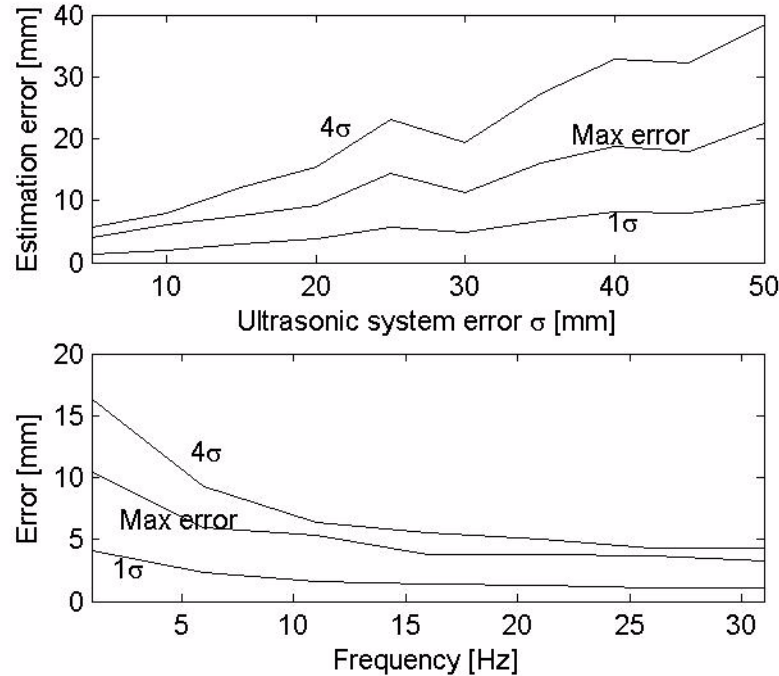


Figure 4.28: Smoothing filter performance as function of primary system error (*upper*) and primary system sampling frequency (*lower*). Sampling frequency of the ultrasonic system was 10Hz for the upper plot, and the system error  $\sigma = 10\text{mm}$  for the lower plot. Accelerometer noise modelled as the sum of a first order Markov process and white noise.

#### Residual minimization and higher order terms

As mentioned in section 3.8.3, the input acceleration as a function of accelerometer output can be calculated as the solution to  $E(a) = O + Ka$  or as the solution of the equation  $E(a) = O + Ka + K_2a^2 + K_3a^3$ . The results shown in the previous figures were all obtained by using the third order polynomial. The improved performance by using this model, can clearly be seen in figure 4.29.

Using these methods gave the estimates  $K_2=-8.7e-6$  and  $K_3=4.8e-5$  when the accelerometer noise was modelled as white noise and  $K_2=-2.5e-5$  and  $K_3=3e-5$  when the process noise was modelled as the sum of a first order Markov process and white noise.

#### 4.4.2 Simulations with Periods without Primary System Measurements

Figure 4.30 shows an example of a simulation where the data from the primary system is left out for a 10 seconds long interval. As mentioned in section 3.8.6, the error in the estimated position during the period without primary system measurement updates

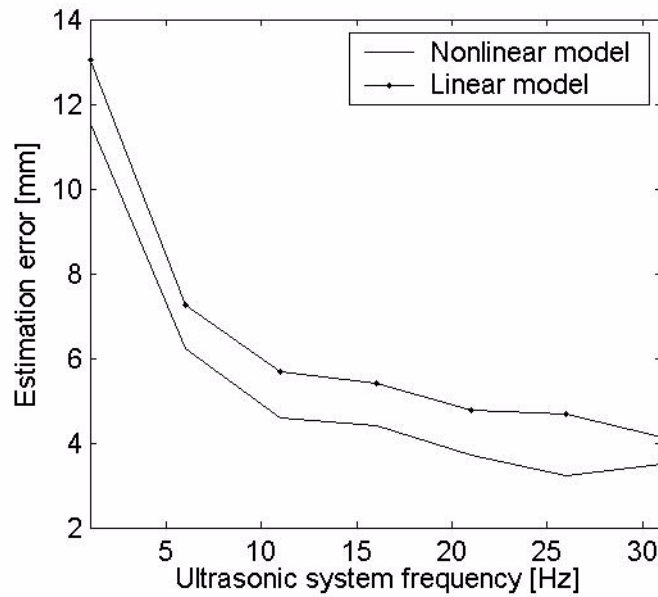


Figure 4.29: Smoothing filter performance as function of frequency for a linear and nonlinear accelerometer model. Maximum error plotted. Primary system error  $\sigma = 10\text{mm}$ . Accelerometer noise modelled as sum of Markov process and white noise.

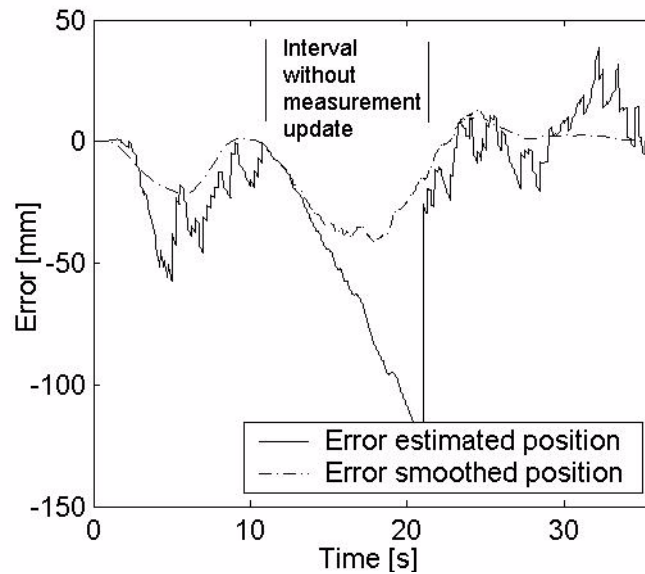


Figure 4.30: Example of Kalman filter performance in periods without measurement update.

depended on several factors, as for instance the shape of the movement of the system during the period.

In order to obtain an estimate of the error for a given interval, the interval was placed at different times during each of the experiments described in section 3.8.2. The maximum error and the standard deviation of the error was then logged for each starting time. For each starting time 5 different sets of ultrasonic position

measurements were generated, and the results averaged. The accelerometer noise was modelled as given in equation (3.67) during all of these simulations, with the noise parameters as estimated in section 4.4.1.

### High-accuracy primary system

The high accuracy primary system was defined as added noise with standard deviation  $\sigma=1\text{mm}$  added to the position data, and a sampling frequency of 20Hz. The errors when the interval without measurements update was placed at different times are shown in figure 4.31. When the length of the interval was five seconds, the mean maximum error was 6.0mm and the mean of the standard deviation of the estimated error was 2.3mm. For the 10 seconds long interval without measurement updates, the mean of the maximum error was 9.7mm and the mean of the estimated standard deviation was 3.2mm.

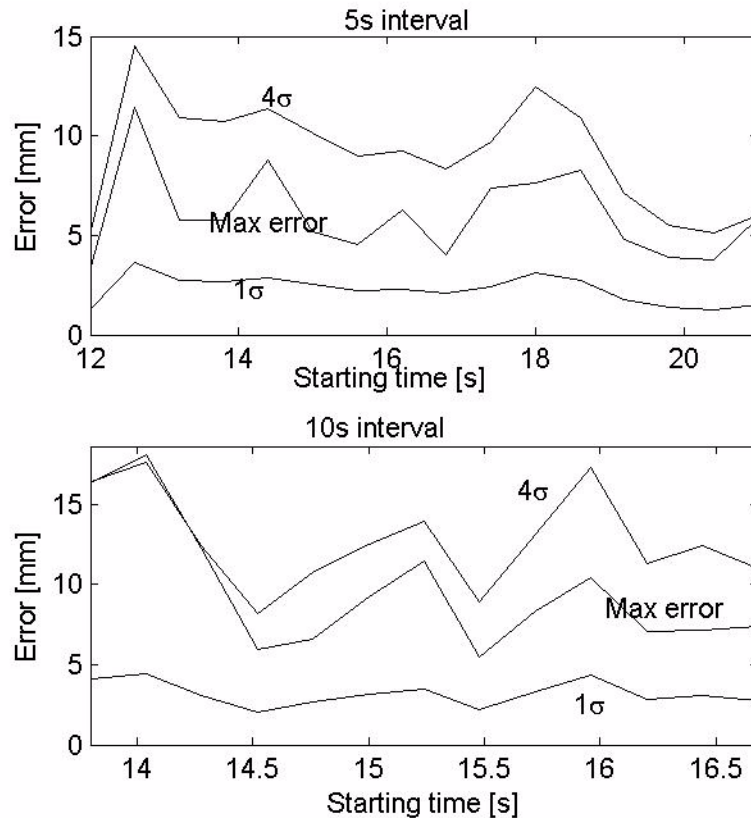


Figure 4.31: Smoothing filter performance during periods without measurement updates as function of the interval starting time with a high accuracy ultrasonic system as primary system.

### Low accuracy primary system

A low accuracy ultrasonic system was simulated by adding noise with 30mm standard deviation and sampling with a frequency of 4Hz, and the results of the simulations are

shown in figure 4.31. The maximum error did not stay within four standard deviations when the interval length was ten seconds. The mean of the result are shown in table 1.

**TABLE 1. Mean smoothing filter performance during periods without measurement updates with a low accuracy primary ultrasonic system.**

Interval length (s)	Mean maximum error (mm)	Mean standard deviation of error (mm)
3	18.7	3.28
5	24.6	6.26
10	34.8	16.7

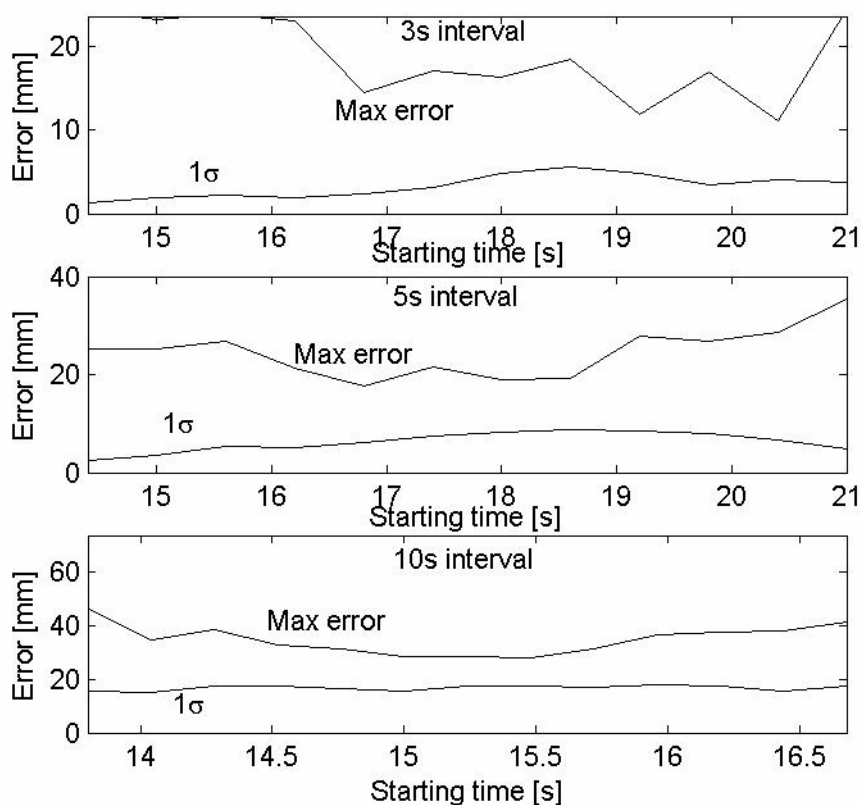


Figure 4.32: Smoothing filter performance during periods without measurement updates as function of the interval starting time with a low accuracy ultrasonic system as primary system.

### No primary system

These simulations corresponded with a situation where the inertial system is placed in a known position before and after each run. The same estimated offset, scale factor and higher order coefficients were used in the estimations on each run. The six different runs of the type shown in figure 3.41 were used, and the position was estimated

without primary system measurements during the time the system was moving. The results of the estimations are shown in table 2.

**TABLE 2. Smoothing filter performance with the inertial system as a stand-alone system.**

Run number	Movement time (s)	Standard deviation of error (mm)	Maximum error (mm)
1	14.4	4.36	17.2
2	15.0	3.29	16.4
3	14.58	2.05	8.83
4	13.8	2.54	10.7
5	18.48	2.06	10.0
6	14.58	2.14	8.88

#### 4.4.3 Simulations of an Ultrasonic System

By minimizing residuals, the correlation time was found to be  $T = 1/\beta = 62.5\text{ms}$  and the spectral density of the white noise to be  $0.94g\sqrt{f}/\text{Hz}^{1/2}$ . The performance of the smoothing filter is shown in figure 4.33.

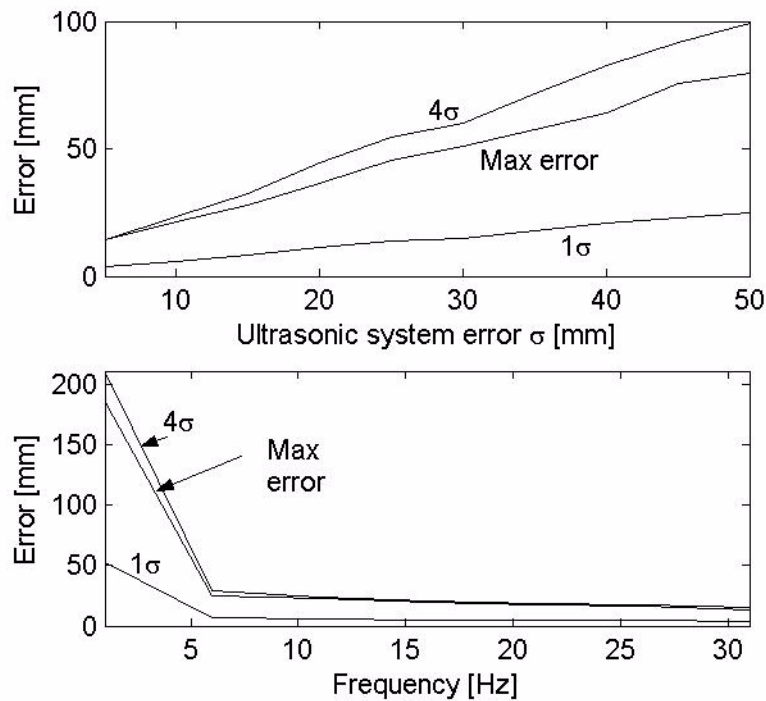


Figure 4.33: Performance of the smoothing filter with a stand-alone ultrasonic system as function of system error (*upper*) and sampling frequency (*lower*). Sampling frequency 10Hz for the upper plot, and system error  $\sigma = 10\text{mm}$  in the lower plot.

## 5 Discussion

### 5.1 Data Acquisition Setup

As mentioned in section 3.3 the main problem with the DAQ setup used in this thesis was the variations in the offset voltage of the instrumentation amplifier. The bipolar capacitance used to lowpass the offset voltage, CB10 in figure 3.14, was the main suspect as the cause of the error. The error could have been avoided by choosing a different capacitance with smaller leakage current, using two capacitances with opposite temperature characteristics, or simply by removing the capacitance. The gain of the AD620 instrumentation amplifier is set by a single resistance, R13 in figure 3.14. Although a metal film resistor with 1% tolerance was used, the temperature coefficient was unknown. A metal film or wire wound resistor with lowest possible temperature coefficient should have been used to avoid changes in the amplifier gain as a result of variations in the ambient temperature.

The source of the problem may also have been temperature dependent changes in the performance of the AD620 amplifier, which would have been more difficult to solve. Extensive testing would have been needed to model and compensate for such errors.

When the problems with variations in the offset had been discovered and compensated for by measuring the gain and offset before each experiment, the measurement accuracy was good enough. This was an ad hoc solution, but proved to be effective.

An alternative to using an instrumentation amplifier could be the setup shown in figure 5.1. This setup use a high resolution delta-sigma ADC with a built in voltage reference and amplifier with programmable gain. The voltage reference may be used to set the offset voltage of the input as the negative input to the differential ADC.

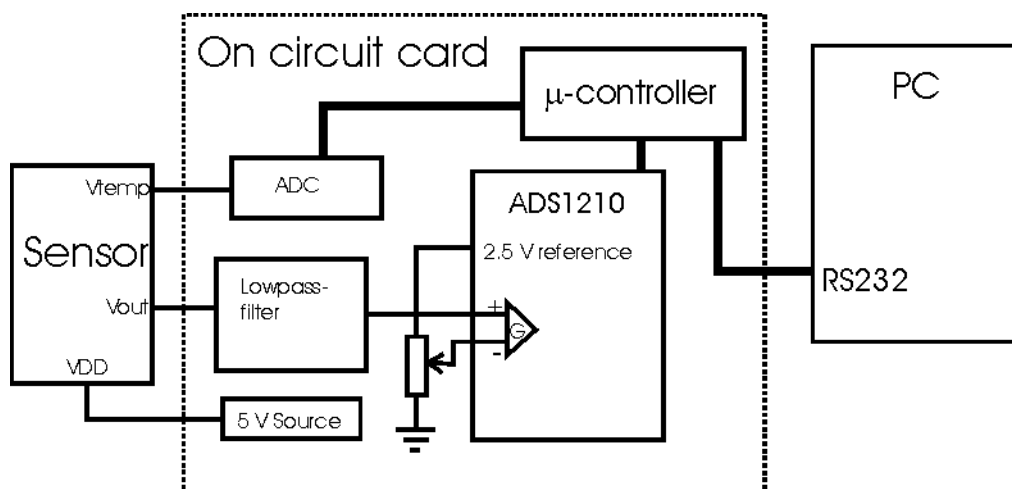


Figure 5.1: DAQ setup using an external ADC interfaced via the PC serial port.

The main advantage of using the instrumentation amplifier and a PC-based commercial DAQ card, was the ability to control the sampling intervals. The input data was time-stamped with good accuracy, and the outputs from the position sensor and the



accelerometer during the dynamic experiments were synchronized. Using an external ADC as shown in figure 5.1 would require two different digital interfaces for the position sensor and for the accelerometer, thus complicating the system and introducing problems with time-stamping and synchronization. This would not cause any problems in a tumble-test where the accelerometer output averaged over time is used, but in a local positioning system and during the dynamic experiments, the outputs from the position sensor and the IMU need to be synchronized. When using an external ADC this problem may be solved by over-sampling accelerometer output and taking the mean over small intervals, or by constructing a setup that time-stamps the data from both systems.

## 5.2 Static Experiments

### 5.2.1 The setup

Many of the measurements suffered from the flaw in the DAQ setup described in the previous subsection. When the problem was discovered, and compensated for by measuring the amplifier offset and gain before each experiment, the repeatability of the estimations over longer periods improved.

The absolute accuracy of the rotation stage used during the experiments was 0.05 degrees, which is in the medium performance range. As described in the IEEE accelerometer testing standard (IEEE Std 1293-1999), extreme accuracy in the measurements is needed to get accurate estimations of quadratic and especially cubic nonlinearities. Thus further testing is needed to estimate the nonlinear coefficients, either by increasing the number of measurement in a test, or using a more accurate rotation device. This was reflected in the variations of the estimated parameters.

Because the input angle of the rotation stage was set manually, each experiment lasted up to one hour. As all the input data is used in the estimations, it is impossible to detect changes in accelerometer parameters with bandwidths higher than  $1/T$ , where  $T$  is the duration of the measurements. An automated device may decrease the time needed to do each measurement series, and thus increase the ability to track time dependent variations in the estimated parameters. A machine dividing head, as suggested in the IEEE accelerometer testing standard may be a good instrument for these tests, but would increase the complexity and cost of the setup.

Mounting the accelerometer on a sheet of metal with a 90 degrees bend as described in section 3.5.3 and shown in the figures 3.22 and 3.23, worked as a simple way to be able to rotate the accelerometer about both the principle- and the output-axis. However the procedure described to align the sheet to previous mounts on the rotations stage did not give the wanted results. A difference of  $0.45^\circ$  was estimated between two of the remounts, see table 4.12. To improve the method, the accelerometer under test may be mounted on a cube which is machined close to perfectly cubic. The cube could then be clamped on the rotation stage using a spring, as seen in figure 5.2. This method would make it possible to do rotations about all the three axes, the input-, output- and principle-axis. Rotations about the input axis could be used to measure the alignment of the input axis to the side of the cube.

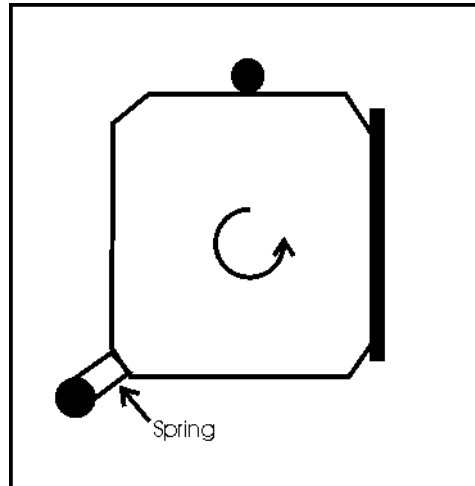


Figure 5.2: Alternative setup aimed to be able to rotate the accelerometer about more than one axis.

The experiments showed that the heat boxes described in section 3.2 were inconvenient as a mean to measure performance as a function of temperature. The control systems using a Peltier element like the MPT5000 seemed like an ideal solution for these kind of experiments.

The system specifications listed in section 2.1.1 gives that the input range of the system is  $[-2g, 2g]$ . The multi-position tumble tests, at least when performed on the surface of the earth, can only characterize accelerometers in the input region  $a=[-g, g]$ . At the higher input accelerations, the higher order nonlinear terms may become more dominant, and the estimated scale factor may change if a different range is used, as illustrated in figure 3.37.

### 5.2.2 The results

The long term static measurements proved like a good way to examine changes in offset over time, but other experimental methods is needed to look at changes in scale factor. Using the method of identifying noise slopes in a PSD plot from section 2.4.3, the output from half an hour of the high-accuracy Crossbow accelerometer in a static position shown in figure 4.3 can be interpreted as the sum white noise and random walk. The random walk has frequencies below 0.1 Hz. The slopes for the ADXL105 accelerometers were in general not that easy to identify. As seen from figure 4.2 the spectral density is higher, and the point where the white noise becomes flicker noise or random walk is at 1 Hz or higher. Thus the better performance of the Crossbow accelerometer is evident from these plots.

The results in the tables 4.3 to 4.9 shows the estimated standard deviation for each parameter increased as the number of parameters included in the estimation was increased. This is probably due to the fact that the number of estimated parameters is increased, but not the corresponding number of measurements. Table 4.10 and 4.11 shown that the standard deviation estimated for the offset and the scale factor increases significantly when second and third order nonlinearities are estimated, but the value of the estimated parameters stays close to unchanged. The estimated standard deviation of

the offset and the scale factor when only the offset and the scale factor are estimated from the experiment, may be a better indication of the true error.

The mean of the estimated standard deviations for the offset shown in table 4.15, is 0.2 mV which matches the variation of the estimated offset between each measurement series. The scale factor had an estimated standard deviation of around  $1e-5$  V/(m/s<sup>2</sup>) which also matched the variations between each measurement series. The estimates done for the Crossbow accelerometer in table 4.10 gave a standard deviation of  $1e-5$  V for the offset and  $1e-6$  V/(m/s<sup>2</sup>) for the scale factor. This indicates that the estimated standard deviations in for the ADXL105 accelerometer is limited by random walk or other noise phenomena, not by the accuracy of the experimental setup. A standard deviation of 0.2 mV matches the plot shown in figure 4.5, where the random walk over half an hour has an amplitude of the order 0.2 mV. Thus measurements may have to be repeated over longer periods of time in order to compensate for the offset random walk in the ADXL105.

The standard deviations of the second and third order nonlinear coefficients and the cross-axis sensitivity also seen from the table 4.15, are several orders larger than the estimated parameters. This is also reflected in the variations of the estimated parameters. The same variations were seen in the estimated nonlinear coefficients of the Crossbow accelerometer, indicating that further and more accurate test procedures are needed to estimate these parameters.

The three different series shown in the figures 4.6, 4.7 and 4.8 suggest that the offset and the scale factor as function of temperature is more predictable than the  $\pm 1$  mV/°C specified in the datasheets from Analog Devices. The tests were done with two different accelerometers and they both showed that the offset decreased with temperature between 15 and 40 °C, while the scale factor stayed constant within the measurement accuracy. The difference between the two series done with the same accelerometer shown in the figures 4.7 and 4.8, may be caused by measurement errors due to misalignments or it may be real variations in how the offset varies with temperature. Further testing is needed to look into this.

## 5.3 Dynamic Experiments

### 5.3.1 The setup

The main problem with the setup used during these experiments was the vibrational accelerations picked up by the accelerometers and the acceleration due to stick-slip friction. The maximum amplitude of the vibrations was up to 0.26 g, which is quite high. With a frequency of up to 300Hz, the amplitude of sinusoidal movements would have to be  $A_p = (0.26 \cdot 9.81 \text{ m/s}^2) / (2\pi 300 / \text{s})^2 = 0.7 \mu\text{m}$ , which is below the configured resolution of the position sensor.

The following suggestions are given as possible improvements of the experimental setup:

- Use anti-vibrational mounting. The accelerometer could be mounted in a more rigid way inside the aluminium box, instead of using two plastic screws. Passive or active damping could be also be applied.
- Take more care to avoid vibrations transferring through cable. Other kind of system could be used to transfer the signals from the accelerometer, for instance by wires attached to the inside of the box instead of attaching the wires from the wire directly to the chip carrier board by. Using a battery and wireless communication would be the ideal solution, provided that the linear voltage regulator is insensitive to changes in the supply voltage. This method would make it harder to synchronize data from the accelerometer and the position sensor.
- Changing the ball bearings, which may have a damaged or deformed ball, or consider another way of mounting the platform on the rig.

The influence of the stick-slip friction on the estimations was discussed in section 3.6.5. The stick-slip friction was shown to vary as a function of the shape of the movement and the velocity. The following points are suggested to reduce this effect.

- Use mechanical force to move the accelerometer back and forth. The input movement can be better controlled, and the slip-stick friction be reduced if the system is rigid enough.
- Use less rigid ball bearings. The shock experienced every time the sensor starts moving at the peak of each cycle, could be reduced by reducing the friction between the platform and the rig.
- Disregard signals in the areas where the effects of stick-slip friction occurs. This reduces the amounts of data, as the sensor spends the most time in the areas where the velocity is lowest. The total amount of input data may have to be increased to keep the same accuracy.

The accuracy of the estimations could be improved to some extent if some or all of the vibrations could be removed. Figure 5.3 shows an A-V plot from a simulated movement. The accelerometer is modelled as the results from table 4.15.

### 5.3.2 The results

As seen in the tables 4.17 and 4.18, the estimated accelerometer offset and scale factor depended on the filter parameters and the shape of the movement. The long time stability of the estimated offset seen in the figures 4.9 and 4.10 was higher than the variations in the half-hour plot in figure 4.5, independent of the chosen filter parameters. The long time variations in the scale factor were also larger than the variations in the scale factor estimated from the static tests.

The results of the alignment experiments were independent of whether the estimated scale factor matched the real value. The results in table 4.19 shows a repeatability of 0.1 degrees which is quite good, as alignment better than 0.1 degrees puts great demands on the resolution of which the scale factor has been estimated. The resolution has to be in the order  $\cos(0.1^\circ) = 0.9999984$  or  $10^{-6}$  which is better than the resolution obtained during the static experiments. As the repeatability of the estimations was shown to depend on the movements of the platform, the good results may also have

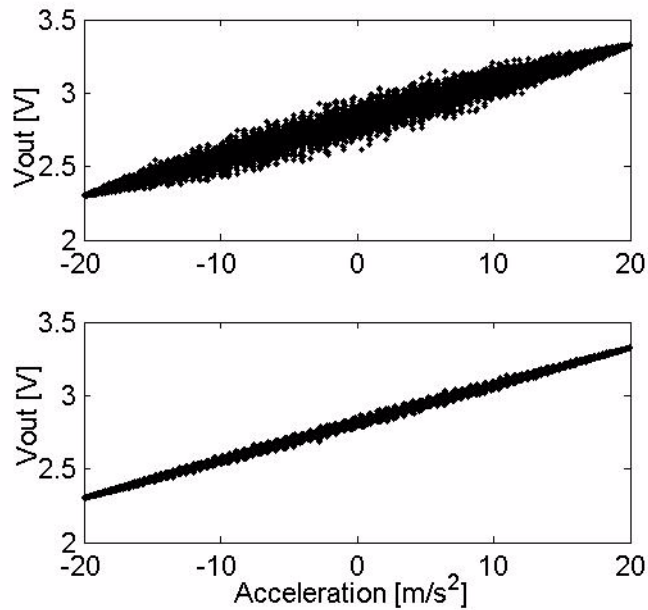


Figure 5.3: A-V plot for simulated dynamic experiment. Upper plot with vibrations with magnitudes of 0.26 g, correlated with velocity. Lower plot with vibrations with magnitudes of  $0.26g/4=65$  mg

been a result of that the operator who pushed the platform was able to repeat the same movements for each of the experiments.

Decreasing the gain of the amplifier and thus increasing the input range improved the repeatability of the estimated accelerometer parameters to some extent. The effects of the stick-slip friction were less dominating when the input range was higher. The apparently lower accuracy in the alignment experiments seen when comparing the results in table 4.19 to the ones in table 4.21, may be caused by the fact that it was much harder to do approximately periodic sinusoidal movements when the velocities increased. The accelerations after the gain was decreased were measured to have amplitudes up to 3 g, which fits the system specifications from section 2.1.1 better than the input during the static experiments which was limited to 1 g.

The density plots from figure 4.16 shows a phase delay in the signal from the Crossbow accelerometer. The delay may have been caused by the a malfunction in the sensor, or it may be phase delay in the output of the sensor that not is specified in the datasheets from Crossbow. This phenomenon could not have been discovered in a multi-position tumble test or in a centrifuge test.

## 5.4 Simulations of an Inertial Tracking System

### 5.4.1 The simulations

As shown in the figures 4.19 to 4.21, the simulations appeared to be a good way estimate the influence of one single error in an estimated parameter on the performance

of the IMU in a stand-alone inertial system. The performance of the system as a function of several combined errors was more complex. A statistical method like for instance the algorithm called ‘variation of the parameters’ could have been used to examine the combined effects of the errors.

In addition the setup had the advantage that models of all thinkable accelerometers can be inserted into the simulations, and thus also the performance of any future components with expected specifications could be tested. This could also be seen as a disadvantage as models instead of real accelerometer output were used. Any unmodelled accelerometer errors are not included in the simulations. As mentioned in section 3.7 the main flaw with these simulations were the missing gyro- and misalignment errors.

The method of generating random movements and calculating the mean error seemed to work as way to test the performance of the system. The method could be expanded by increasing the number of simulations and mapping the probability density of the error.

Solving equation 3.61 numerically by the method of least squares was shown to be effective to compensate for cross-axis sensitivity, although time-consuming. The processing time for 20 seconds of output from the IMU increased from 20-30 seconds to about three minutes. No analytical solution to the equation was found, at least within the computational power of the mathematical software Maple. An analytical solution was found when numeric values were inserted for all the parameters except the IMU accelerometer outputs, but still far too complex to be implemented effectively in a MATLAB function.

Accelerometer errors were modelled as constants and no time-varying errors were implemented. This could easily have been achieved by modelling the errors as random processes and using the random number generator of MATLAB. The models for the ADXL105 accelerometer noise obtained in section 4.4.1 could have been inserted in the simulations.

## 5.4.2 The results

Figure 4.19, figure 4.20 and figure 4.21 showed that the system error as a function of offset error, scale factor error and white noise, was proportional to the parameter errors and the standard deviation of the white noise. The results shown in figure 4.22 and other similar simulations indicated that the combined effect of these three errors was a linear combination of the separate effects.

The effect of nonlinear errors and cross-axis sensitivities on the system performance was more complex. As seen in table 4.23 the nonlinear parameters can be compensated for, while the error can be large if they are not accounted for. In order to get the wanted results, the compensation have to be performed when the magnitude of the coefficients reach a certain level. for instance the values  $K_2 = -1e-5 \text{ V}/(\text{m}/\text{s}^2)^2$  and  $K_3 = 1e-5 \text{ V}/(\text{m}/\text{s}^2)^3$  gave an error of 94 meter when not compensated for and 50.4 mm when compensated for.

The results displayed in table 4.24 showed that errors due to the cross-axis terms is in the same order as for the nonlinear terms. The results also showed that the errors can be compensated, using the nonlinear least squares algorithm to solve the nonlinear equation (3.61).

The simulations using the ADXL105 accelerometer model from the static measurements which results are shown in table 4.26, showed that compensating for the nonlinear and cross-axis terms is necessary when their magnitude reach a certain level. As seen in table 4.26, compensating for the cross-axis terms is necessary when the magnitudes are in the order of  $10^{-6}$ , while compensating for the nonlinear terms may be good enough when they are in the order of  $10^{-7}$ .

When the specific force is calculated from the IMU output as the least squares solution of the nonlinear equation (3.61), the performance of the system is limited by the errors in the estimated parameters and the noise floor of the accelerometers. The error in the estimated parameters also include parameter random walk. If compensation for third order nonlinearities is applied as solution to equation (3.60), then the performance is decided by measurement errors, noise floor and the magnitude of the cross-axis coefficients. If no form of compensation is applied and the specific force is calculated from the linear equation (3.59), then the performance of the system is decided by magnitude of the nonlinear coefficients and the cross-axis sensitivities together with the accelerometer noise floor and the measurement errors.

## **5.5 Simulation of a local positioning system**

### **5.5.1 The simulations**

The main advantage with these simulations was that real accelerometer output was used in the simulations, which made them close to reality. The main disadvantage was that the movements were limited to one dimension, and that any gyro errors which according to Titterton and Weston (1997) will dominate the error of an strap-down inertial system, are left out from the simulations. In addition, the model of the primary system, may not have been realistic.

Using a linear Kalman smoother, showed to be very effective and especially increased the time of which the inertial system can manage within a given error without data from the primary system. The ideal smoother also worked as way to test and verify different noise models for the accelerometer, which are needed in a minimum variance filter like the Kalman filter.

### **5.5.2 The results**

When estimating the accelerometer noise, different noise models were shown to fit for different primary system parameters as shown in section 4.4.1. This indicates that there are unmodelled errors in the accelerometer output. The vibrations picked up by the accelerometer from the rig would probably have to be modelled to get optimal filter performance, but the vibrations are correlated with the velocity. According to Jazwinski (1970) Ito calculus is needed to cope with errors correlated with the state of the system, which was considered to be way beyond the scope of this thesis.

For low primary system sampling frequencies, the accelerometer noise was best modelled as the sum of a first order Markov process and white noise, see figure 4.28. The cutoff frequency of the Markov process was shown to be close to the cutoff frequency of the first order lowpass filter in the analog signal conditioning.

The figures 4.31 and 4.32 showed that the performance of the inertial system during periods without measurement updates depends on the accuracy and sampling frequency of the primary system as well as the quality of the inertial system. Thus no clear conclusion can be made on the performance of the inertial system during stand-alone periods without knowing the primary system specifications. In order to get simulations close to reality, some focus should be set on measuring and modelling the performance of primary systems as well as inertial components and systems.

The improvement of the results when compensating for higher order nonlinear terms shown in 4.29, may be due to the fact that the nonlinear terms compensate for errors in the estimation of the scale factor. The scale factor was estimated using the methods from the dynamic experiments, which was shown to give a relatively large uncertainty in the estimations. The estimated nonlinear terms did not match magnitude of the nonlinear coefficients estimated during the multi-position tumble tests.

The inertial system as a stand alone system, can not be directly be compared to the results from the simulations of an IMU, as a linear smoother was applied on the accelerometer output. As seen in figure 4.30 the smoothing filter improves the performance of the estimation to a great extent.

## 5.6 General Discussion

The simulations of an inertial system which results were given in section 4.3, showed that it is possible to extend the navigation algorithms to compensate for both nonlinear terms and cross-axis sensitivities if values of these coefficients are known. The simulations also showed that the performance of the system will be dominated by these errors if they are large enough and not compensated for.

The experiments setups tested in this thesis did not have the accuracy to measure these coefficients, which made it impossible to make any conclusions on the long time stability of the terms. Using an accurate on one or more units, may give indications on whether these parameters are stable over time. To repeat such procedures after each start-up, would require a lot of expensive experimental equipment and be time-consuming. If these coefficients prove to be relatively constant over time, each component may calibrated once. The scale factor and the offset could be determined more often by using less accurate tests.

The accuracy over time for a given accelerometer if compensation for nonlinear terms and cross-axis sensitivities is not applied, can be estimated by the magnitude of these parameters together with the parameter random walk and noise floor.



## 6 Conclusion and Recommendations

### Conclusion

Two different experimental setups to characterize accelerometers were developed and tested on different low-cost accelerometers.

The multi-position tumble tests which used the static force generated by gravity as input, was shown to be accurate enough to estimate the accelerometer parameters offset and scale factor with a repeatability in the order  $\delta K/K \sim 10^{-4}$ . The accuracy of the setup was not good enough to get accurate estimations of nonlinear factors or the sensitivity to accelerations normal to the input axis. The setup can be improved by using more accurate equipment, or increasing the number of measurements in each experiment.

The second setup which used dynamic accelerations in one dimension and an accurate position sensor to calculate the accelerations, was shown to be corrupted by vibrations in the experimental equipment picked up by the accelerometer. The method could be improved and possibly used to characterize accelerometers if the experimental equipment is modified to damp or remove most of the vibrations. However the setup proved to be useful for doing simulations of a one-dimensional local positioning system.

Using multi-positions tumble tests on the low-cost ADXL105 accelerometer from Analog Devices showed that each unit was different characteristics and had to be calibrated individually. The data gathered from the experiments where the parameters were examined as function of temperature, concluded that the accelerometer offset varied with temperature as  $-0.3 \text{ mV}/^\circ\text{C}$  to  $-0.4 \text{ mV}/^\circ\text{C}$  in the region  $T = [15, 40]^\circ\text{C}$ , while the scale factor stayed constant within the measurement accuracy. This is a significant improvement of the performance specified in the ADXL105 datasheets, that the offset could change with temperature as  $\pm 1 \text{ mV}/^\circ\text{C}$ . Further experiments are needed to look at long-time variations like run-to-run differences and aging if any conclusion is to be made on whether the accelerometer model needs to be recalibrated each time after power-up.

The simulations of a strap-down inertial navigation system conducted in this thesis, showed that both cross-axis sensitivity and higher order nonlinear coefficients can be compensated for if they are known, and thus improving the performance of the system several orders. This requires much computational power and would be difficult to achieve in a real time system.

The simulations of a one-dimensional local positioning system indicated that combining a chosen primary system with an inertial system would improve the performance of the local positioning compared to the primary system as a stand-alone application. The performance of the inertial system as a back-up system during periods where data from the primary system is unavailable was shown to depend on the accuracy of the primary system. The application of an ideal smoothing filter improved the performance of the system significantly.

As the errors of a strap-down inertial system is dominated by errors of the gyros, it is hard to make any conclusion on possible performance of any present or future system on the basis of the results gathered in this thesis. In one dimension, the error using the ADXL105 accelerometer as a stand-alone system was proven to be as low as 1 cm over 10-15 seconds.

### **Future work**

This has only looked in to parts of the problem of constructing a local positioning system using a primary system and low-cost inertial components. Much work is still left, for instance testing and modelling of low-cost gyroscopes. These models could then be inserted into simulations similar to the ones described in this thesis to get a better estimation of the performance of an low-cost inertial system.

Magnetic heading systems could also be tested as extra means to determine the heading of the IMU in addition to the gyros. The systems should be tested, and experiments conducted to verify performance.

Accelerometer parameter estimation can also be done with an extended Kalman filter. Attempts were done to include this in this thesis, but suitable filters had not been developed before the end of the work. Parameter estimation using nonlinear Kalman filters or nonlinear observers could be an integrated part of the system.

If a model good model of the primary system is available, the sensor outputs from the IMU and the primary system can be integrated in a tightly coupled system to increase the performance, as described by Vik (2000). Thus experimental testing to look at the performance and make models of different primary systems may be a good idea.

## References

Bar-Shalom, Y., X.-Rong LI and Kirubarajan, T.: *Estimation with Applications to Tracking and Navigation*. Joh Wiley & Sons, USA (2001)

Barbour, N. and Schmidt, G.: *Inertial Sensor Technology Trends*, (1998)

Bortz, J. E.: *A new mathematical formulation for strapdown inertial navigation*. IEEE Transp. Aerosp. Electron. Systems, (1971).

Farrell, J.A. and Barth, M.: *The Global Positioning System & inertial Navigation*, McGraw-Hill, USA (1998)

Gelb, A: *Applied Optimal Estimation*, MIT, Cambridge, USA (1974)

Hill, W., Horowitz, P., : *The Art of Electronics* 2nd edition, Cambridge University Press, USA (1989)

*IEEE Standard Specification Format Guide and Test Procedure for Linear, Single-Axis, Nongyroscopic Accelerometers*. The Institute of Electrical and Electronics Engineers, Inc, New York, USA.(1999)

Jazwinski, A. H.: *Stochastic Processes and Filtering Theory*, Academic Press Inc, London, UK, (1970)

Lawrence, A: *Modern Inertial Technology; Navigation, Guidance, and Control*, Second Edition. Springer Verlag, New York, USA (1998)

Flannery B. P., Press W., Teukolsky S. A., Vetterling W. T.: *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, USA, (1992)

Savitzky, A and Golay M.J.E: *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*, Analytical Chemistry, Vol 36, pp 1627-1639 (1964)

Shin, E.-H. : *Accuracy Improvement of Low Cost INS/GPS for Land Applications*. Department of Geomatics Engineering, Calgary, Alberta, Canada., (2001).

Shuster, M., Briano B. and Kitchin C. : *Mounting Considerations for ADXL Series Accelerometers* Analog Devices, AN-379 Application note.

Squires , G. L. : *Practical physics*. Third Edition. Cambridge University Press, Cambridge, United Kingdom, (1968).

Titterton, D.H. and Weston, J.L.: *Strapdown Inertial Navigation Technology*, Peregrinis, London, on behalf of the Institution of Electrical Engineers, (1997)

Tyssø, A: *Automatiseringsteknikk*, NKI-Forlaget, Norway, (1992)

Vik, Bjørnar: *Nonlinear Design and Analysis of Integrated GPS and Inertial Navigation Systems* Doc. Ing. thesis, NTNU, Trondheim, Norway, (2000)

Williams, J.H. Jr: *Fundamentals of Applied Dynamics*, John Wiley & sons INC, USA, (1996)

## APPENDIX A Matlab Code

```
1 %%%%%%%%% Savitzky-Golay algorithm %%%%%%%%%
2 %Written by Morten Stakkeland, Physics Department, University of Oslo
3 function output=savgol(m,nr,nl,id,inp,tid)
4 %The Savitzky-Golay smoothing and differentiation algorithm
5 % 1. Calculate A as given in 14.8.6
6 sx=nr+nl+1; sy=m+1; le=length(inp);
7 A=zeros(sx,sy);
8 for i=1:sx
9     for j=1:sy
10        A(i,j)=(i-nr-1)^(j-1);
11    end
12 end
13 % 2. Calculate (A'A)^-1
14 mi=A*A;
15 B=inv(mi);
16 % 3 Calculate cn
17 c=zeros(sx,1);
18 for i=1:sx
19     for j=1:sy
20        c(i,1)=c(i,1)+B(id+1,j)*(i-nr-1)^(j-1);
21    end
22 end
23 le2=length(c);
24 %3 Construct matrix suitable for convolution
25 d=floor(le2/2);
26 e=zeros((d*2+le),1);
27 for i=1:le+d
28     if i<=d
29        e(i,1)=inp(1,1);
30        e(2*d+le-i+1,1)=inp(le,1);
31    else
32        e(i,1)=inp(i-d,1);
33    end
34 end
35 % 4 Calculate normalizaton constant
36 if id==0
37     fak=1;
38 else
39     cp=cumprod(1:id);
40     fak=cp(id);
41 end
42 mi=(-1)^(id);
43 nor=mi*(1/fak)*((tid(le)-tid(1))/(le-1))^(id);
44 % 5 Convolution
45 f=conv(e,c);
46 f=f/nor;
47 output=f(le2:(le2+le-1));
48
49
50 %%%%%%%%% least squares estimate %%%%%%%%%
51 function least_squares_estimate(E,T,g0)
52 % g0=9.81897919;
53 n=length(E);
54
55 W=zeros(1,n);
56 W=((1+abs(2105*sin(T)))/2106);
57 W2=W.^2;
58
59 %Summing
60 cost=g0*cos(T);
61 cost2=cost.^2;
62 cost3=cost.^3;
63 cost4=cost.^4;
64 cost5=cost.^5;
65 cost6=cost.^6;
66 scost=sum(cost./W2);
67 scost2=sum(cost2./W2);
68 scost3=sum(cost3./W2);
69 scost4=sum(cost4./W2);
70 scost5=sum(cost5./W2);
```

```

71 scost6=sum(cost6./W2);
72
73 sint=g0*sin(T);
74 sint2=sint.^2;
75 ssint=sum(sint./W2);
76 ssint2=sum(sint2./W2);
77
78 costsint=cost.*sint;
79 costsint2=cost.*sint2;
80 cost2sint=cost2.*sint;
81 cost2sint2=cost2.*sint2;
82 cost3sint=cost3.*sint;
83 cost4sint=cost4.*sint;
84 scostsint=sum(costsint./W2);
85 scostsint2=sum(costsint2./W2);
86 scost2sint=sum(cost2sint./W2);
87 scost2sint2=sum(cost2sint2./W2);
88 scost3sint=sum(cost3sint./W2);
89 scost4sint=sum(cost4sint./W2);
90
91 sig=sign(cost);
92 sig2=sig.^2;
93 sigcost=sig.*cost;
94 sigcost2=sig.*cost2;
95 sigcost3=sig.*cost3;
96 sigcost4=sig.*cost4;
97 sigcost5=sig.*cost5;
98 ssig=sum(sig./W2);
99 ssig2=sum(sig2./W2);
100 ssigcost=sum(sigcost./W2);
101 ssigcost2=sum(sigcost2./W2);
102 ssigcost3=sum(sigcost3./W2);
103 ssigcost4=sum(sigcost4./W2);
104 ssigcost5=sum(sigcost5./W2);
105
106 sigsint=sig.*sint;
107 sigsint2=sig.*sint2;
108 sigcostsint=sig.*costsint;
109 sigcost2sint=sig.*cost2.*sint;
110 sigcost3sint=sig.*cost3.*sint;
111 ssigsint=sum(sigsint./W2);
112 ssigsint2=sum(sigsint2./W2);
113 ssigcostsint=sum(sigcostsint./W2);
114 ssigcost2sint=sum(sigcost2sint./W2);
115 ssigcost3sint=sum(sigcost3sint./W2);
116
117 ecost=E.*cost;
118 ecost2=E.*cost2;
119 ecost3=E.*cost3;
120 esint=E.*sint;
121 ecostsint=E.*costsint;
122 esig=E.*sig;
123 esigcost=E.*sig.*cost;
124 esigcost2=E.*sig.*cost2;
125 se=sum(E./W2);
126 secost=sum(ecost./W2);
127 secost2=sum(ecost2./W2);
128 secost3=sum(ecost3./W2);
129 sesint=sum(esint./W2);
130 secostsint=sum(ecostsint./W2);
131 sesig=sum(esig./W2);
132 sesigcost=sum(esigcost./W2);
133 sesigcost2=sum(esigcost2./W2);
134
135 sw=sum(1./W2);
136
137 %%%%%%%%%%% Estimate Offset, Scale factor and misalignment %%%%%%%%%%%
138 M=[ sw    -scost  -ssint;
139     -scost  scost2  scostsint;
140     -ssint  scostsint  ssint2];

```

```

141 x=[ se;
142     -secost;
143     -sesint];
144 XM=M\X;
145 %Normal Matrix
146 A=M;
147 p=[XM(1),0,XM(2),0,0,0,XM(3),0,0];
148 res=(E-full_model(p,T)).^2;
149 sResM=sum(res);
150 s=sqrt(sum(res./W2)/n);
151 UM=sqrt(diag(inv(A))*s);
152
153 %%%%%%%%%%%%%% Offset, scale factor and 2. order nonlinear term %%%%%%%%%%%%%%
154 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
155 %Construct the linear system
156 M=[ sw      -scost   scost2;
157     -scost   scost2   -scost3;
158     scost2   -scost3   scost4];
159 x=[ se;
160     -secost;
161     secost2];
162
163 %Solve the linear system
164 X2=M\X;
165
166 %Construct the Normal matrix
167 A=M;
168
169 %Calculate the standard deviation of the residuals
170 p=[X2(1),0,X2(2),X2(3),0,0,0,0,0];
171 res=(E-full_model(p,T)).^2;
172 s=sqrt(sum(res./W2)/n);
173 sRes2=sum(res);
174 U2=sqrt(diag(inv(A))*s);
175
176 %%%%%%%%%%%%%% Offset scale factor and cross axis sensitivity %%%%%%%%%%%%%%
177 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
178 %Construct the linear system
179 M=[ sw      -scost   scostsint;
180     -scost   scost2   -scost2sint;
181     scostsint -scost2sint scost2sint2];
182 x=[ se;
183     -secost;
184     secostsint];
185
186 %Solve the linear system
187 XC=M\X;
188
189
190 %Construct the Normal matrix
191 A=M;
192
193 %Calculate the standard deviation of the residuals
194 p=[XC(1),0,XC(2),0,0,0,0,XC(3),0];
195 res=(E-full_model(p,T)).^2;
196 s=sqrt(sum(res./W2)/n);
197 sResC=sum(res);
198 UC=sqrt(diag(inv(A))*s);
199
200 %%%%%%%%%%%%%% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
201 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
202 %Construct the linear system
203 M=[ sw      -scost   ssigcost;
204     -scost   scost2   -ssigcost2;
205     ssigcost -ssigcost2 scost2];
206 x=[ se;
207     -secost;
208     sesigcost];
209
210 %Solve the linear system

```

```

211 XSA=M\X;
212
213 %Construct the Normal matrix
214 A=M;
215
216 %Calculate the standard deviation of the residuals
217 p=[XSA(1),0,XSA(2),XSA(3),0,0,0,0,0];
218 res=(E-full_model(p,T)).^2;
219 s=sqrt(sum(res./W2)/n);
220 sResSA=sum(res);
221 USA=sqrt(diag(inv(A))*s);
222
223 %%%%%%%%%%Offset, scale factor, 2. and 3 order nonlinear factors %%%%%%%%%%
224 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
225 M=[ sw      -scost   scost2  -scost3;
226     -scost   scost2   -scost3  scost4;
227     scost2  -scost3   scost4   -scost5;
228     -scost3  scost4   -scost5   scost6];
229 x=[ se;
230     -secost;
231     secost2;
232     -secost3];
233
234 %Solve the linear system
235 X3=M\X;
236
237 %Construct the Normal matrix
238 A=M;
239
240 %Calculate the standard deviation of the residuals
241 p=[X3(1),0,X3(2),X3(3),X3(4),0,0,0,0];
242 res=(E-full_model(p,T)).^2;
243 s=sqrt(sum(res./W2)/n);
244 sRes3=sum(res);
245 U3=sqrt(diag(inv(A))*s);
246
247 %%%%%%%%%%Offset, Offset asymmetry and scale factor%%%%%%%%%
248 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
249 %Construct the linear system
250 M=[ sw      -ssig    -scost;
251     -ssig    ssig2    ssigcost;
252     -scost   ssigcost  scost2];
253
254 x=[ se;
255     -sesig;
256     -secost];
257
258 %Solve the linear system
259 XA=M\X;
260
261 %Construct the Normal matrix
262 A=M;
263
264 %Calculate the standard deviation of the residuals
265 p=[XA(1),XA(2),XA(3),0,0,0,0,0,0];
266 res=(E-full_model(p,T)).^2;
267 s=sqrt(sum(res./W2)/n);
268 sResA=sum(res);
269
270 UA=sqrt(diag(inv(A))*s);
271
272 %%%%%%%%%% Offset, scale factor, misalignment sensitivity and 2. order coeff %%%
273 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
274 %Construct the linear system
275 M=[ sw      -scost   scost2  scostsint;
276     -scost   scost2   -scost3  -scost2sint;
277     scost2   -scost3   scost4   scost3sint;
278     scostsint -scost2sint scost3sint scost2sint2];
279 x=[ se;
280     -secost;

```



```

281     secost2;
282     secostsint];
283
284 %Solve the linear system
285 X2C=M\X;
286
287 %Construct the Normal matrix
288 A=M;
289
290 %Calculate the standard deviation of the residuals
291 p=[X2C(1),0,X2C(2),X2C(3),0,0,0,X2C(4),0];
292 res=(E-full_model(p,T)).^2;
293 s=sqrt(sum(res./W2)/n);
294 sRes2C=sum(res);
295 U2C=sqrt(diag(inv(A))*s);
296
297 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% All parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
298 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
299 %Construct the linear system
300 DI=diag([sw;
301         ssig2;
302         scost2;
303         scost2;
304         scost4;
305         scost6;
306         ssint2;
307         scost2sint2;
308         scost4]);
309 c1=[ 0;
310     -ssig;
311     -scost;
312     ssigcost;
313     scost2;
314     -scost3;
315     -ssint;
316     scostsint;
317     -ssigcost2];
318 c2=[ 0;0;
319     ssigcost;
320     -scost;
321     -ssigcost2;
322     ssigcost3;
323     ssigcost3;
324     -ssigcostsint;
325     scost2];
326 c3=[ 0;0;0;
327     -ssigcost2;
328     -scost3;
329     scost4;
330     scostsint;
331     -scost2sint;
332     ssigcost3];
333 c4=[ 0;0;0;0;
334     ssigcost3;
335     -ssigcost4;
336     -ssigcostsint;
337     ssigcost2sint;
338     -scost3];
339 c5=[ 0;0;0;0;0;
340     -scost5;
341     -scost2sint;
342     scost3sint;
343     -ssigcost4];
344 c6=[ 0;0;0;0;0;0;
345     scost3sint;
346     -scost4sint;
347     ssigcost5];
348 c7=[ 0;0;0;0;0;0;0;
349     -scostsint2;
350     ssigcost2sint];

```



```

420 disp(strcat('Offset=',num2str(XSA(1)), ' Std=',num2str(USA(1))));
421 disp(strcat('Scale factor=',num2str(XSA(2)), ' Std=',num2str(USA(2))));
422 disp(strcat('Scale factor asymmetry=',num2str(XSA(3)), ' Std=',num2str(USA(3))));
423
424 disp(strcat('Sum of residuals=',num2str(sResSA)));
425 disp(' ');
426 disp('All coefficients:');
427 disp(strcat('Offset=',num2str(X(1)), ' Std=',num2str(U(1))));
428 disp(strcat('Offset asymmetry=',num2str(X(2)), ' Std=',num2str(U(2))));
429 disp(strcat('Scale factor=',num2str(X(3)), ' Std=',num2str(U(3))));
430 disp(strcat('Scale factor asymmetry=',num2str(X(4)), ' Std=',num2str(U(4))));
431 disp(strcat('2. order coefficient=',num2str(X(5)), ' Std=',num2str(U(5))));
432 disp(strcat('3. order coefficient=',num2str(X(6)), ' Std=',num2str(U(6))));
433 disp(strcat('Misalignment factor',num2str(X(7)), ' Std=',num2str(U(7))));
434 disp(strcat('Cross axis sensitivity=',num2str(X(8)), ' Std=',num2str(U(8))));
435 disp(strcat('Odd second order coefficient=',num2str(X(9)), ' Std=',num2str(U(9))));
436 disp(strcat('Sum of residuals=',num2str(sRes)));
437
438
439 %%%%%%%%%%% REGDATA %%%%%%%%%%%
440 function [std,varresid,r2,cor,vcv,varinf]=regdata(param,yfit,ydata,jac)
441 % [std,varresid,r2,cor,vcv,varinf]=regdata(param,yfit,ydata,jac)
442 % Calculate and Plot regression statistics from lsqcurvefit.m
443 % OUT
444 % std -standard error of each parameter
445 % varresid- Variance of residuals
446 % r2 - R^2 Correlation coefficient
447 % cor - Correlation matrix for Parameters
448 % vcv - Variance Covariance Matrix for Parameters
449 % varinf- Variance inflation factors >10 implies Multicollinearity in x's
450 % IN
451 % param -Least squares parameter values
452 % yfit -Response fit using param to get yfit from lsqcurvefit use yfit=residual+ydata
453 % where residual is the error matrix from lsqcurvefit
454 % ydata -Response data
455 % jac -Jacobian value at Least squares parameter values
456
457 % Arthur Jutan Univ of Western Ontario Dept of Chemical Engineering
458 % ajutan@julian.uwo.ca
459 % Revised 11-20-98,5-19-99
460
461 e=yfit(:)-ydata(:); %error vectorize the Y matrix for multiple outputs
462 ss=e*e; % best sum of squares
463 m=length(yfit);n=length(param);
464 if (m~=n),varresid=ss./(m-n);else, var=NaN;end % variance of Residuals
465
466 % CALC VARIANCE COV MATRIX AND CORRELATION MATRIX OF PARAMETERS
467 %convert jac to full matrix for ver 5.3
468 jac=full(jac);%aj 99
469 xtx=jac*jac;
470 xtxinv=inv(xtx);
471
472 %calc correlation matrix cor and variance inflation varinf
473 varinf = diag(xtxinv);
474 cor = xtxinv./sqrt(varinf*varinf);
475
476 % Plot the fit vs data
477 % t=1:m;
478 % plot(t,ydata,'o',t,yfit,'g-')
479 % title(' ydata and ymodel versus observation number')
480 % xlabel(' observation number');
481 % ylabel(' ydata o and ymodel-')
482 % grid;
483
484 % disp(' Least Squares Estimates of Parameters')
485 % disp(param)
486 % disp(' correlation matrix for parameters ')
487 % disp(cor)
488 vcv=xtxinv.*varresid; % mult by var of residuals~=pure error
489 % disp('Variance inflation Factors >10 ==> Multicollinearity in x"s')

```

```

490 % disp(varinf)
491
492 %Formulae for vcv=(x'.vo.x)^-1 *sigma^2 where meas error Var, v=[vo]*sigma^2
493 std=sqrt(diag(vcv)); % calc std error for each param
494 % disp('2*standard deviation (95%CL) for each parameter')
495 % disp(2*std')
496 %Calculate R^2 (Ref Draper & Smith p.46)
497 r=corrcoef(ydata(:),yfit(:));
498 r2=r(1,2).^2;
499 % disp('Variance of Residuals ')
500 % disp( varresid )
501 % disp( 'Correlation Coefficient R^2')
502 % disp(r2)
503
504 %%%%%%%%%%%%% line test %%%%%%%%%%%%%
505
506 function [m,b]=line_test_Isq(X,E);
507 %function [m,b,S]=line_test_Isq(X,E);
508 n=length(X);
509 sx=sum(X);
510 sE=sum(E);
511 sx2=sum(X.^2);
512 sxE=sum(X.*E);
513 M=[ sx2 sx;
514     sx n];
515 x=[sxE;sE];
516 u=M\x;
517 m=u(1);
518 b=u(2);
519 st=sum((E-m*X-b).^2)/n;
520 S=diag(inv(M)*st);
521
522 %%%%%%%%%%%%% diskretiser %%%%%%%%%%%%%
523
524 function [F,D,G,Q]= diskretiser(A,L,C,QU,dt)
525 F=expm(A*dt);
526 si=size(A);
527
528 [tt,x]=ode45(@dudt,[0 dt],zeros(length(L),1),[],A,L);
529 D=x(length(x),:);
530
531 sx=si(1,1);sy=si(1,2);
532 A2=-A';
533 MID=C*QU*C';
534 AP=zeros(2*sx,2*sy);
535 for i=1:sx,
536     for j=1:sy,
537         AP(i,j)=A(i,j);
538         AP(i,j+sy)=MID(i,j);
539         AP(i+sx,j+sy)=A2(i,j);
540     end;
541 end;
542 F1=expm(AP*dt);
543 F12=zeros(sx,sy);F22=zeros(sx,sy);F22l=zeros(sx,sy);
544 F12=F1(1:sx,sy+1:2*sy);
545 F22=F1(sx+1:2*sx,sy+1:2*sy);
546 F22l=inv(F22);
547 FAK=F12*F22l;
548
549 Q=eye(sx,sy);
550 G=chol(FAK);
551 G=G';
552
553 function vu=dudt(t,y,A,L)
554 vu=zeros(length(L),1);
555 vu=expm(t*A)*L;
556
557
558 %%%%%%%%%%%%% Kalman filter %%%%%%%%%%%%%5
559

```

```

560 function Kalman_Markov()
561
562 stdsup=3e-2;
563 stdpos=1e-4;
564 beta0=301;
565 F=[0 1 0;0 0 1;0 0 -beta0];
566 L=[0;1;0];
567 G=[0;0;1];
568 H=[1 0 0];
569 R=(stdpos)^2; %Position sensor covarians
570 R2=(stdsup)^2; %Ultrasound covarians
571
572 [b,a]=butter(3,70/833.33);
573 freq=4;
574 beg=1000;
575
576 QU=(2.6e5*9.8*1E-6)^2; %The accelerometer noise spectral density
577 Has_done=0;
578 dt=0.0006;
579 [Fi,La,Ga,Q]=diskretiser(F,L,G,QU,dt);
580 n1=load('2.mat');
581 t=n1.dscapture.X.Data';
582 %u=accelerometer data
583 u=double(n1.dscapture.Y(1).Data');
584 z=double(n1.dscapture.Y(3).Data)*1E-6';
585 z=filfilt(b,a,z);
586 off=mean([u(1:1.8e4); u(3e4:end)]);
587 bn=2e4;nd=4.26e4;
588 h=42;fak=3.14;c=round(fak*h);
589 aa=savgol(2,h,h,2,z(bn:nd)),t(bn:nd));
590 vs=savgol(2,c,c,0,u(bn:nd),t(bn:nd));
591 p1=polyfit(aa(c:end-c),vs(c:end-c),1);
592 sc=p1(1);
593 z=z(beg:end-beg);
594 leng=length(z);
595 u=u(beg:leng+beg);
596 uz=(u-off)/sc;
597 Of=off; K=sc;
598
599 [bff,aff]=butter(1,150/833.33);
600 % uz=filfilt(bff,aff,uz);
601 meas=zeros(2,leng);
602
603 P0=[(stdpos)^2 0 0;0 (10000E-6)^2 0;0 0 (12425e-6)^2]; % Uncertainties in initial position
604 int1=1000;
605 %meas=1: primary system
606 meas(:,1:int1)=1;
607
608 meas(:,leng-int1:end)=1;
609 % meas(:,:)=1;
610
611 for i=int1+1:leng-int1-1
612     if rem(i,round(1/(freq*6e-4)))==0
613         %meas = 2: ultrasound measurement
614             meas(1,i)=2;
615         end
616         meas(2,i)=2;
617     end
618     int3=19000-beg; int4=10*1666;
619     meas(:,int3:int3+int4)=0;
620
621     Pp=zeros(3,3,leng); %covariance matrices
622     Pe=zeros(3,3,leng);
623     Pp(:,:,1)=P0;
624     Pe(:,:,1)=P0;
625
626     xp=zeros(3,leng);
627     xe=zeros(3,leng);
628     randn('state',sum(100*clock));
629

```

```

630 xe(1,1)=z(1);%initial position
631 xe(2,1)=(z(5)-z(3))/1.2E-3; %initial velocity
632 xe(3,1)=1;
633 xp(:,1)=xe(:,1);
634
635 K3=4.8e-5;
636 K2=-8.7e-6;
637 uz=(three_solve(Of,K,K2,K3,u));
638 % uz=two_solve(Of,K,K2,K3,u);
639
640 m=randn(1,leng)*stdsup;
641 z2=z+m;
642 for k=1:leng
643 % if(rem(k,1000)==0)disp(k);end;
644 if meas(1,k)==1
645 KG=k_gain_update(Pp(:,k),H,R);
646 xe(:,k)=xp(:,k)+KG*(z(1,k)-H*xp(:,k));
647 if k<leng
648 xp(:,k+1)=Fi*xe(:,k)+La*uz(k);
649 end
650 Pe(:,k)=(eye(3,3)-KG*H)*Pp(:,k);
651 elseif meas(1,k)==2
652 KG=k_gain_update(Pp(:,k),H,R2);
653 xe(:,k)=xp(:,k)+KG*(z2(1,k)-H*xp(:,k));
654 if k<leng
655 xp(:,k+1)=Fi*xe(:,k)+La*uz(k);
656 end
657 Pe(:,k)=(eye(3,3)-KG*H)*Pp(:,k);
658 else
659 %no measurement update
660 if k<leng
661 xp(:,k+1)=Fi*xp(:,k)+La*uz(k);
662 end
663 Pe(:,k)=Pp(:,k);
664 xe(:,k)=xp(:,k);
665 end
666 if k<leng
667 Pp(:,k+1)=Fi*Pe(:,k)*Fi'+Ga*Q*Ga';
668 end
669 end
670
671
672 %Smoothin part, use the Rauch Tung Striebel formulas from Gelb
673
674 Pb=zeros(3,3,leng);
675 xb=zeros(3,leng);
676 Pb(:,:,leng)=Pe(:,:,leng);
677 xb(:,leng)=xe(:,leng);
678
679 for k=1:leng-1
680 Ak=Pe(:,leng-k)*Fi*inv(Pp(:,leng-k+1));
681 xb(:,leng-k)=xe(:,leng-k)+Ak*(xb(:,leng-k+1)-xp(:,leng-k+1));
682 Pb(:,leng-k)=Pe(:,leng-k)+Ak*(Pb(:,leng-k+1)-Pp(:,leng-k+1))*Ak';
683 end
684
685
686 Li=0;
687 [q,qq,aa]=computer;
688 if aa=='L'
689 Li=1;
690 end
691 fil=fopen('xdir','w');
692 uu=[meas(2,:),'z',xb(1,:),xe(1,:),t(1:leng)];
693 fwrite(fil,leng,'double');
694 fwrite(fil,Li,'double');
695 fwrite(fil,uu,'double');
696 fclose(fil);
697
698
699 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% inertial algoritm %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

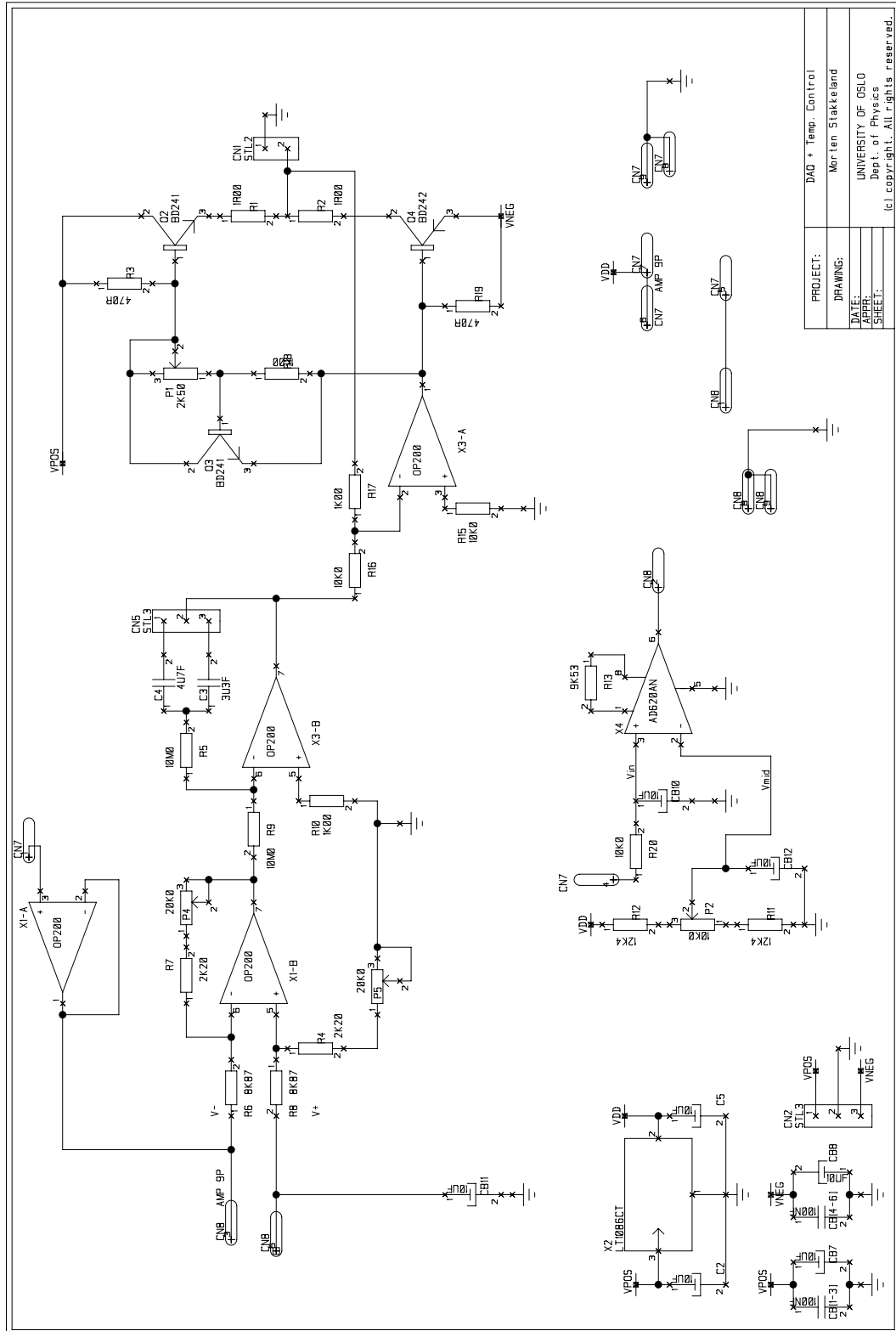
```

```

700
701 function [Cnbo,vko]=inertial_alg(fb,wb,Cnb,v0,dt,LA)
702 g0=[0;0;9.81897919];
703 latitude=LA*pi/180;
704 w0=(1+365.25)*2*pi/(365.25*24*3600);
705 w_in=w0*[cos(latitude);0;-sin(latitude)];
706 % w_in=w0*[0;0;-1];
707 OM_in=skew_sym(w_in);
708 leng=length(fb);
709 Cnbo=zeros(3,3,leng);
710 vko=zeros(3,leng);
711 Wb=zeros(3,3,leng);
712 for i=1:leng
713     Wb(:,i)=skew_sym(wb(:,i));
714 end
715 Cnbo(:,1)=Cnb(:,1);
716
717 inter=100;
718 ddt=dt/inter;
719 for i=2:leng
720
721     Cmid=RK(Cnbo(:,i-1),Wb(:,i-1),Wb(:,i),OM_in,ddt);
722     for j=2:inter
723         Cmid=RK(Cmid,Wb(:,i),Wb(:,i),OM_in,ddt);
724     end
725     Cnbo(:,i)=Cmid;
726 % Cnbo(:,i)=RK(Cnbo(:,i-1),Wb(:,i-1),Wb(:,i),OM_in,dt);
727 end
728
729 vko(:,1)=v0;
730 vko(:,2)=vko(:,1)+dt*(Cnbo(:,1))*fb(:,1)-2*cross(w_in,vko(:,1))+g0+(Cnbo(:,2))*fb(:,2)+g0-2*cross(w_in,vko(:,1)))/2;
731 vko(:,3)=vko(:,1)+dt*((Cnbo(:,1))*fb(:,1))+4*(+Cnbo(:,2))*fb(:,2)+(Cnbo(:,3))*fb(:,3))+6*g0/3;
732 for i=4:leng
733     vko(:,i-2)=vko(:,i-2)-2*dt*OM_in*vko(:,i-2);
734     vko(:,i)=vko(:,i-2)+dt*((Cnbo(:,i-2))*fb(:,i-2))+4*(+Cnbo(:,i-1))*fb(:,i-1)+(Cnbo(:,i))*fb(:,i))+6*g0/3;
735 end
736
737 function OM=skew_sym(v);
738 OM=[0 -v(3) v(2); v(3) 0 -v(1); -v(2) v(1) 0];
739
740 function DC=RK(Ck,Wm,Wp,Win,dt)
741 k1=dt*(Ck*Wm-Win*Ck);
742 k2=dt*((Ck+0.5*k1)*Wp-Win*(Ck+0.5*k1));
743 k3=dt*((Ck+0.5*k2)*Wp-Win*(Ck+0.5*k2));
744 k4=dt*((Ck+k3)*Wp-Win*(Ck+k3));
745 DC=Ck+k1/6+k2/3+k3/3+k4/6;
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761

```

# APPENDIX B The Analog Signal Conditioning Circuit





## APPENDIX C Extracts from the Accelerometer Datasheets

### ADXL105

- Range:  $\pm 5g$ .
- Nominal sensitivity: 250mV/g. Minimum 225mV/g, maximum 275mV/g
- Nominal zero g: 2.5V, bias:  $\pm 625mV$ .
- Nonlinearity: 0.2 \% of full-scale output.
- Alignment error:  $\pm 1^\circ$ .
- Cross axis sensitivity: 1 - 5 \%
- Noise Performance: 225 - 325  $\mu g/\sqrt{\text{Hz}}$
- Bias offset vs. temperature: approx. 1mg/ $^\circ C$
- Sensitivity change vs. temperature:  $\pm 0.5\%$ .
- Bandwidth: Adjustable, max=10kHz.
- Power 2mA. Vdd=5V. Source/sink min. 50 $\mu A$ , buffer may be required for some DCs. Output is ratiometric to VDD.
- Temperature range: 0-70 deg C. Ideal: 25 deg C.
- Temperature sensor:  $T(^{\circ}C) = (V_{out}-2.5V)/0.008V/^{\circ}C+25^{\circ}C$

### Crossbow CLX02TG3

- Range: 2g.
- Nominal sensitivity: minimum 767mV/g, typical 833mV/g and maximum 900mV/g
- Nominal zero g: 2.5V, 10mV
- Nonlinearity: 0.75-1.5% of FS.
- Alignment error: 1% of FS.
- Cross axis sensitivity: 1 - 5 \% of FS.
- Noise Performance: 20  $\mu g/\sqrt{\text{Hz}}$
- Bias offset vs. temperature: 170 $\mu V$
- Bandwidth: 800Hz.
- Power 2mA. Vdd=5V. Output is ratiometric to VDD.
- Temperature range: -40-85 deg C.
- Temperature sensor:  $T=(V_{out}-1.375V)*44.4 C/V$

## APPENDIX D The IEEE Accelerometer Model Equation

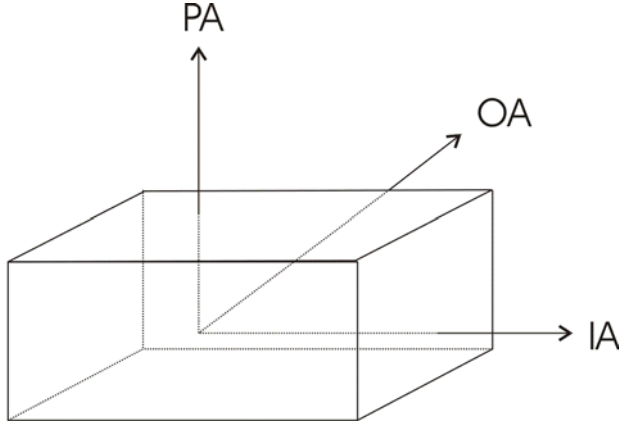


Figure (a): The principle axes of an accelerometer

A complete model of a non-gyroscopic inertial accelerometer as a function of the applied acceleration and rotation around the axes shown on figure is given as (IEEE Std 1293-1999):

$$E = K_1 \left\{ \frac{K'_0}{2} \cdot \text{sign}(a_i) + \left( 1 + \frac{K'_1}{2} \cdot \text{sign}(a_i) \right) a_i + K_{oq} a_i |a_i| + K_2 a_i^2 + K_3 a_i^3 + \sum_{n \geq 4} K_n a_i^n \right. \\ \left. + \delta_o a_p - \delta_p a_o + K_{ip} a_i a_p + K_{io} a_i a_o + K_{po} a_p a_o + K_{pp} a_o^2 + K_{oo} a_o^2 + K_{spin} \omega_i \omega_o \right. \\ \left. + K_{angacc} \dot{\omega}_o + \varepsilon \right\}$$

$E$  is the accelerometer output in units (V)

$\varepsilon$  is the measurement and process noise and unmodelled error (g or m/s<sup>2</sup>)

$K_1$  is the scale factor (units/g or units/m/s<sup>2</sup>)

$a_i, a_o, a_p$  is the applied acceleration along IA, PA and OA (g or m/s<sup>2</sup>)

$\text{sign}(a_i) = 1$  if  $a_i > 0$ ,  $-1$  if  $a_i < 0$  and  $0$  if  $a_i = 0$ .

$\omega_i, \omega_o, \omega_p$  are the angular velocities around the IA, PA and OA (rad/s)

$\dot{\omega}_i, \dot{\omega}_o, \dot{\omega}_p$  are angular accelerations around IA, PA and OA

$K_i$  is bias (g or m/s<sup>2</sup>)

$K'_0$  is bias asymmetry (g or m/s<sup>2</sup>)

$K'_1$  is scale factor asymmetry (dimensionless)

$K_{oq}$  is odd quadratic coefficient (g/g<sup>2</sup> or s<sup>2</sup>/m)

$K_2$  is second order coefficient ( $g/g^2$  or  $s^2/m$ )

$K_3$  is third order coefficient ( $g/g^3$  or  $s^4/m^2$ )

$K_n$  is ogher higher order coefficients,  $n=4,5,6\dots$  ( $g/g^n$  or  $s^{(2n-1)}/m^{(n-1)}$ )

$\delta_o, \delta_p$  are misalignments of the IA with respect to the input reference axis about the OA and PA (rad)

$K_{ip}, K_{io}, K_{po}$  are cross-coupling coefficients ( $g/g^2$  or  $s^2/m$ )

$K_{pp}, K_{oo}$  are cross axis nonlinearity coefficients ( $g/g^2$  or  $s^2/m$ )

$K_{spin}$  is spin correction coefficient

$K_{angacc}$  is angular acceleration coefficient.

### Modified version of the IEEE model equation

$$E = O + O_a \text{sign}(a_i) + K a_i + K_a \text{sign}(a_i) a_i + K_2 a_i^2 + K_3 a_i^3 + \sum_{n \geq 4} K_n a_i^n \\ + \delta_o a_p - \delta_p a_o + K_{ip} a_i a_p + K_{io} a_i a_o + K_{po} a_p a_o + K_{pp} a_o^2 + K_{oo} a_o^2 + K_{spin} \omega_i \omega_o \\ + K_{angacc} \dot{\omega}_o + \varepsilon$$

$E$  is the accelerometer output in units (V)

$\varepsilon$  is the measurement and process noise and unmodelled error (g or  $m/s^2$ )

$K$  is the scale factor ( $V/m/s^2$ )

$a_i, a_o, a_p$  is the applied acceleration along IA, PA and OA ( $m/s^2$ )

$\text{sign}(a_i) = 1$  if  $a_i > 0$ ,  $-1$  if  $a_i < 0$  and  $0$  if  $a_i = 0$ .

$\omega_i, \omega_o, \omega_p$  are the angular velocities around the IA, PA and OA (rad/s)

$\dot{\omega}_i, \dot{\omega}_o, \dot{\omega}_p$  are angular accelerations around IA, PA and OA

$O$  is the accelerometer offset (V)

$O_a$  is the offset asymmetry (V)

$K_a$  is the scale factor asymmetry ( $V/m/s^2$ )

$K_{oq}$  is odd quadratic coefficient ( $V/(m/s^2)^2$ )

$K_2$  is second order coefficient ( $V/(m/s^2)^2$ )

$K_3$  is third order coefficient ( $V/(m/s^2)^3$ )

$K_n$  is ogher higher order coefficients,  $n=4,5,6\dots$

$\delta_o, \delta_p$  are misalignments of the IA with respect to the input reference axis about the OA and PA ( $V \text{ rad}/(m/s^2)$ )

$K_{ip}, K_{io}, K_{po}$  are cross-coupling coefficients ( $V/(m/s^2)^2$ )

$K_{pp}, K_{oo}$  are cross axis nonlinearity coefficients ( $V/(m/s^2)^2$ )

$K_{spin}$  is spin correction coefficient

$K_{angacc}$  is angular acceleration coefficient.