Tamper Sensing using Low-Cost Accelerometers

TOBIAS RÅDESKOG

Master’s Degree Project
Stockholm, Sweden March 2012

XR-EE-SB 2012:003
Tamper Sensing using Low-Cost Accelerometers
KTH Master Thesis, Spring 2012

Tobias Rådeskog
Abstract

Certain security products, in this case sirens, must according to industry specifications generate a tamper warning in case the system is significantly moved from its place of mounting or subject to significant violence such as drilling. Currently, detecting removal from the place of mounting is typically done by mechanical switches placed between the siren and wall. This type of detection can be defeated by using thin blades and moreover complicates the process of mounting.

In this M.Sc. thesis carried out in collaboration with the Stockholm-based security equipment manufacturer Indusec AB, we investigate whether it is instead possible to use an inexpensive, but high-resolution three-axis micro-electromechanical (MEMS) type accelerometer in conjunction with an equally inexpensive microcontroller to detect such dislodgement. This is done by digital signal processing techniques implemented in the microcontroller.

Emphasis is put on finding reliable algorithms, which despite being reliable and fit for the purpose, do not use significant amounts of complex mathematics or memory, so that the detection can run on battery power with acceptable longevity and the algorithms implemented in the 8-bit microcontroller. For the sake of verification and analysis the algorithms were also implemented in MATLAB. The detection methods are primarily derived based on real-world trials and investigations.

The reliability of the developed detector algorithms is graphically presented as a functions of relevant detector parameters. The reliability estimates are strongly indicative of that an accelerometer in conjunction with an inexpensive microcontroller can be used as a reliable detector of system dislodging and chassis piercing by electric drills. The risk of misdetection from structural disturbances such as wall knocks or on site machinery seems to be small, with none of the test cases resulting in a false positive or false negative with the detector parameters properly set.
Foreword

This master thesis, made at the KTH in Stockholm, and in collaboration with the Stockholm-based security equipment developer Indusec AB, is intended to investigate possibilities for using low cost, high resolution micro-electromechanical (MEMS) accelerometers as tamper detection devices in sirens and similar systems.

During the thesis work, the questions have been numerous. Those questions have both regarded the general focus of the work, such as what type of movement should be classified as “benign” and “malign” and how they differ. Also there have been purely analytical questions - such as how software execution patterns in a highly memory and speed restricted microprocessor should be properly designed, how unwanted noise and drift in the accelerometer signals should be handled and what approaches to designing detector statistics could be feasible.

In this thesis report, I have aimed to answer the questions asked - all those answers eventually lead to a working implementation of tamper detection fit for the purpose: making it highly difficult to move or sabotage a siren or similar mounted to a surface without an alert being raised, while the likelihood of “mistriggerings” from external, benign vibrations within reason are small.

I would like to thank the following people and companies: my guide Asko Antikainen and CEO Johan Sandberg at Indusec AB, who helped me define and start the work; Ghazaleh Panahandehnigjeh and Magnus Jansson, my guide and examiner at KTH, respectively, who helped me with literature and general suggestions on how to improve the work. Also, there shall be thanks to the representatives at Freescale Semiconductor, whose names I cannot remember, for sending laboratory samples and software free of any cost.

Stockholm, the 20th of February, 2012

Tobias Rådeskog
Table of Contents

1 Introduction ................................................................................................................................. 1
  1.1 Background ............................................................................................................................ 1
  1.2 Motivation .............................................................................................................................. 2
  1.3 General thesis outline ............................................................................................................ 2
  1.4 Goals ..................................................................................................................................... 3

2 Platform and equipment ............................................................................................................... 4
  2.1 Microcontroller ..................................................................................................................... 4
  2.2 Accelerometer ......................................................................................................................... 4
  2.3 Personal computer software and development environment .................................................... 5
    2.3.1 Microcontroller IDE and compiler .................................................................................. 5
    2.3.2 Mathematics software ..................................................................................................... 5
  2.4 Lab platform .......................................................................................................................... 5
  2.5 Restrictions ............................................................................................................................ 6
    2.5.1 Component power consumption ...................................................................................... 6
      2.5.1.1 Microcontroller ......................................................................................................... 6
      2.5.1.2 Accelerometer ........................................................................................................ 8
      2.5.1.3 Total power calculations ........................................................................................ .9
    2.5.2 Mathematics capabilities ................................................................................................. 9
    2.5.3 Microcontroller memory model and associated restrictions ............................................10

3 Disturbance detection ................................................................................................................ 12
  3.1 Accelerometer theory ........................................................................................................... 13
  3.2 Intrinsic noise analysis ......................................................................................................... 14
  3.3 What is malign movement? ................................................................................................... 15
    3.3.1 General reasoning .......................................................................................................... 15
    3.3.2 Distinguishability ........................................................................................................... 16
    3.3.3 Prerequisites for reliable detection ................................................................................ 16
    3.3.4 Disturbance types of interest ......................................................................................... 17
  3.4 Prestudial reasoning .............................................................................................................. 19
    3.4.1 How disturbances affect the accelerometer (System model) .............................................19
      3.4.1.1 Sampling and oversampling ..................................................................................... 20
    3.4.2 Conditions at rest .......................................................................................................... 21
      3.4.2.1 The gravity, and tilt parameters ............................................................................... 22
      3.4.2.2 Gravity cancellation ................................................................................................. 22
      3.4.2.3 Thermal drift ........................................................................................................... 24
    3.4.3 Malign movement - Displacement ................................................................................ 24
    3.4.4 Malign movement - Dislodgement from surface (Tilt) .................................................... 27
    3.4.5 Malign movement - Handholding ................................................................................... 27
    3.4.6 Malign movement - Sawing and drilling ...................................................................... 28
    3.4.7 Malign movement - Blatant abuse ................................................................................. 28
    3.4.8 Benign movement - Beating mounting surface ............................................................. 28
    3.4.9 Benign movement - Slight earthquakes ....................................................................... 29
    3.4.10 Benign movement - Hi Fi systems and on site machinery ........................................... 29

4 Trials ........................................................................................................................................... 30
  4.1 General ................................................................................................................................ 30
  4.2 The setup .............................................................................................................................. 30
  4.3 Initial observations ............................................................................................................... 31
  4.4 Specific cases ....................................................................................................................... 32
    4.4.1 Malign movement - Displacement ............................................................................... 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2 Benign movement - Mounting surface disturbances</td>
<td>33</td>
</tr>
<tr>
<td>4.4.3 Malign movement - Removal from surface</td>
<td>34</td>
</tr>
<tr>
<td>4.4.4 Malign movement - Chassis piercing</td>
<td>34</td>
</tr>
<tr>
<td>4.4.4.1 Drilling</td>
<td>34</td>
</tr>
<tr>
<td>4.4.4.2 Sawing</td>
<td>35</td>
</tr>
<tr>
<td>4.4.5 Malign movement - Blatant abuse (Striking system with hammer)</td>
<td>36</td>
</tr>
<tr>
<td>4.4.6 Benign movement - On site machinery / Hi-Fi systems</td>
<td>37</td>
</tr>
<tr>
<td>4.4.7 Benign movement - Weak earthquakes</td>
<td>38</td>
</tr>
<tr>
<td>4.4.8 Conclusions about trials</td>
<td>38</td>
</tr>
<tr>
<td>5 Derived detectors</td>
<td>40</td>
</tr>
<tr>
<td>5.1 General</td>
<td>40</td>
</tr>
<tr>
<td>5.2 Unipolar Persistent Acceleration (UPA) detector</td>
<td>40</td>
</tr>
<tr>
<td>5.3 Magnitude-Persistence Detector (MPD)</td>
<td>41</td>
</tr>
<tr>
<td>5.4 Large transients</td>
<td>43</td>
</tr>
<tr>
<td>5.5 Tuning the detectors</td>
<td>43</td>
</tr>
<tr>
<td>5.6 Detection logic</td>
<td>45</td>
</tr>
<tr>
<td>6 Implementation</td>
<td>46</td>
</tr>
<tr>
<td>6.1 Software structure</td>
<td>46</td>
</tr>
<tr>
<td>6.1.1 Data flow</td>
<td>46</td>
</tr>
<tr>
<td>6.1.2 Execution flow diagram</td>
<td>46</td>
</tr>
<tr>
<td>6.1.3 Parameters in implementation used at demonstration</td>
<td>48</td>
</tr>
<tr>
<td>7 Results</td>
<td>49</td>
</tr>
<tr>
<td>7.1 General</td>
<td>49</td>
</tr>
<tr>
<td>7.2 Current draw</td>
<td>49</td>
</tr>
<tr>
<td>7.3 Detection functionality</td>
<td>49</td>
</tr>
<tr>
<td>7.3.1 Selecting appropriate MPD parameters</td>
<td>50</td>
</tr>
<tr>
<td>7.3.2 General detection reliability result</td>
<td>51</td>
</tr>
<tr>
<td>7.4 Parameter selection from trial results</td>
<td>52</td>
</tr>
<tr>
<td>7.5 UPA/MPD event distinguishability</td>
<td>53</td>
</tr>
<tr>
<td>7.6 Alternative MPD detectors</td>
<td>56</td>
</tr>
<tr>
<td>7.6.1 Windowed 3D RMS and windowed 3D MEAN</td>
<td>56</td>
</tr>
<tr>
<td>7.6.2 Windowed MPD</td>
<td>58</td>
</tr>
<tr>
<td>8 Conclusions and discussion</td>
<td>60</td>
</tr>
<tr>
<td>8.1 Conclusions</td>
<td>60</td>
</tr>
<tr>
<td>8.2 Discussion and future work</td>
<td>60</td>
</tr>
<tr>
<td>8.2.1 Changing used buffers</td>
<td>60</td>
</tr>
<tr>
<td>8.2.2 Changing microcontroller</td>
<td>60</td>
</tr>
<tr>
<td>9 Bibliography</td>
<td>61</td>
</tr>
<tr>
<td>10 Appendix - Functionality testing event list</td>
<td>64</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background

Indusec AB is the developer and manufacturer of a patented siren system, INFERNO, which upon triggering, gives off a sound so disturbing and loud that it is very hard to remain in the immediate vicinity of such a siren. The idea, of course, is to discourage thieves from remaining in the area and thereby preventing theft. While it may seem so at a first glance, the sound which emanates from the sirens does not cause hearing damage, something concluded in another KTH master thesis1. The siren functionality in itself, however, has nothing to do with this thesis work, but is just given as an example of in what application the final solution is likely to be used.

To reduce the possibility of someone sabotaging such a siren, there are some mechanisms in place. Integrated in the siren is a mechanical tamper detector, simply a switch which opens when the siren chassis cover is removed. Optionally available are external seismic detectors from e.g. Siemens such as model GM 770, popular when the sirens are used to defend against intrusion into shipping containers by different tools2. These detectors come at a great cost of several hundred dollars. We did not want to dig too deep into how these are designed because of the risk of patent violation, and moreover they are not made for the same purpose as the detector developed in this thesis (they monitor disturbances in the environment as opposed to direct disturbances and movement of a system.)

The switch in the siren, naturally, does not react on if the siren is mistreated or displaced. According to standardization bodies such as the National Standards Authority of Ireland3, sirens of the lowest security classification, given they offer protection against tampering, may not be moved or dislodged from the mounting place more than 1 cm. If this happens, the system should raise a tamper alert immediately. Such requirements are commonly fulfilled by putting tamper switches between the siren and the wall; these could be defeated with tools such as thin blades and make the process of mounting more complicated.

Micromachined (MEMS) accelerometers of a high resolution and quite high precision can today be bought for less than one U.S. dollar in volume. The INFERNO sirens essentially are already embedded systems with a Microchip PIC16 series microcontroller on the main board, which generates waveforms and acts as a DC/DC converter controller, as well as handling other system duties. The PIC microcontrollers have integrated analog-to-digital converters, as well as the industry-standard serial interfaces for integrated circuit intercommunication. This opens up for possibilities to integrate almost any available MEMS accelerometer available with the INFERNO siren circuit board. Cleverly written and well thought out software for the microcontroller would then be able to interpret the incoming accelerometer signals as either of a benign or malign characteristic.

---

1 Kjellström, 1997.
3 NSAI, 2009.
The purpose of the project is to find out what malign and benign disturbances are, and how the two kinds differ when it comes to affecting the accelerometer in the system. With this in mind we work out software for a particular PIC18 microcontroller which in an appropriate manner distinguishes between the two types of disturbances.

1.2 Motivation

This is an application which is interesting both for Indusec AB, as it would enable low cost tamper sensing in an integrated system without too many modifications. All that is required, should the solution be commercialized, is making room for an accelerometer on the circuit board and integrating the software which resulted from this thesis work. Academically it is also an interesting thesis project as a fair bit of analytical work and investigation is required to make for an acceptable result.

Tamper sensing using accelerometers also seems to be quite an unexplored field, with very little information apparently available from Internet searches. A car alarm based on detecting tilt has been proposed\(^4\), but there seems not to be much more than that. Freescale Semiconductor claims\(^5\) that “The latest trend in metering is to use advanced accelerometers to detect small degrees of movement and record the tamper event.”, in the context of protecting water meters from end-user tampering, but do not go into any details on how this should be done.

There has, however, been projects done on KTH which have at least some connection to the field of tamper sensing, including mathematical methods for detecting mechanical stationarity of an accelerometer\(^6\).

Freescale Semiconductor has also proposed some methods for inertial navigation from an accelerometer alone in an application note\(^7\); this paper was partially what gave birth to the idea of doing this thesis project and some inspiration has been taken from it.

1.3 General thesis outline

The project’s first part was deciding for an accelerometer and microprocessor, then making a lab platform by interconnecting the two, which could stream serial data to a PC, as well as benchmarking the platform (Chapter 2).

The solution methods for finding detection methods (chapters 3, 4 and 5) are essentially field sessions and investigation along with common-sense reasoning and some reading of research articles, and we make an attempt to decide what characterizes malign and benign movement, respectively. Mathematics fit for malignancy decision are then worked out and throughout the work it is kept in mind that the simpler mathematics the better, as long as detection stays reliable.

Lastly, it is decided how a program structure can be implemented in a chosen microcontroller that does

---

5 Freescale Semiconductor Website, 2011.
not use too much power for the calculations. The detection parameters are also calibrated, as well as specific aspects of detector performance explored (chapters 6 and 7). Some discussion and suggestions for improvement are given in Chapter 8.

1.4 Goals

The goal with the project is to make a working prototype of a tamper detector which could be integrated in a system similar to Inferno.

It should try to follow standards for tamper detection set in a specifications document by the NSAI\(^8\) with the exception that this solution is allowed to work only in a stationary environment; if used in a vehicle or equipment prone to generating vibrations, it must be turned off.

Keeping the system as mathematically simple and using as little memory as possible is of the essence to keep current draw low (< 1 mA) and keeping implementation possible on inexpensive microcontrollers. This must not, however, come at the cost of unreliable detection.

The main objects are:

- Creating a laboratory platform using the chosen platform and accelerometer, which can stream raw acceleration data to a PC.

- Do some prestudial reasoning about what acceleration patterns are expected in common cases of disturbances, to get a better scientific base when doing trials.

- Record true acceleration data from interesting cases, and from what is seen, derive detection methods which can be used, and also come to motives for why the derived detectors should be reliable.

- Implement the detection system in such a way that it gives room for other system features in the microcontroller, i.e. using interrupts to control the program flow.

- Trying out the finalized detectors and presenting different aspects of detector performance graphically.

- Suggesting some possible improvements by future work.

\(^8\) NSAI, 2009.
# Platform and equipment

## Microcontroller

Currently, at the core of an INFERNO siren is a Microchip PIC16F914 microcontroller. This is a very computationally weak 8-bit microcontroller, lacking any type of hardware multiplier and it is traditionally programmed “straight off” in assembly language. At the very beginning of this thesis project, the INFERNO system was undergoing a major revision, and it therefore seemed a good idea to change to the newer PIC18 platform, which is still an 8-bit processor, but it does have a hardware multiplier and an extended instruction set optimized for the ANSI C programming language (such a freeware compiler can be downloaded from Microchip). The unit cost for this microprocessor is even less than the one currently used in the system. Microchip offers a large variety of microcontrollers in the PIC18 family. The exact one chosen is not important as they are essentially code compatible, but with some peripheral hardware and memory size differences. For flexibility on the laboratory platform we wanted excessive resources and therefore chose the 28-pin PIC18F26K20 microcontroller, with all necessary hardware for interfacing with an accelerometer and personal computer, a 64 MHz internal oscillator, 64 kB of program memory and 3936 bytes of available RAM. It costs about USD $2 in quantities over 5000.

## Accelerometer

From the large IC manufacturers, Freescale had a very interesting offer and were willing to send free samples of the MMA8451Q, which costs $0.98 in quantities over 10000. It is a 14-bit accelerometer, with a finest resolution of 244µg/bit setting the range to the smallest possible (+/- 2 g), this range is used all the way throughout this thesis. Raw noise levels are essentially a few mg RMS in the typical case. The noise figures are studied more in detail later on in this thesis. The accelerometer chip has an integrated A/D converter and logic circuit, so it can be read digitally using the industry-standard I2C serial protocol, which the chosen microcontroller fully supports in hardware. The accelerometer can also be configured over I2C by writing to certain registers in it.

There are also a lot of special features in the accelerometer chip such as event interrupts, tilt detection, angle calculation, averaging factor, power modes etc. The only ones of those we plan to use are: setting sampling rate and averaging factor (for noise reduction), and interrupt on data available (interrupt pin on accelerometer signals to processor that new data is available to be read). The reason for using as few vendor specific features as possible is that it was preferable the solution be remain vendor and equipment neutral – the accelerometer should be easily replaceable with another without modifying the system too much.
2.3 Personal computer software and development environment

2.3.1 Microcontroller IDE and compiler

Microchip offers the MPLAB IDE environment, for programming and debugging circuits in Microsoft Windows, and the MPLAB C18 (ANSI C) compiler for free; even commercial use is allowed. The choice simply fell on this because it was the most obvious for the application. There are third-party compilers available as well but having the compiler come from the same corporation that created the microprocessor is probably a good thing for the sake of reliability.

A “PICkit 3” circuit programmer and debugger was obtained by Indusec for the project; it is the actual hardware programmer intended for use with MPLAB IDE and can write the program memory of all current PIC microcontrollers.

The microcontroller supports ICSP (In Circuit Serial Programming) - this means the program memory is written by a serial stream to a set of pins on the chip. This means it does not have to be removed from the socket when programming, which greatly simplifies development work and the testing of new ideas.

2.3.2 Mathematics software

A choice of software for general analysis, algorithm testing and data plots was not hard to make because of this author’s extensive usage of it in his prior studies. The choice is MATLAB 7.

2.4 Lab platform

The accelerometer and microcontroller were set up and interconnected in the simplest fashion possible using a so called prototyping board with hole spacing for DIP packages. The accelerometer was set up in the suggested fashion shown in its datasheet, however with slight, insignificant deviations from the suggested decoupling capacitor values, which in practice should have no effect. On the laboratory board three LEDs were also placed (green, yellow and red) for indicating detection status; these are very useful for debugging purposes. There is also a programming connector, a Seiko 3.3 linear voltage regulator, reverse-polarity protection (rectifier diode 1N4148) and battery contact for a 9-volt battery. The created board is shown in Figure 1.

In the implementation of the lab platform, code execution was initially continuous and not interrupt driven, as all that was wanted was really to stream acceleration data to a PC and study patterns. Unfortunately, Freescale’s own laboratory platform left much to be desired, and it was apparently impossible to even save the raw acceleration data as a text file for analysis, this was one of the reasons for implementing a test platform from scratch so early in the project.

The roughly coded, power hungry and unoptimized laboratory platform was, throughout the project, successively improved to yield the final solution.

---

9 MMA8451Q datasheet, page 7.
2.5 Restrictions

2.5.1 Component power consumption

2.5.1.1 Microcontroller

The INFERNO system is not allowed to consume more than about 1.5 mA when idle, as it has a 1800 mAh backup battery which should be able to power the system for up to a month\(^\text{10}\) in case the power line goes out, and also of course be able to fire off the alarm for at least three minutes in case it gets triggered; during an alarm the current draw is approximately 2 A.

System features of the INFERNO make use of roughly 500 µA while idle on the current model, to comply with a current draw less than 1.5 mA it is therefore a recommendable goal to keep the average current draw of the tamper detection functionality less than 1 mA. With this in mind, it was very useful to look at the current draw figures of the processor at different speed settings for the built-in oscillator system in the PIC18 chip. Unfortunately, this is not well defined in the datasheet (some figures at the highest and lowest speeds available are given), so further investigation required the use of a high...
precision multimeter with current measurement capability.

For these measurements a recently calibrated Agilent U1252A was used. It has an alleged bandwidth of only 250 kHz, so the microcontroller chip was decoupled with µF-size electrolytic capacitors as well as nF-size ceramics to filter out high frequency current pulses. Hopefully, the values are somewhat truthful but what is interesting here is the current draw on different clock speeds relative to each other, not the absolute numbers, as current draw could vary by tens of percent between parts, according to Microchip.

What can be noted from the measurements, presented in Table 1, is that the current draw budget, that is, frequency per current, gets generally better the higher the processor frequency is. In the processor’s “sleep” mode - when it is turned off and can be awoken only by an external event, it draws virtually no current (< 1 µA). In many cases it is therefore much better to do the required work, and then put the processor in “sleep” mode, rather than running it at a continuously low frequency to save power.

If this is done it is however important that the processor can switch between a chosen high frequency and “sleep” mode quickly, without requiring too much power during the “wake-up” and “go-to-sleep” phases. Laboratory testing with a digital sampling oscilloscope did show that this could possibly be the case as long as the microcontroller’s frequency-quadrupling PLL(Phase Locked Loop) functionality is not used. Without it, the transition between “sleep” to 16 MHz and vice versa takes about 10 µAs. With it enabled (transition between “sleep” and 64 MHz) a transition takes almost 1 mAs, which is not acceptable. Probably, it takes a lot of extra time for the PLL to stabilize. So 16 MHz is the highest allowable clock frequency in the system, which results in 4 MIPS since processor instructions are run through at 1/4 of the clock frequency due to the lack of a pipeline in the PIC18 architecture.

<table>
<thead>
<tr>
<th>Frequency (kHz)</th>
<th>Current (µA)</th>
<th>kHz/µA</th>
<th>Oscillator type</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>36</td>
<td>0,86</td>
<td>LFINTOSC</td>
</tr>
<tr>
<td>250</td>
<td>412</td>
<td>0,61</td>
<td>HFINTOSC/64</td>
</tr>
<tr>
<td>500</td>
<td>473</td>
<td>1,06</td>
<td>HFINTOSC/32</td>
</tr>
<tr>
<td>1000</td>
<td>591</td>
<td>1,69</td>
<td>HFINTOSC/16</td>
</tr>
<tr>
<td>2000</td>
<td>827</td>
<td>2,42</td>
<td>HFINTOSC/8</td>
</tr>
<tr>
<td>4000</td>
<td>1284</td>
<td>3,12</td>
<td>HFINTOSC/4</td>
</tr>
<tr>
<td>8000</td>
<td>2282</td>
<td>3,51</td>
<td>HFINTOSC/2</td>
</tr>
<tr>
<td>16000</td>
<td>3673</td>
<td>4,36</td>
<td>HFINTOSC</td>
</tr>
<tr>
<td>64000</td>
<td>13600</td>
<td>4,71</td>
<td>HFINTOSC, 4X PLL</td>
</tr>
</tbody>
</table>

Table 1: Current draw at different oscillator speeds of the microcontroller. 3.3 V / 23ºC. Absolute accuracy is questionable. The measured current draw at 16 MHz and 1 MHz is very consistent with what is noted in the datasheet.

12 Unless one could use huge buffers before processing data. Considering the response time for detection must be somewhat quick (< 1 s) this would not be acceptable.
13 Million Instructions Per Second.
2.5.1.2 Accelerometer

The accelerometer draws a maximum of 165µA current, according to the datasheet. It could prove valuable to stay at this consumption all the time as good noise reduction is achieved by oversampling as much as possible at the selected data output rate. It is however possible to get the current draw down to 50µA and even below, but considerable sacrifice of precision and a higher risk of aliasing could be the cost because less oversampling is done\textsuperscript{14}. In the power model the maximum current draw of 165µA is assumed.

I\textsuperscript{2}C bus

With the context being current draw figures measured in fractions of mA, this may unfortunately be a major contributor to power consumption, since the bus (both data and clock lines) are driven high by 4.7 kΩ pull-up resistors hardwired to the system voltage of 3.3 V. This is required by the specification for the high-speed (400 kHz) I\textsuperscript{2}C bus used in this case. When the bus is pulled low current runs through the device asserting it low, via those resistors.

![Figure 2: Simplified diagram of the I\textsuperscript{2}C bus connection between the accelerometer and microcontroller.](image)

Assuming the worst case, that the lines are low during the entire duration of a byte transaction, a rough calculation is given below:

\[ I_{LOW} = \frac{3.3}{4700/2} = 1404 \, \mu A \] - this is the current running through the resistors when both lines are low.

\[ T_{BIT} = \frac{1}{400000} = 2.5 \, \mu s \] - the cycle time to transmit one bit at I\textsuperscript{2}C High Speed (400 kHz).

\[ T_{BYTE} = T_{BIT} \cdot 10 \] - one byte is 8 bits; we add two bits extra for the Start & Stop condition in I\textsuperscript{2}C.

\[ T_{BYTE} \cdot I_{LOW} \approx 35 \, nAs/byte \] - the current draw times time per byte.

In this case we have two bytes per channel for the 14-bit data and three channels, i.e. six bytes per complete dataset.

\[ I_{LOW} \cdot T_{dataset} = 210 \, nAs \]

Even multiplying by a good factor (doubling) for margins, this is still only a quite a minor contributor to

\textsuperscript{14} MMA8451Q datasheet; p.45.
current draw. It is assumed to be 500 nA/Hz in the current draw model (25µA at 50 Hz).

### 2.5.1.3 Total power calculations

The current draw may not be larger than $I_{\text{LIMIT}}$:

$$I_{\text{MCU}} + I_{\text{ACCELEROMETER}} + I_{\text{I2C}} < I_{\text{LIMIT}}$$

$$I_{\text{MCU}} \approx 3700 \times \left(\frac{\%\text{dutytime}}{100}\right) + 10 \times [\text{frequency}] \, \mu A$$

$$I_{\text{ACCELEROMETER}} \approx 165 \, \mu A$$

$$I_{\text{I2C}} \approx 25 \, \mu A$$

Hence the power consumption can be approximated as:

$$\{165 + 25 + 3700 \times \left(\frac{\%\text{dutytime}}{100}\right) + 10 \times [\text{frequency}]\} \, \mu A$$

We can here see that a sampling frequency of 100 Hz would consume the entire power budget from changing the power state for data acquisition. 50 Hz, the next step down, is however a feasible frequency, so solving the equation

$$165 + 25 + 3700 \times \left(\frac{\%\text{dutytime}}{100}\right) + 10 \times 50 = 1000$$

gives the maximum allowable processor duty time. Solving for $\%\text{dutytime}$ gives 8.38.

Generally: To comply with the power consumption limits, it is not allowable for the processor to run at 16 MHz speed for more than approximately 8% of the time on average. This gives room for 320k cycles per second. Benchmarking this in practice can be done by setting a processor pin high when the processor wakes up, and setting it low just before it goes to sleep. Measuring the duty cycle and frequency of the output signal then gives the current draw from the formula for $I_{\text{MCU}}$.

### 2.5.2 Mathematics capabilities

Since the processor is an 8-bit processor, with merely an 8-bit X 8-bit hardware multiplier as the accelerating mathematics feature, and the compiled software will mostly handle 16-bit data in the signal processing routines (due to the resolution of the accelerometer), it was suspected that the available computing power for signal processing would prove to be rather vile. To get a grasp of what the processor can handle, some common mathematics operations were carried out in C, compiled and benchmarked using the simulation tool in MPLAB IDE. The results are presented in Table 2.
Table 2: Different mathematics operations benchmarked using the simulation "stopwatch" tool in the development environment.

As expected, mathematics operations handling datatypes longer than 8 bits generally require hundreds of cycles. The exception (perhaps unsurprisingly) are the 16-bit comparisons, these could be very useful for algorithms making use of thresholds.

4 million clock cycles are available per second at the maximum speed. Considering a high incoming sample speed of 150 Hz (3 channels * 50 Hz), roughly 26000 cycles per sample would be available; about 100 floating point multiplications are then available to handle each and every incoming sample at the highest expected processing speed. This is hardly impressive performance but very typical for a modern 8-bit microcontroller; it is however clear that this amount of processing power should be able to do at least simple interpretation work or maybe even simple transforms.

A simple, completely unoptimized floating point implementation\textsuperscript{16} of a 64-point FFT was actually implemented in the microprocessor. It turned out this takes 672k cycles per run (which would be 2016k cycles for all three channels simultaneously). So while it is out of the power budget to run an FFT continuously, it would actually be possible to do so momentarily up to a sampling speed of approximately 100 Hz if necessary.

2.5.3 Microcontroller memory model and associated restrictions

The microcontroller is equipped with 64 kB of FLASH program memory, which should be capable of running this project and house fairly complex algorithms if needed.

The RAM memory (SRAM) is tightly connected to and clocked with the same speed as the processor,

\begin{table}
\begin{tabular}{|l|l|}
\hline
Maths operation: datatype(no.bits) & Clock cycles \\
\hline
char(8)×char(8)=short(16) & 70 \\
char(8)+char(8)=short(16) & 24 \\
char(8)-char(8)=short(16) & 24 \\
short(16)×short(16)=long(32) & 108 \\
short(16)+short(16)=long(32) & 41 \\
short(16)-short(16)=long(32) & 45 \\
short(16)/short(16)=long(32) & 264 \\
short(16)<short(16)=boolean(1) & 20 \\
short(16)>short(16)=boolean(1) & 21 \\
short(16)==short(16)=boolean(1) & 10 \\
float(32)×float(32)=float(32) & 250 \\
float(32)/float(32)=float(32) & 890 \\
float(32)+float(32)=float(32) & 277 \\
float(32)-float(32)=float(32) & 230 \\
float(32)\textsuperscript{2}=float(32) & 1500 \\
\sin(float(32))=float(32) & 5000 \\
64-point FFT (float(32))=float(32) & 672000 \\
\hline
\end{tabular}
\end{table}

\textsuperscript{16} Smith, 2011. Chapter 12.
which means data can often be had bytewise in a single instruction. However, the processor addresses
the RAM with only 8 bits, which means only 256 bytes of the 3936 bytes would be addressable - if it
were not for the fact that the memory is segmented into 16 “banks”. To access RAM in a certain bank, a
register for bank selection must first be written with the correct bank number; e.g. to access byte #256,
byte #0 should be accessed after first selecting bank #1. The C compiler handles all this automatically
of course, but it brings a small time overhead to RAM accesses. Also, the compiler restricts the size of a
data object, such as an array, to one bank (256 bytes), something important to consider when
implementing buffer-like data structures such as the queues which are used in the system
implementation.

Moreover, it can be said that 4kB of RAM is a quite scarce amount of memory when analysis of high-
frequency sampled data is involved - considering the sample size and order of expected sampling speeds
it is used up in seconds. Heuristics could therefore be used to characterize the history of disturbances.
One thinkable method is to segment the sampled data into “analysis frames” (some fraction of seconds)
and only store some basic statistical data (standard deviation, mean, or values given an autoregressive
model, etc.) about what happened in those frames, if turns out to be necessary.
## 3 Disturbance detection

<table>
<thead>
<tr>
<th>Term</th>
<th>Unit</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vec{a} = [a_x, a_y, a_z] )</td>
<td>( g )</td>
<td>Apparent measured acceleration on each of the three axes, that is, what the accelerometer measures.</td>
</tr>
<tr>
<td>( \vec{a}_{avg} )</td>
<td>( g )</td>
<td>Gravity cancellation vector. Intended to remove the gravity component before detection. Calculated by a filter or some heuristic.</td>
</tr>
<tr>
<td>( \vec{A} = [A_x, A_y, A_z] )</td>
<td>( g )</td>
<td>The acceleration after offset null is subtracted i.e. ( \vec{A} = \vec{a} - \vec{a}_{avg} ).</td>
</tr>
<tr>
<td>( \vec{e}_x, \vec{e}_y, \vec{e}_z )</td>
<td>Dimensionless</td>
<td>The directions of the accelerometer measurement axes (defines the coordinate system). Length of each always 1, and any axis always perpendicular to another.</td>
</tr>
<tr>
<td>( \vec{g} )</td>
<td>( g )</td>
<td>The direction and amount of the gravity relative to the coordinate system as described above. Length approximately 1.</td>
</tr>
<tr>
<td>( \vec{q} = [q_x, q_y, q_z] )</td>
<td>( g )</td>
<td>Auxiliary acceleration relative to the measurement axes (the acceleration measured which is a result of actual disturbances, not gravity).</td>
</tr>
<tr>
<td>( \sigma_{RAW} )</td>
<td>( g )</td>
<td>Standard deviation of the raw sampling noise from the accelerometer.</td>
</tr>
<tr>
<td>( ODR )</td>
<td>Hz</td>
<td>Output data rate from the accelerometer, i.e. the rate at which sample sets ( \vec{a} ) are created.</td>
</tr>
<tr>
<td>( OSR )</td>
<td>Hz</td>
<td>Oversampling ratio. Can be set by a command to the accelerometer. High ratios reduce sampling noise but bandwidth as well. Oversampling means using several samples to create one mean value as output, see Chapter 3.4.1.1.</td>
</tr>
<tr>
<td>( n_e )</td>
<td>( g )</td>
<td>Standard deviation of the noise after oversampling. In theory the same as ( \sigma_{RAW} / \sqrt{OSR} ) (seems consistent with reality).</td>
</tr>
</tbody>
</table>

*Table 3: Table with explanation of terms that are used in Chapter 3 and 4*

---

17 Note: \( g \) is a unit of acceleration, multiples of Earth’s gravity acceleration constant. \( 1 \) \( g \) \( \approx 9.8 \) m/s². In some cases, where noted, acceleration is given in acceleration counts instead of \( g \)'s (1 acceleration count = \( 244 \) \( \mu g \)).
3.1 Accelerometer theory

It is necessary to understand some accelerometer theory before progressing: what quantity a MEMS accelerometer actually measures and how to interpret the resulting data.

While the precise internal structure of the chosen accelerometer from Freescale Semiconductor is unknown, a MEMS accelerometer essentially works by measuring the position of a test mass loaded in a spring-like structure. This is done by measuring the capacitance in the structure, which changes as the mass moves.18

Conditioning of the signals is then carried out by the accelerometer chip to try and compensate for noise and nonlinearities. The result is a momentaneous measure on how much force via acceleration \((F = ma)\) was applied to the test mass, which in a MEMS accelerometer is very small (on the order of micrograms). Very little mechanical energy input hence needs to propagate to the accelerometer for a disturbance to be recorded. This means, that for the accelerometer to truthfully record the acceleration of the object it is mounted to, it would be required that said object and the mounting to it is infinitely stiff and unable to transfer vibrations.

What the accelerometer measures is the so called proper acceleration, that is, the acceleration relative to a freefall 19:

\[ [a_x, a_y, a_z] \] - where the different components are the acceleration components in the directions of which the test masses are mounted:

---

18 Andrejašič, 2008.
where · denotes the dot product.

It can be noted that the test mass axes are in theory perpendicular to each other, i.e. $\vec{e}_x \cdot \vec{e}_y = 0$, $\vec{e}_y \cdot \vec{e}_z = 0$ and $\vec{e}_x \cdot \vec{e}_z = 0$. As far as acceleration is concerned there is hence no degree of freedom, any acceleration should affect at least one of the axes. However, what is not detectable is if merely the measurement axes $\vec{e}_x, \vec{e}_y, \vec{e}_z$ change, that is, the accelerometer is rotating around its own axis, other than that it will affect the projection of gravity if not at least two axes are perpendicular to it during the rotation. True inertial measurement units solve this problem by beside the accelerometer also employing an angular detector\textsuperscript{20}. This project only has a triple axis accelerometer available.

### 3.2 Intrinsic noise analysis

To verify the noise figures in the datasheet and see what characteristics the sampling noise has, the accelerometer was run horizontally oriented in room temperature and at rest for approximately 1 minute at 50 Hz and the highest possible oversampling ratio at the frequency (OSR=32). The test was repeated five times at hourly intervals and at different orientations of the system each time.

These are plots of $A_x, A_y, A_z$, respectively. It can be noted that the well known Gaussian "bell shape" appears to (roughly) be the distribution on all three channels. The estimated autocorrelation functions also shows low values for all time shifts but zero, which means the noise seems to be white\textsuperscript{21}. The noise

\begin{align*}
    a_x &= \vec{e}_x \cdot (\vec{g} + \vec{q}) \\
    a_y &= \vec{e}_y \cdot (\vec{g} + \vec{q}) \\
    a_z &= \vec{e}_z \cdot (\vec{g} + \vec{q})
\end{align*}

\textsuperscript{20} Skog et al, 2010.

\textsuperscript{21} Händel et al, 2002; p. 59.
therefore could be said to resemble white gaussian noise (WGN).

<table>
<thead>
<tr>
<th>Trial no.</th>
<th>sX (mg)</th>
<th>sY (mg)</th>
<th>sZ (mg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0,457</td>
<td>0,467</td>
<td>0,41</td>
</tr>
<tr>
<td>2</td>
<td>0,451</td>
<td>0,501</td>
<td>0,421</td>
</tr>
<tr>
<td>3</td>
<td>0,456</td>
<td>0,47</td>
<td>0,419</td>
</tr>
<tr>
<td>4</td>
<td>0,455</td>
<td>0,48</td>
<td>0,413</td>
</tr>
<tr>
<td>5</td>
<td>0,473</td>
<td>0,475</td>
<td>0,417</td>
</tr>
<tr>
<td>Mean</td>
<td>0,4584</td>
<td>0,4786</td>
<td>0,416</td>
</tr>
</tbody>
</table>

Table 4: Standard deviations of noise trials. The gray area contains the average standard deviation of the three channels.

An oversampling ratio (OSR) of $N$ applied to a series of independent samples causes the standard deviation to be divided by the factor $\sqrt{N}$. Assuming the observed noise is WGN would imply independence.

The statistics in Table 4 were generated from data at OSR=32, the average standard deviation is 0,45 mg; this is indicative of the standard deviation of the raw samples from the accelerometer was

$$\sigma_{\text{RAW}} = 0,45 \text{ mg} \cdot \sqrt{32} \approx 2,55 \text{ mg}.$$

(3.2)

Trying out realizations with OSR=2, the average standard deviation, similarly compensated with a $\sqrt{2}$ factor turned out to be 2,77 mg.

Both these values fairly well match the noise RMS figure given by Freescale, which is 2,63 mg (to within 10%). Thus we conclude that the noise can be seen as approximately white Gaussian and have estimated the standard deviation of the raw intrinsic sampling noise in the accelerometer.

### 3.3 What is malign movement?

#### 3.3.1 General reasoning

This whole thesis is about finding algorithms that will classify measured acceleration patterns as either benign or malign, and it is therefore impossible to progress without having at least some crude definition of what malign disturbances are.

Essentially it is likely to be impossible to make an accelerometer-based solution work reliably and classify malign disturbance from benign in all cases, even so with a very sophisticated implementation of artificial intelligence. The reason is that the accelerometer is the only input. While acceleration patterns

22 Smith, 2011; chapter 2.
23 Tuck, 2010.
can say a lot about the nature of the disturbances experienced, it hardly tells anything about the environmental context the siren is mounted in.

### 3.3.2 Distinguishability

The more the detector environment is restricted, the easier the methods of detection become. The easiest case would be allowing no disturbances at all and letting the system run a continuous threshold detection; the hardest case would be a detector which works in almost all conditions, including vehicles, this is likely to require an advanced AI and other types of sensors than just accelerometers.

This project is however focused at finding a solution which would work reasonably well as a security feature in a supposedly stationary environment. To progress, some prerequisites about where the detection should work reliably has to be set up, and what types of disturbances should be detected and not.

### 3.3.3 Prerequisites for reliable detection

The system casing is tightly mounted to (and hence mechanically coupled with) a stable stiff or semi-stiff passive wall (not that of a machine prone to vibration), such as hard metals or gypsum. It may be mounted either with strong magnets or screws, as is commonly the case with INFERNO sirens. The printed circuit board is stiffly mounted within the siren enclosure (freedom of movement should be kept at a minimum). The circuit board should also be approximately parallel to the mounting surface. The system shall be used in a stationary environment (no vehicles). It can be noted that ISEN sets no requirement on methods of mounting, just that "The warning device shall be mounted in accordance with the manufacturer's instructions"\(^{24}\), we interpret this as that it is actually possible to exclude mobile usage.

The system should be highly sensitive to movement and dislodging from the place where it is mounted, according to ISEN a maximum of 10mm perpendicular movement from the surface is allowed. It should also be sensitive to abuse which could harm the siren, such as drilling with a 4mm drill, also according to ISEN requirements.

The system should, to the extent that seems possible, not raise an alert from patterns arising from phenomena which do not result in movement of the siren nor risk harming it. Most notably this includes bumps and shocks in the wall the siren is mounted on, nearby doors opening and closing, nearby heavy duty machines or Hi-Fi systems, and optimally even weak earthquakes.

Because of these given prerequisites the trials are mainly carried out in situations which conform to them. An illustration of how a common test setup looks physically is given in Figure 5.

\(^{24}\) NSAI, 2009; page 16.

- 16 -
3.3.4 Disturbance types of interest

After discussing with Indusec AB, as well as taking into account the requirements set by the NSAI, possible interesting disturbances of a benign and malign nature are tabulated in Table 5 and Table 6.

Figure 5: Typical environment fulfilling the prerequisites fit for trials.
<table>
<thead>
<tr>
<th>Type of benign disturbance</th>
<th>Discovered characteristics &amp; comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beating on mounting surface</td>
<td>Transient behavior around equilibrium. See Chapter 4.4.2.</td>
</tr>
<tr>
<td>Weak beating of siren enclosure</td>
<td>Transient behavior around equilibrium. See Chapter 4.4.2.</td>
</tr>
<tr>
<td>Slight earthquakes (seismic activity)</td>
<td>Infeasible to simulate. Literature gives hints to that a separate filtering of earthquakes might not even be necessary. See Chapter 4.4.7.</td>
</tr>
<tr>
<td>On-site machinery: washing machines, powerful Hi-Fi systems etc.</td>
<td>Transient behavior around equilibrium and relatively weak amplitude. See Chapter 4.4.6.</td>
</tr>
</tbody>
</table>

*Table 5: Interesting “benign” disturbance types*

<table>
<thead>
<tr>
<th>Type of malign disturbance</th>
<th>Discovered characteristics &amp; comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movement</td>
<td>Dead reckoning (integration) not feasible. See Chapter 3.4.3. In practice however, generally gives rise to the feature of <em>unipolar persistent acceleration</em> in this actual context. See Chapter 5.2.</td>
</tr>
<tr>
<td>Dismounting and removal from surface</td>
<td>Significantly and quickly changing tilt (gravity projection) vectors; see also Movement and Handholding. Solved by detecting so called <em>unipolar persistent acceleration</em>. See Chapter 5.2.</td>
</tr>
<tr>
<td>Handholding</td>
<td>Acceleration characteristics of tremors coupling into axes. Possibly known frequency band, see Chapter 3.4.5. Detection made possible by <em>unipolar persistent acceleration</em>.</td>
</tr>
<tr>
<td>Drilling (4mm drill according to ISEN spec)</td>
<td>High-amplitude repeating patterns, see Chapter 4.4.4. Detection made possible by exactly that characteristic, magnitude persistence (see Chapter 5.3).</td>
</tr>
<tr>
<td>Sawing, Filing</td>
<td>See Drilling</td>
</tr>
<tr>
<td>Blatant abuse(hammering, baseball bat, kicking, etc.)</td>
<td>Would make the accelerometer outputs clip if available bandwidth was higher. Not solved. However, detection of this is optional in all but sirens of the absolutely highest security classification.</td>
</tr>
</tbody>
</table>

*Table 6: Interesting “malign” disturbance types*
3.4 Prestudial reasoning

To better understand the acceleration patterns which will be observed in the field tests, it is necessary to have a sort of “theoretical framework” to better know what is expected to be observed and not elaborate aimlessly. Therefore, in this section, after looking at the expected conditions at rest, the disturbance types presented in Table 5 and Table 6 are discussed theoretically.

3.4.1 How disturbances affect the accelerometer (System model)

There is quite a convoluted path from the disturbances affecting the surroundings to the number series that are actually seen on the output from the accelerometer. It is of interest to have at least a general idea about the generic properties of this signal path. Just using common sense and knowledge about systems, we here propose a model for a single channel, the signal path can be seen in Figure 6.

**System M - mechanical coupling**

This is simply the mechanical coupling between the disturbance and the accelerometer. It could be highly varying and is not necessarily even approximately an LTI (linear time-invariant) system. An extreme example of system change would be if the system is initially mounted to a wall, which is disturbed by beating, and then is removed from it. M then becomes a system of zero amplification, i.e. no coupling between the disturbances and the output of M.

Because of the potential variability of this system it is very difficult to analyze or even draw general conclusions about it, but assuming the prerequisites in Chapter 3.4.1, most importantly the stiff mounting to a hard surface, are true, at least some assumptions about characteristics could be argued valid.

We could for instance assume that the acceleration signal before sampling is generated by the function

$$\ddot{A}(t) = \alpha(t) \cdot \ddot{A}_M + \beta(t) \cdot \ddot{A}_R + \ddot{n}_e \quad (3.3)$$
where $\vec{A}_M$ stands for an acceleration process generated by system movement or gravitational vector deviation (tilting), $\vec{A}_R$ stands for an acceleration process generated by resonance effects which arrives through surface coupling, and $\vec{n}_e$ is the sampling noise process. $\alpha$ and $\beta$ are time-dependent coupling functions which would depend on the current disturbance situation, i.e. at no disturbances $\alpha=\beta=0$.

The "movement-generated acceleration" $\vec{A}_M$ should prove to be of much lower frequency than the "resonance-generated" $\vec{A}_R$ acceleration. A movement or tilting would be expected to last for at least tenths of seconds while the resonant frequencies of a hard surface probably is on the order of hundreds or thousands of Hertz considering the high disturbance propagation speeds in common stiff materials. This implies that the energy spectrum $R_\vec{A}(f)$ of a movement-generated signal should contain comparatively high energy at low frequencies compared to when there is no movement. So a high apparent content of $\vec{A}_M$ would be especially characteristic for malignant movement.

**System A - accelerometer effects**

These are internal accelerometer effects caused by the internal construction of the accelerometer. Many of these parameters such as expected bandwidth and linearity are given in the datasheet, but unfortunately not in detail, it is for instance uncertain what the internal signal conditioning in the accelerometer actually does with the measured signals. Of course most of these details are likely to be trade secret of Freescale.

**System N - approximately white sampling noise**

This is the system that can be analyzed with the greatest certainty. Noise figures are given in the datasheet and the noise observed in trials appears normally distributed and consistent with these figures, as noted in Chapter 3.2.

The model, however, includes no parameter for mechanical noise, as this is actually real acceleration as experienced by the chip (but might not be actual acceleration of the chassis but just haphazardly propagated vibrational energy from the environment).

### 3.4.1.1 Sampling and oversampling

The sampling rate is set by a command to the accelerometer. From the datasheet it is unclear whether this does something internally to the accelerometer (in the A system) to restrict the mechanical bandwidth before sampling, or if the given bandwidth figure of ODR/2 just characterizes the Nyquist frequency. The trials done at a later stage (Chapter 4.4.5) were indicative of that no mechanical bandwidth restriction is being done. This means that there is a true risk for aliasing distorting the output data at low sample rates. However using maximum oversampling, which is the normal situation, the true samplerate is always 1600 Hz in the case of the MMA8451Q, so this means mechanical input frequencies at 800 Hz and below are less of a cause for concern. How oversampling is supposedly done mathematically, converting the raw sample sequence $x_S$ to the processed data $x_O$ and using $OSR=\Omega$ is shown in formula 3.4.

---

\[ x_O[n] = \frac{1}{\Omega} \sum_{k=n\Omega}^{n\Omega+\Omega-1} x_S[k] \] (3.4)

Instead of giving pure aliasing for frequencies over ODR/2, oversampling instead makes for a strongly decaying frequency response between ODR/2 and ODR/2 x OSR. That is, signals with a frequency greater than ODR/2 still give rise to aliasing, but the amplitude of the alias components is on average greatly reduced. See Figure 7 for an example of amplitude response when oversampling is used. Above ODR/2 x OSR, frequency response again increases up to a frequency of ODR x OSR according to a mirror image of this figure.

If the frequency content of the resonance-generated noise process \( \vec{A}_R \) is generally much higher than the selected ODR then this process ends up being dampened, which in this case could be beneficial because resonance-generated acceleration should per the already made reasoning in general be benign.

![Figure 7: Example of the alias-dampening effects of oversampling. Simulated oversampling amplitude plot in MATLAB from incoming sinewave. Note that exact multiples of the ODR frequency have an amplitude response of zero - this is due to the zero mean property of a sine wave signal.](image)

**Conclusions about system**
Because of the complexity of the system (especially the "M" system) and the limited time and resources available, it was decided that most of the decisions on how the detection would work had to be based on quite crude assumptions and observations in an environment fulfilling the prerequisites, and by some reasoning come to why what is being observed is actually observed.

Some general reasoning about the malign and benign conditions of interest are however made hereafter.

### 3.4.2 Conditions at rest

If the accelerometer is mounted in an arbitrary orientation on Earth, and experiences no mechanical disturbances, the only signals it will experience is the gravity \( g \), projected in a specific manner onto the
three axes called \([a_x, a_y, a_z]\) depending on orientation, and also the sampling noise
\[n_e \in N(0, \sigma_{\text{raw}}/\sqrt{\text{OSR}})\] (approximately) where \(\sigma_{\text{raw}} \approx 2.6\,\text{mg}\) according to trials in Chapter 3.2, and \(\text{OSR}\) is a configurable oversampling ratio parameter in the accelerometer. Considering the intrinsic sampling noise negligible,

\[
\frac{\Delta a_x}{\text{ODR}} \approx 0; \quad \frac{\Delta a_y}{\text{ODR}} \approx 0; \quad \frac{\Delta a_z}{\text{ODR}} \approx 0
\]

(3.5)

is the most apparent condition which is true at rest. No disturbance or energy input gives rise to no acceleration change. Approximately this condition was used in the calibration method decided for in Chapter 3.4.2.2.

### 3.4.2.1 The gravity, and tilt parameters

The observed gravity should be of the same amplitude no matter in what direction the accelerometer is mounted. The experienced acceleration on one axis is essentially the dot product between the true acceleration in a Cartesian coordinate system and the directions of the measurement axis (all always of length 1 and perpendicular to each other), so the conditions

\[
a_x = \hat{e}_x \cdot \hat{g}; \quad a_y = \hat{e}_y \cdot \hat{g}; \quad a_z = \hat{e}_z \cdot \hat{g}
\]

(3.6)

and

\[
\|\vec{a}\| = \sqrt{a_x^2 + a_y^2 + a_z^2} \approx \|\vec{g}\|
\]

(3.7)

should also always be true at rest. \(\|\vec{a}\|\) is the the length of the Euclidean vector \(\vec{a}\) as follows from the Pythagorean theorem. \(\hat{e}_x\), \(\hat{e}_y\) and \(\hat{e}_z\) are the directions of the measurement axes, as previously explained at the beginning of Chapter 3.

### 3.4.2.2 Gravity cancellation

The gravity must be calibrated out before the system can set itself in standby and wait for events to occur. This is because the large acceleration caused by gravity would otherwise characterize malignancy.

There are several possible methods for doing this. One is using a first-order IIR high pass filter with a very long time constant, on the order of a minute - however, possibly considerably shortened while the system is starting up so the user does not have to wait for the system to get ready. This does however require three floating point multiplications and three additions per sample \(26\), which in this weak

---

26 Smith, 2011; chapter 19.
computer system is not negligible, and would require a CPU load of 6% if samples arrive at 50 Hz (see Chapter 2.5.2). A FIR filter would absolutely not be feasible due to the long convolution window required for a long time constant.

It has been proposed\(^{27}\) to use formula 3.7 as a stationarity detector, i.e. \( \| \vec{a} \| - \| \vec{g} \| < C \), where C is an arbitrary sensitivity constant. This could be used to find a "silent" sequence fit for calibration. However calculating the square root on this 8-bit microprocessor is unnecessarily intense, using several thousand clock cycles, and it is unclear whether the detector is fit for this purpose.

Another alternative is to use a simple heuristic: check for a sequence which hardly varies at all (which would indicate good mechanical stationarity), and use the average of this sequence as the offset. This must be done at startup, and it could also be done periodically, once every few minutes or so, to counteract effects such as thermal drift. Computationally this is not intense at all requiring only a comparison and an integer addition (essentially just 50 clock cycles) per sample. This is the solution that was opted for.

The test condition which is used does not calculate variance, but picks an initial vector of acceleration values and then checks so that no subsequent samples differ strongly from this one, i.e. the conditions

\[
|a_x[0] - a_x[k]| < C_t; \quad |a_y[0] - a_y[k]| < C_t; \quad |a_z[0] - a_z[k]| < C_t
\]

or, if an initial calibration has been done at startup and the system needs to recalibrate for some reason,

\[
|A_x[0] - A_x[k]| < C_t; \quad |A_y[0] - A_y[k]| < C_t; \quad |A_z[0] - A_z[k]| < C_t
\]

So, (3.8), or if applicable, (3.9) must all be true all the way during a calibration session; where 0 is the index of the initial vector of samples, and \( k \) the index of the subsequent sample vectors; it would then run from 1 until the length of the window; \( k=1...n \). \( C_t \) is the calibration threshold - no sample may differ more from the initial value than this. If one condition becomes false just once due to a disturbance we can immediately cancel the session of calibration and wait for the next scheduled calibration time.

If the calibration succeeds then the sample mean of the not gravity compensated sequence is used as the offset null.

\[
\vec{a}_{avg} = \frac{1}{n} \sum_{k=1}^{n} \vec{a}[k]
\]

and consequently

\(^{27}\) Skog, et al. 2010; section IV B.
\[ \vec{A} = \vec{a} - \vec{a}_{avg} \]  

When \( \vec{A} \approx \vec{0} \) (essentially \( A_x < C_t; A_y < C_t; A_z < C_t \)) we from here refer to this as the system being in "equilibrium".

Freescale has suggested a similar method to this one in an application note\(^{28}\).

### 3.4.2.3 Thermal drift

According to the datasheet of the accelerometer\(^{29}\), it experiences a thermal drift of max 0.15mg/°C. Assuming a temperature drift of maximum 70°C, which is actually possible in an extreme unheated subarctic environment\(^{30}\), the output could then drift by as much as 11 mg, this is not negligible and calls for that the system should recalibrate itself (essentially reperform the suggested gravity cancellation) from time to time for the sake of precision and to avoid mistriggering. This will also prevent mistriggering in case the siren is mounted onto a structure which expands and contracts by changes in temperature.

### 3.4.3 Malign movement - Displacement

The main object, as already explained in Chapter 1.2, is to make it hard to move the siren from its mounting without this being detected. In an ideal world where the accuracy, precision and sampling rate from the accelerometer all were perfect/infinite, this would not be hard because according to Newtonian physics,

\[
\vec{p}(t) = \int_0^t \int_0^t \vec{A}(t) \cdot g \, dt + \int_0^t \vec{v}_0 \, dt + \vec{p}_0 \tag{3.12}
\]

position equals the vectorwise double integral of acceleration plus possibly an initial speed \( v_0 \) and position \( p_0 \). In the standard case, the initial conditions would be zero upon start. \( p \) denotes position, \( v \) velocity and \( A \) acceleration. \( \vec{p}(t) \) should evaluate to approximately zero over time in the case of no movement.

This assumes the accelerometer plane of movement does not change (no change in the \( \vec{e} \) vectors) - which has to be assumed as the system has no angular detector available (see Chapter 3.1).

There is noise in the system which causes problems. If we assume white noise, which the sampling noise resembles as concluded in Chapter 3.2, then the discretized apparent acceleration on a single axis is

---

29 MMA8451Q datasheet; p.9.
\[ A_{\text{app}}[n] = A[n] + n_{e}[n] \]

\( n_{e} \) is then the noise part, and is approx. white gaussian: \( n_{e} \in N(0, \sigma_{e}) \)

If no interpolation, but rectangular integration\(^{31}\) is used (easiest possible), then the apparent position error which results from the sampling noise is

\[
E_{\text{app}}[n] = \sum_{k=1}^{n} \Delta t \cdot \sum_{l=1}^{k} n_{e}[l] \quad \Rightarrow \quad E_{\text{app}}[n] = (\Delta t)^2 \cdot \sum_{k=1}^{n} \sum_{l=1}^{k} n_{e}[l]
\]

(3.13)

Note that \( \Delta t = ODR^{-1} \) in formula set 3.13 is the time slice over which velocities and accelerations are assumed to be static. In the right hand formula it has been factored out. What we are then left with as a result of the double summation is a sum of increasingly long cumulative sums of the noise realizations, this sum can be reordered and so, using multiplication, equation 3.13 can be rewritten as

\[
\frac{E_{\text{app}}[n]}{(\Delta t)^2} = 1 \cdot n_{e}[n] + 2 \cdot n_{e}[n-1] + 3 \cdot n_{e}[n-2] + \ldots + n \cdot n_{e}[1]
\]

(3.14)

Every term in the right hand side of equation 3.14 can then be seen as a realization of an independent normally distributed random variable, having a standard deviation which is down-scaled linearly depending on sequence position. \( n_{e}[1] \), for instance, is seen \( n \) times in the sum of cumulative sums, making its expected standard deviation \( n \cdot \sigma_{e} \), and the random variable realization with the highest index, \( n_{e}[n] \) is only seen once, making its expected standard deviation \( \sigma_{e} \).

If a number of normally distributed random variables are independent, then the sum of the variables is expected to have a variance which is the sum of the normally distributed variables’ individual variances.

Since the variance is the standard deviation squared, and we have a sum of standard distributions; each distribution having a standard deviation that decreases depending on \( n \), the standard deviation of position error can be written as a rather simple formula,

\[
\sigma_{E_{\text{app}}}[n] = (\Delta t)^2 \sqrt{\sum_{k=1}^{n} (k \cdot \sigma_{e})^2} = (\Delta t)^2 \sqrt{\frac{\sigma_{e}^2}{6} (2n^3 + 3n^2 + n)}
\]

(3.15)

With the expected parameters\(^{32}\) (\( \Delta t = 0.02 \text{ s} \); \( \sigma_{e} = 0.45 \cdot 10^{-3} \cdot 9.8 \text{ m/s}^2 \)) this leads to a normally distributed position error with a mean of zero and standard deviation of 16.7 cm after a minute (\( n = ODR \cdot 60 \)) from the sampling noise alone. It can be noted that this number is very optimistic\(^{33}\) during a session of movement other kinds of noise such as mechanical friction from the surface is likely

\(^{31}\) Hart, 2005.
\(^{32}\) Note that apparent accelerations, normally counted in g’s, must for purposes of distance calculated in meters be multiplied by 9.8 to get the proper unit m/s\(^2\).
to couple into the system. Not even this may necessarily integrate to approximately zero, as aliasing could occur. Also, even small values of constant bias does give rise to enormous position error (increasing quadratically due to the double sum of said bias).

It is therefore clear detecting movement or dislodgement calls for using assumptive heuristics or statistical analysis, not refined double integration as it is hard to do anything against white noise except averaging, and even more infeasible to do anything against noise we do not even know the general characteristics of. If we however assume certain underlying statistical properties on how the acceleration data has been generated it is possible to use a method which makes use of such statistics to dampen noise once the signal has deviated significantly from the equilibrium, such as the Alpha-Beta filter. De-facto Kalman filtering, something of a standard tool, is likely to be a dead end as it only really excels when a precise model of how the system should react to stimuli is known34, and especially so if several different sensors are available, none of which is true in this case. Moreover it is quite computationally intensive.

Increasing the oversampling ratio decreases both $\sigma_e$ and $k$, hence giving a dramatic decrease of position error as long as the system is stationary. However this also reduces frequency response for true disturbances as explained in Chapter 3.4.1 so doing this beyond a certain point would make the system useless. The final choice of output frequency for a system like this should be a trade off between three parameters: power consumption, frequency/transient response and noise reduction.

Increasing the oversampling ratio decreases both $\sigma_e$ and $k$, hence giving a dramatic decrease of position error as long as the system is stationary. However this also reduces frequency response for true disturbances as explained in Chapter 3.4.1 so doing this beyond a certain point would make the system useless. The final choice of output frequency for a system like this should be a trade off between three parameters: power consumption, frequency/transient response and noise reduction.

![Figure 8: Expected acceleration pattern from system movement along an axis](image)

Figure 8 shows the acceleration pattern which is expected from simple movement along an axis, that is, starting and then bringing the system to a stop. How well a movement follows this pattern and whether any properties of this kind in the real movement patterns can be utilized for detection is investigated in Chapter 4.4.1.

34 Gustafsson, p. 267.
3.4.4 Malign movement - Dislodgement from surface (Tilt)

As explained in Chapter 3.1, the projection of the gravity onto the three axes should stay fairly constant. If it is not, that is, if low-frequency components start appearing and by some measure are persistent, then this should count as malign. Here we show how tilting should in theory affect the projection of the gravity vectors.

If the parameters \textit{pitch} and \textit{roll} (common terms in aerospace applications) are defined as the angular deviation from the y and x-axis (given those are perfectly perpendicular to the gravity), respectively:

\begin{equation}
\text{pitch} = \arcsin\left(\frac{a_x}{\|\vec{a}_{avg}\|}\right) \quad \text{and} \quad \text{roll} = \arcsin\left(\frac{a_y}{\|\vec{a}_{avg}\|}\right)
\end{equation}

where \(\|\vec{a}_{avg}\|\) is the length of the gravity cancellation vector, which as explained in Chapter 3.4.2.2 could be calculated and updated when no apparent disturbances are seen, but it should in theory always be very close to \(\|\vec{g}\|\).

The pitch and roll could also be updated at the same time. Angular differences from the "undisturbed" pitch and roll could then be calculated straight away, and the sensitivity for this type of disturbance set with an angular parameter, if so turns out to be necessary.

3.4.5 Malign movement - Handholding

If the system is held in a person’s hands, then the hand tremors of a person should start coupling into it. All people have some degree of tremor in the hands, and the accelerometer should be able to detect this.
According to one source\textsuperscript{35} the frequency of a hand tremor is generally along the order of 4-9 Hertz. The detection method for this could be some sort of filter or frequency detector.

### 3.4.6 Malign movement - Sawing and drilling

Drilling the chassis is a fairly aggressive operation hard to theoretically analyze. From brief inspection (testing) it could be concluded that it gives rise to high-amplitude, high frequency components that are generally persistent and couple into all the axes.

The same goes for sawing, which also yields strong HF components. Sawing is however not included in the NSAI requirements but was tested just out of interest.

Possibly a moving variance (RMS) detector could be used to detect persistent nonstationarity, as suggested in a research paper\textsuperscript{36}; this should react to both handholding, sawing and drilling. However the question is if this will cause problems when it comes to the fact that the system should ignore benign similar phenomena such as drilling in a nearby wall. Hopefully those events will tend to cause a much lower amplitude, but it is left to Chapter 4 to see if this is the case.

### 3.4.7 Malign movement - Blatant abuse

This is essentially the easiest type of mistreatment to classify as malign (things such as beating the system with a baseball bat or crowbar), it should simply make the signal clip at the max/min values due to the high momentaneous acceleration, probably on several channels due to the haphazard mechanical spread of such strong disturbances throughout the chassis. Sensitivity could be set by deciding how many samples within a certain timeframe should be allowed to clip without triggering the detection.

### 3.4.8 Benign movement - Beating mounting surface

Disturbances that are coupled into the system from the wall could be very commonplace, in the case of construction work going on, nearby machines running, etc. The system should classify this as benign, as long as the amplitude and persistence are both restricted.

It is difficult to make a hard and rigid surface transfer low-frequency vibrations of significant amplitude. This is because very strong forces are required to create any persistent displacement or deformation of such a surface, and the disturbance energy from any small displacements made propagates quickly through hard materials\textsuperscript{37}, generally on the order of several thousand meters per second for metals and wood, which means the resonance frequency of such hard structures is prone to be high.

\textsuperscript{35} Stiles, 1976.
\textsuperscript{37} RF Café, 2011.
3.4.9 Benign movement - Slight earthquakes

That the siren alerts during a strong earthquake would probably be seen as perfectly acceptable. However, low-grade seismic activity occurs in many places on Earth, and it cannot be recommendable that the siren erroneously alerts at a weak brief earthquake.

The problem is that, naturally, it is hard to simulate real-world earthquakes. However an attempt to reduce the sensitivity to this type of movement could be done by studying seismograms or similar earthquake related data. Considering earthquakes generally move entire structures, low frequency movement is likely to actually reach the accelerometer in this case. Even if not implemented in the system due to the lack of testability, it should at least be considered when developing the detectors.

3.4.10 Benign movement - Hi Fi systems and on site machinery

A nearby washing machine or stereo system are (within reason) not malign events and should not be classified as such as they cause no harm to the siren.

While the sources of such disturbances can vary strongly in frequency, the fact that the system shall be mounted to a hard wall to fulfill the prerequisites is a factor which should in theory isolate out persistent low frequencies and make the system swing back and forth around its equilibrium point. This should hence have some sort of connection to the case of disturbing the mounting surface (Chapter 3.4.8).
4 Trials

4.1 General

The trials, that is, recording how patterns of malign and benign movement turn out in practice, were carried out in accordance with the prerequisites listed in Chapter 3.3.3. In the following chapter, those are described thoroughly: what patterns were observed in specific cases. Throughout the analysis, we notice what patterns are observed, and try to come to explanations as to why they are observed.

The interesting cases are treated with highest importance first, alternatingly benign and malign as to take the most important signal features into consideration first.

The goal with this chapter is to come to a conclusion of what really makes malign movement differ from benign in practice, so a mathematical framework then can be developed to make the distinction between those automatically.

What we are searching for is really easily calculable test statistics to determine if the siren has become displaced or not. It could be one single statistic or several in conjunction used. An optimal detector to decide for a system’s mechanical stationarity has been proposed in the literature\textsuperscript{38}; other than exploring if this option is viable we look for usable parameters to derive our own detectors.

4.2 The setup

A chassis for the smaller model of an Inferno siren (model name Micro) was used for the trials. The "breadboard" prototype accelerometer board was cut and filed to have a very smooth fit in the rails that normally house the factory made PCB. Some sticky goo is put between the 9-volt battery and card to prevent a loose fit. An example of a trial mounting using strong magnets on a refrigerator door is shown in Figure 10.

Data is sent to a PC over the serial (RS232) line at a 115200 baud speed. Before being transmitted, the data is conveniently formatted into Comma Separated Values (CSV) format by the sprintf ( ) C library function\textsuperscript{39}.


\textsuperscript{39} CSV in this case means the data output is formatted in rows of \((A_x \cdot 16384), (A_y \cdot 16384), (A_z \cdot 16384)\). The acceleration range of -2 to +2 g is hence fills out the data range of the C signed short integer datatype (-32767 to 32768).
4.3 Initial observations

The output frequency from the accelerometer in the trials was chosen as 50 Hz. The reason is that this frequency is seemingly high enough to capture transients in the case of movement, and interesting patterns in the data while not giving rise to an insurmountable amount of samples to process. The next step down, 12.5 Hz, is far too slow, and 100 Hz would consume too much power continuously, as already concluded in Chapter 2.5.1.3.

What could be noted in initial trials is that any indication of direction when the siren is stationary is no good measure for malignancy. The hopes were that disturbances arising from beating or otherwise disturbing the mounting surface would show up mostly in the direction of the surface normal (e,-axis), considering the nature of the mechanical coupling towards the surface. This turned out not to be the case; resonance effects and scattering in the siren chassis could be a possible explanation; a research paper moreover suggests that impact-like disturbances applied to an accelerometer are prone to affect all axes and that this phenomena may be “difficult if not impossible to model.”

There also seems to be a tendency for benign signals to do a lot of zero crossings around equilibrium and be transient.

---

40 Aslan, Saranli (2008); section 3.3.
4.4 Specific cases

4.4.1 Malign movement - Displacement

A very interesting type of malign disturbance, movement in an ideal environment (smooth feet on chassis on a tabletop), actually gives rise to an acceleration pattern which is quite similar to the one theorized in Chapter 3.4, as is shown below. While this situation does not follow the set prerequisites, checking for possibilities to do inertial navigation is central to any application where movement or dislodgement detection is involved. It is especially of interest to see how the friction noise from a surface disturbs the signal in reality.

![Figure 11](image1.png)

*Figure 11: Siren moved back and forth slightly trying to keep the direction of the a\_y-axis. Movement is done in quite a fast manner (~2cm/s).*

![Figure 12](image2.png)

*Figure 12: Siren moved back and forth slightly trying to keep the direction of the a\_y-axis. Movement is done extremely slowly and as carefully as possible. Artificial biases applied to data, see legend.*

At very low speeds, the signals hardly keep any unipolarity or a recognizable start-up/slow-down pattern at all. However as far as movement detection is concerned this case would not be realistic. In an actual tamper detection application, this kind of "miniscule and smooth" movement is impossible due to the fact that the siren is mounted by magnets or screws, as already explained in Chapter 3.3.3. The stiffness of the mounting simply prevents it and considerably much more violence is required to move the system in reality.
At reasonable speeds, an interesting property seems to be that the malignant movement signals retain the characteristic of several samples being quite far from equilibrium without returning to near equilibrium or doing a zero-crossing in a short time frame. This is also true, naturally, in the case of significant tilting, as concluded in Chapter 4.4.3.

With those experiences in mind we move on to the next type of disturbance which is the most common benign one.

### 4.4.2 Benign movement - Mounting surface disturbances

This is a case of so called structure-borne disturbances. Examples of those are: someone fastening nails into a wall not too far away; a loud party or event going on in a nearby apartment or doors opening and closing heavily. This section aims to make clear what kind of acceleration patterns some of those disturbances cause in reality.

Getting to the point, some sampling was done with the system mounted to a refrigerator using magnets. The door is beat once weakly, then strongly, and then it was beat moderately hard continuously.

![System mounted to door of refrigerator. Artificial biases applied to X and Z data, see legend.](image)

As can be noted in the legend, artificial biases of 0.5 and -0.5 g have been applied to Figure 13 to separate the different channels and make more details visible.

Additionally, what can be noted here is that this is an example of a benign signal, and it tends to swing back and forth around the equilibrium point. Zero crossings and a tendency to return to around zero is commonplace. The amplitude is at most approximately 0.4 g peak to peak.

Moving the system to a gypsum wall and mounting it with screws gives rise to similar patterns but of a lower amplitude (0.2g); on a concrete wall generally even less amplitude (hard to get a peak to peak swing of more than 0.10 g).
These trials gave even more support to the hypothesis given in the previous chapter: Benign disturbance types show a lack of persistent unipolar deviation from the equilibrium point and tend to be quite transient in nature.

4.4.3 **Malign movement - Removal from surface**

Removing the system from its place of mounting is highly likely to make the projection of gravity start to differ, and if the perpetrator’s hands are used to do so, some degree of hand tremor should also start coupling into the system. In this chapter we show an example of this.

![Lifting siren from table, carefully putting it back](image)

*Figure 14: System initially lying loosely on table. System then carefully lifted from table and carefully put back. Artificial biases applied to X and Z data, see legend.*

What is seen here, unsurprisingly, is at least the X axis significantly drifting about 30 mg from equilibrium in a couple of seconds, together with some hand tremors on the Y and Z axes. It is likely the principle of detecting “persistent unipolar deviation” on one or more axes is usable even here. Even if the perpetrator manages to keep the siren completely level, there should be a good chance of the hand tremors at least at some point having the property of persistent unipolar deviation.

4.4.4 **Malign movement - Chassis piercing**

4.4.4.1 **Drilling**

For the first trial here, drilling, the siren was fastened onto a gypsum wall and the front drilled with a 4mm metal drill until through. The pattern is shown in Figure 15.
This malignant disturbance does not generally follow the principle of “persistent unipolar deviation”. However it discerns itself in two other ways: one is the very high amplitude of acceleration, on the order of 0.5 g and above, not seen in any other case yet. The other one is the persistent nature of the pattern. While the persistence will probably prove to be true for some benign disturbances as well, such as disturbance from machines or HiFi systems, the persistence in conjunction with the very high amplitudes of acceleration could serve as an indicator for malignancy. Drilling other places on the chassis seems to give rise to similar patterns however with a slightly different amplitude distribution over the channels.

4.4.4.2 Sawing

Attacking the system with a saw is most definitely to be classified as malign. Even if not included by NSAI as a malign disturbance, it is another case of chassis piercing which is interesting. To investigate how this situation looks in reality the system was mounted to a gypsum wall with screws and then sawed on the side a few times back and forth with a moderate intensity using a metal saw with fine teeth.

![Drilling chassis with standard 4mm drill, stopping halfway before continuing](image1)

*Figure 15: Front of chassis drilled until through. Note the artificial biases in the legend. The drill is stopped at approximately sample 1320 and then re-started. Drill goes through at about sample 1470.*

![Sawing chassis using fine metal saw](image2)

*Figure 16: Sawing the chassis. Note the artificial biases.*

- 35 -
Noticeable are a high peak-to-peak values of more than 0.2g at least momentarily in the pattern. Subsequent trials with different but similar saws show a similar pattern but a seemingly slightly different frequency distribution (this is indeed highly dependent on the size of the sawteeth and speed of sawing).

Generalizing, the detection methods appropriate for this would probably be in line with drilling. While it does not keep the same style of persistence, there is still a magnitude of energy in the signal which will hopefully not, in general, be seen in cases of benign vibrations.

Conclusion: Drilling typically gives rise to an amplitude-persistent acceleration pattern of a generally high amplitude. Sawing also gives rise to an amplitude-rich pattern, but it is not as persistent.

### 4.4.5 Malign movement - Blatant abuse (Striking system with hammer)

To see the acceleration characteristics of the system being beat by a hammer, it was fastened to a gypsum wall and then beat semi-hard with a small hammer. Due to the risk of dislodging solder joints or otherwise destroying the laboratory board we didn't want to be too rough.

The signal was expected to clip even in this case but it does not. In fact it is not even close to clipping, with amplitudes of approximately 0.5 g being seen (clipping would be 2g from equilibrium for X and Y).

This is most likely a bandwidth problem. There is not much reason to doubt that this violent disturbance actually gives rise to very large accelerations\(^\text{41}\), but in this case it is so momentaneous that the event is probably simply averaged out to quite insignificant values. To verify that this is the case the averaging was set to OSR = 2, lowest possible, momentarily and the test re-run:

\(^{41}\) Endevco, 2010.
and in this case, clipping is seen. This answers a question that was initially asked in Chapter 3.4.1.1; with so large amplitude differences depending on oversampling ratio it must be deemed likely the bandwidth figures in the datasheet are just based on the Nyquist frequency, and not restricted internally.

Unfortunately there is no real way to detect this type of transients with the internal OSR set to a high value. Requesting hundreds of sample sets a second and doing the oversampling filter in software is out of the question due to performance and power limitations.

One possible solution that was considered, even if it would violate the "equipment neutrality policy" stated in Chapter 2.2, was using the accelerometer's internal transient interrupt logic to detect clipping. Unfortunately, from the datasheet, it seems like this logic is triggered by the data as seen after oversampling, not before.

Conclusion: Blatant abuse such as beating the system with a hammer is prone to cause clipping to the +/- 2 g levels. Available bandwidth due to averaging in this case is however a problem when it comes to detecting it.

### 4.4.6 Benign movement - On site machinery / Hi-Fi systems

This should have strong similarities to the case of mounting surface disturbances, as it is a type of disturbance reaching the accelerometer via the mounting surface. However the type of stimuli is quite different in nature: it might have a wildly differing frequency content depending on machine (or music). If it in the general case turns out that amplitude is restricted and that it keeps the property of generally no unipolar stationarity then probably no new method of ignoring it will have to be considered. With that in mind, we ran a test where the system was fastened to a wall in a room where a heavy duty (~2kW RMS) hi-fi system was present. The system was placed merely a meter from a huge speaker and aggressive techno music heavy on bass was turned on at a very loud and disturbing volume, making the room shake.

Even though this thesis creator found the bass line very extreme in its intensity, it was found out that the actual amplitudes as seen by the accelerometer are fairly low.
As can be seen in this graph the amplitudes are not even higher than 0.05 g, and it follows the property of unipolar non-persistence, with never more than three consecutive samples deviating more than 10 mg on one side of equilibrium during the entire five-minute test. It was comforting that even an extreme test like this didn’t give rise to patterns of a higher amplitude. Another test with the same music in a different, larger room but the system mounted with magnets to the power amplifier chassis yield similar patterns but of an even lower amplitude. The low amplitude is likely because of the disturbance frequencies being far above the ODR, so most of the incoming signal is averaged out.

Conclusion: Powerful hi-fi systems and machines give rise to patterns which are not unipolarly persistent in characteristic. The amount of energy in the observed signals per time generally appears low compared to malign but characteristically similar phenomena such as drilling.

4.4.7 Benign movement - Weak earthquakes

Unfortunately it is difficult to make earthquakes come by mail order, especially so in the seismologically stable region of Stockholm. Therefore it is futile to try to come to any result from observation in this regard. In certain publications, however, it is noted that the frequency of earthquakes in any direction as detected by seismographs rarely goes over 2 Hertz, and the amplitude less than 0.4 g even in areas where the earthquakes were strong enough to cause severe building damage. Apparently as little as 20 mg of peak ground acceleration characterizes an earthquake strong enough to make people unable to walk and lose balance, according to the National Institute of Building Sciences. Another source also confirms values of the same magnitude. If this is true then no effort needs to be made in creating a detector which filters out earthquakes specifically, as all malign disturbances of interest are much stronger than this amplitude.

4.4.8 Conclusions about trials

The experiments suggest that disturbances that approach the accelerometer through the mounting

---

42 Teymur, 2000; pages 11 and 17.
43 Lorant, 2010.
44 Anbazhagan, 2009.
surface mostly arise from the resonance effects in the surface. Since the surfaces are generally hard and stiff the resonance frequencies are high, so we notice a strongly oscillating and transient characteristic in the output data.

Any persistent deviation from equilibrium over a small threshold seen on any axis is highly characteristic of malignant displacement. To detect drilling and sawing, some sort of detector which is based on persistence over amplitude thresholds seems to be reasonable.

Detecting strong and very brief transients is unfortunately not possible due to the low bandwidth of the oversampled data.
5 Derived detectors

5.1 General

With what we have observed, it was now time to start working out methods for detecting the malign movement while filtering out the benign. Industry-standard methods for pattern analysis include different transforms, most notably the FFT and Wavelet transforms. These are all quite easy to implement but are - in this context - mathematically heavy in terms of execution time and required memory. Therefore looking into simpler methods is required.

5.2 Unipolar Persistent Acceleration (UPA) detector

A strongly reoccurring property of benign signals, is the tendency to not keep far from one side of equilibrium for very long. Something very simple is therefore proposed - the UPA detector (Unipolar Persistent Acceleration), which in this application seems highly characteristic for malignant movement and displacement.

For a single channel (each channel is handled separately), it is mathematically defined like this:

Let the sample \( n \) from a gravity compensated single channel be denoted \( A_n \).

Let the threshold function

\[
T_R(x, \tau_u) = \begin{cases} 
0 & -\tau_u \leq x \leq \tau_u \\
-1 & x < -\tau_u \\
1 & x > \tau_u 
\end{cases}
\] (5.1)

where \( \tau_u \) is a configurable threshold parameter (in g's - or in practical implementation - accelerometer resolution counts).

The UPA value is then the value \( k \) in the longest observed sequence in a series starting with sample \( n \) which fulfills the conditions:

\[
T_R(A_n, \tau_u) \neq 0 \\
T_R(A_n, \tau_u) = T_R(A_{n+1}, \tau_u) = T_R(A_{n+2}, \tau_u) = \ldots = T_R(A_{n+k}, \tau_u) 
\] (5.2)

If at least one sequence encountered where \( k > \tau_p \), issue alert.
In most programming languages this is very easy to implement on the fly, essentially by just increasing a counter with one whenever \( T_R(A_{\text{previous}}) = T_R(A_{\text{current}}) \) \( \land \) \( T_R(A_{\text{current}}) \neq 0 \). If \( T_R(A_{\text{current}}) = 0 \) the counter resets to zero immediately and if the counter goes over a certain threshold, which can be called \( \tau_p \) (measured in number of samples) an alert is raised.

What this means qualitatively is that the sensitivity of the detector is set by two thresholds, both the acceleration threshold \( \tau_u \) and the persistence threshold \( \tau_p \) - (how many consecutive samples may be on one side of the threshold before an alert is raised). Each channel is handled separately, occurrence on only one channel is enough to trigger the detector.

It can be noted that unipolar persistent acceleration implies there must be a notable degree of low-frequency component in the signal. This gives some support to the theory in given in Chapter 3.4.1. The opposite, however, does not hold true: a signal lacking the UPA property may still contain low frequency components masked by high frequency components. This should not in practice matter much as the system disconnects from any "resonance generated acceleration" such as machinery-originating disturbances as soon as it is removed from the site of mounting, triggering the UPA detector anyway.

It should also be noted that UPA is, by its nature, also a detector of persistent tilting, a functionality wanted as per Chapter 3.4.4. Angular deviation from rest on an axis required for triggering is given by

\[
\arcsin(\tau_u)
\]

(5.3)

if \( \tau_u \) in formula 5.3 is given in \( \text{g}'s \). This formula assumes that the system initially was not tilted.

By this we have defined the UPA detector; sensitivity defined by two parameters - the disturbance threshold parameter \( \tau_u \) and the persistence parameter \( \tau_p \). How these could be calibrated for good functionality in the actual application is explained in chapters 5.5 and 7.3.

5.3 Magnitude-Persistence Detector (MPD)

Since not all malignant signals follow the property of unipolar persistent deviation, at least one other detector is required. The most notable one of those is drilling.

What the malignant disturbance types that do not follow this property do follow however, seems to be persistent high amplitude in the signals, i.e. there is simply a lot of energy in the signal. The classic qualitative measure of signal energy is the RMS value, which would in this case be calculated by a simple standard deviation of acceleration amount over a running window,

\[
RMS_{3D} = \left( \frac{1}{n} \sum_{k=1}^{n} \| \vec{A}_k \|^2 \right)^{\frac{1}{2}} = \left( \frac{1}{n} \sum_{k=1}^{n} (A_{xk}^2 + A_{yk}^2 + A_{zk}^2) \right)^{\frac{1}{2}}
\]

(5.4)
for a detection window of size $n$. As the signal is assumed to deviate from the in $\vec{A}$ already included
$\vec{a}_{\text{avg}}$, there is no mean included in this formula. $A_{xk}, A_{yk}, A_{zk}$ denotes the gravity-compensated
acceleration samples from channels $x, y$ and $z$ respectively, of index $k$ in the detection window. An alert
would be issued if $RMS_{3D} > \tau_{MPD}$, where $\tau_{MPD}$ is calibrated to some reasonable value.

However, using this formula straight off, the calculation time and also memory usage would for this
application be highly objectionable, as a buffer on at least the order of a second would be needed to keep
the window; at 50 hertz that would be 300 bytes for the buffer alone. Multiplication and subsequent
summing into a long (32-bit) data type is quite intensive for the chosen CPU to handle as well.

Proposed MPD detector

Therefore the possibility for making a simpler detector which qualitatively does something similar was
investigated. One such possibility is to make a detector which just sums up the number of samples which
have reached over a certain threshold in magnitude during the duration of a window. Let the function

$$T_S(x, \tau) = \begin{cases} 0 & |x| < \tau \\ 1 & |x| \geq \tau \end{cases}$$

The disturbance magnitude function is then given by

$$\gamma_{MPD} = \sum_{k=1}^{n} T_S(A_{xk}, \tau_m) + \sum_{k=1}^{n} T_S(A_{yk}, \tau_m) + \sum_{k=1}^{n} T_S(A_{zk}, \tau_m)$$

(5.6)

for a window of size $n$. $\tau_m$ in formula 5.6 denotes the acceleration amount threshold for the MPD
detector. Note that the channels are handled together because we, in similarity to the RMS$_{3D}$ detector
want to group all channels into one single quantity (a high magnitude on several channels then give
higher probability of detection).

An alert would be issued if $\gamma_{MPD} > \tau_{MPD}$.

To avoid keeping a sample window in memory, however, it would be preferable to have a recursive
solution. There are some change detectors such as the CUSUM detector, proposed in change detection
literature$^{45}$, which is one such. It essentially tries to detect change by summing up the incoming samples,
but in every addition cycle also subtracts a "drift factor". Unfortunately this leads to the property of very
quick divergence upon hefty transients reaching over the drift factor, which is undesirable in this case
and would mean wall knocks could set off the detector.

---

$^{45}$ Gustafsson, 2000; p.66.
However inspiration was taken from this. The option which was finally opted for is to use the proposed \( \gamma_{MPD} \) detector, but to sum up the detector result continuously (not over a saved window), and in a similar way to the CUSUM detector let the result “bleed out” when the warning condition ceases.

For every separate channel \( A_x, A_y, A_Z \):

Initialize with \( \gamma_1 = 0 \)

Then \( \gamma_n = \max(0, \gamma_{n-1} + T_S(A_{xn}, \tau_m) + T_S(A_{yn}, \tau_m) + T_S(A_{zn}, \tau_m) - \nu) \).

Issue alert if \( \gamma_n > \tau_{MPD} \cdot \nu \) is the drift/bleed factor.

The \( T_S \) function is given in formula 5.5 and \( \tau_{MPD} \) as well as \( \tau_m \) are configurable sensitivity parameters.

By this we have defined the MPD detector, which depends on three parameters: the acceleration magnitude threshold \( \tau_m \), the amount threshold \( \tau_{MPD} \) and the drift/bleed factor \( \nu \).

### 5.4 Large transients

It would have been preferable to have a detector for large but very short transients as well. Unfortunately, this is with the used hardware and output data rates not possible, as noted in Chapter 4.4.5, except for possibly reducing the oversampling ratio. If this is done then some trials must be re-run, most notably in the cases of benign disturbances reaching the accelerometer by resonance - to ensure the apparent disturbances from wall beating and hi-fi systems do not cause clipping or set off the MPD detector.

Technically such a detector (LTD, large transient detector) would simply react on if the amplitude on one or more of the channels reaches clipping level. Deciding whether it is feasible to reduce the OSR to deliberately cause aliasing and make this kind of detection possible is left as future work.

### 5.5 Tuning the detectors

While deriving these simple detectors turned out to be a quite loose process due to the many requirements that had to be taken into consideration, and there is no indication of that they are even close to optimal at all, they could at least be tuned optimally using the method suggested in this chapter. If the detection system is implemented in MATLAB or similar software, then one could thereafter record large amounts of relevant acceleration data to the PC: each event in separate files or separated by some other means.

Let \( \vec{A}_M^n \) be the \( n \)th recorded event of knowingly malign acceleration data.

Let \( \vec{A}_B^n \) be the \( n \)th recorded event of knowingly benign acceleration data.
Let \( \delta(\theta, \bar{D}) \) be the full detection system, with \( \theta \) being the array of detector parameters, \( \bar{D} \) the sequence of input data and the system output is according to

\[
\delta(\theta, \bar{D}) = \begin{bmatrix} 1 & \text{detector interpreted event as malign} \\ 0 & \text{detector interpreted event as benign} \end{bmatrix}
\] (5.8)

and

\[
R_{DR}(\theta) = \frac{1}{p} \sum_{k=1}^{p} \delta(\theta, \bar{A}M_k)
\] (5.9)

\[
R_{FW}(\theta) = \frac{1}{q} \sum_{k=1}^{q} \delta(\theta, \bar{A}B_k)
\] (5.10)

Formula 5.9 is then an estimate of the detection reliability factor using parameters \( \theta \), i.e. an estimate of the probability \( P(\text{malignancy detected} \mid \text{malign event}) \), if there are \( p \) malign recorded sequences in total.

Formula 5.10, respectively, is an estimate of the false warning ratio using parameters \( \theta \), i.e. an estimate of the probability \( P(\text{malignancy detected} \mid \text{benign event}) \), if there are \( q \) benign recorded sequences in total.

A straightforward measure on the reliability of the system is then a division between the wanted and unwanted ratio, and the problem is to find

\[
\max \left( \frac{1 + R_{DR}(\theta)}{1 + R_{FW}(\theta)} \right)
\] (5.11)

The reason for adding ones in the denominator and numerator in formula 5.11 is just to prevent division by zero from occurring on a computer in the case of zero false alarms. It does not invalidate the maximization problem.

An optimal vector \( \theta \) is difficult to find analytically, but Monte Carlo simulation could be one approach to this (i.e. just testing millions random combinations of parameters in the applicable mathematics software such as MATLAB). This method is called simple sampling. Stratified sampling (which could in a special case be interpreted as ignoring by common sense impossible or implausible ranges of parameters) could be done to speed up this process. \(^{46}\)

For this thesis project carrying a full calibration study out in practice is left as future work due to time

\(^{46}\) Amelin, 2004.
constraints. It would also be quite pointless to carry out on the prototype board as the M system as explained in Chapter 3.4.1 could change significantly when the accelerometer is mounted to a factory-made PCB. Limited studies using Monte-Carlo simulations for calibration are however presented in chapter 7.

It can also be noted that this method is not restricted to using the derived detectors in this thesis - any better ones that are invented in the future may be used in the calibration. It is essentially a method of off-line machine learning.

5.6 Detection logic

Normally, the detectors would run simultaneously and in parallel, raising an alert by OR logic. Of course, in special applications, it could be preferable to deactivate one detector or the detection system altogether. The default case is shown in Figure 20.

![Diagram of detection logic](image)

*Figure 20: Diagram of detection logic. See chapters 5.2, 5.3 for descriptions of the two detectors UPA and MPD, respectively.*
6 Implementation

6.1 Software structure

The program in the microcontroller is driven by interrupts (which means that hardware external or internal to the microcontroller can control the program flow).

6.1.1 Data flow

Figure 21 shows how data flows through the system.

![Diagram of data flows through the system.]

Figure 21: Diagram of data flows through the system.

6.1.2 Execution flow diagram

The accelerometer asserts an interrupt using a separate hardware interrupt line whenever a set of new data is ready to be gathered. This will happen almost precisely at the frequency of the set output data rate, as long as the processor can handle the data. If it cannot, this is indicative of a serious error in the system such as buffer overflows. In the actual implementation such behavior traps to an error.

Whenever such an interrupt is raised the processor wakes up, if in a sleeping state, and immediately...
branches to the high-priority interrupt vector, which starts an I2C sequence to get the data out of the accelerometer, i.e. it "opens the data flow" from the accelerometer.

After the sequence is done the data is formatted properly as signed 16-bit integers and put into the circular FIFO buffers (essentially the same data structure as a queue) belonging to each respective channel. The reason for using buffers is to make for flexibility in the system. With the simple algorithms that were finally opted for they are not strictly necessary, if on the other hand FFTs or similar methods that require capturing data windows are to be used, queues are necessary as processing a window cannot be done in the short time frame between two samples.

If the buffers are too large by an arbitrary measure, an interrupt flag is set which invokes the DSP routines at the next possibility.

The DSP routines have a lower interrupt priority than the data gathering over I2C, so they can be interrupted by the accelerometer when a new set of samples is available. This is important to not risk that data stays unhandled for any significant period of time.

The DSP routines empty the buffer until it is below a certain threshold, and while doing so it processes and checks the data for malignancy (of course, by running the detectors derived in Chapter 5). If the data is deemed malign a flag is set in the microprocessor, which may then be used by software to light a LED or activate a buzzer for demonstrative purposes.

---

**Figure 22**: Diagram of execution while running normally. The main program loop runs only when no interrupts are pending; and the low priority interrupts run only when no high priority interrupts are pending.
The benefits with an interrupt-driven execution pattern is that it stays open to implementing other kinds of functionality in the system. It also saves lots of power as the chip can be put to sleep when there is nothing to do. Both of these benefits are essential for the expected application.

6.1.3 Parameters in implementation used at demonstration

Output data rate from accelerometer: 50 Hz.
FIFO buffer size: 32 signed short (16-bit) objects per channel (64 bytes per channel).
DSP processing flag is set when buffer size contains 25 objects to prevent overflow.

Calibration attempt interval (see Chapter 3.4.2.2): 10000 samples (=200 seconds).
Calibration abortion threshold (see Chapter 3.4.2.2): $C_t = 6 \text{ mg}$ for all channels.
Calibration window length (see Chapter 3.4.2.2): 150 samples (=3 seconds).

Detector parameters: As decided from calibration (see Chapter 7.4)

Total program memory used: 13 kB of 64 kB.
Total data memory in use: 1023 of 3936 bytes.
7 Results

7.1 General

With seemingly reliable detectors developed, and the software implemented into the microcontroller, only two things now remained:

- measuring the current draw figures of the final implementation.
- checking whether the detectors actually perform well under the set prerequisites.

7.2 Current draw

Measuring the current with the multimeter in the same fashion as described in Chapter 2.5.1.1, but connected in series with the battery instead of the microcontroller, and averaging over ten minutes, the apparent current draw was measured to \(739 \, \mu\text{A}\). This value makes sense, and is indicative of the microcontroller spending approximately 4% of time in an awake state. Measuring the amount of time spent in an awake state by toggling an I/O pin confirms a value of the same order (~5%).

This current draw is well below the set goal of 1 mA. Implemented in the existing system would result in an idle total power consumption of approximately 1,25 mA in total. At this current draw, in theory half of an 1800 mAh battery’s capacity would remain after a month of standby.

How the power consumption could be reduced further is discussed in Chapter 8.

7.3 Detection functionality

The detection functionality was tuned and tested by making a limited study: recording 30 malign events and 30 benign events into separate files onto a PC. What those events were is noted in the appendix. Recorded malign events longer than a second were manually truncated at a second after the first apparent sign of disturbance because the detectors must react within a second after the disturbance begins.

The detectors, implemented in MATLAB, then interpret all events. The formula for calculating detection score is similar to formula 5.11, but in a diagram it is more relevant to have a score between 0 and 1. 0 then represents irrelevant/worthless detection, and 1 represents a perfect score where all malign events were detected and no benign events were misinterpreted as malign.
Hence, the detector score is calculated as Formula 7.1.

\[
\delta_{\text{SCORE}} = \left( \frac{1 + R_{DR}(\theta)}{1 + R_{FW}(\theta)} \right) - 1
\]  

(7.1)

The full detection system depends on five detector parameters, the vector \( \theta \), as explained in Chapter 5: the UPA parameters are \((\tau_u, \tau_p)\) and the MPD parameters are \((\tau_m, \tau_{MPD}, \nu)\).

7.3.1 Selecting appropriate MPD parameters

The UPA detector is by far most important to the system, as it detects the type of malignancy required by systems of the lowest security classification: dislodgement and movement. The MPD detector is only relevant in the case of chassis-piercing sabotage such as drilling. This is an additional feature which must be present in systems of a higher security classification.

An alternative is therefore to just fix the MPD detector to parameters which perfectly detect the few (13) cases of drilling which are included in the limited study, and give no false positives for benign data. A set of such parameters was found by using formula 7.1 in a Monte Carlo simulation. However, in this simulation, the detection system had no UPA detector active; also, \( R_{DR}(\theta) \) was calculated only using the cases of drilling as the input (while \( R_{FW}(\theta) \) uses all benign cases as the input).

![Figure 23: Parameter sets resulting in a perfect MPD detector score, seen as functions of pairs of the three different detector parameters, see axes labels.](image)

10,000 cases were simulated in MATLAB, for each case the MPD parameters were selected at random: \( \tau_m \) having an uniform distribution from 0 to 2000 accelerometer counts, \( \tau_{MPD} \) having a uniform distribution from 0 to 24, and \( \nu \) having a uniform distribution from 0 to 2.

554 of the 10,000 cases resulted in a perfect detection score. The parameters of those cases are shown in Figure 23. As can be seen, there is a wide range of possible parameters. The exact pros and cons of different positions in the feasible parameter range is not well understood, except for the obvious, namely...
that low values of parameters makes the system more sensitive at detecting chassis piercing, as well as becoming more prone to false alarms.

Interestingly it can also be noted from the figure that low $\tau_m$ values require a high $\nu$ for the detection to be reliable, and vice versa. This is expected: if the detector is sensitive to low amplitude thresholds then the samples reaching over the threshold will be much more commonplace than by using a high amplitude threshold, so a high drift factor is required to prevent false alarms.

In the end, parameters right in the middle of the feasible range were chosen, by calculating the vectorwise mean of all possible parameter sets. The parameters selected were:

\[
\begin{align*}
\tau_m &= 310 \text{ accelerometer counts (}=19 \text{ mg)} \\
\tau_{MPD} &= 16 \\
\nu &= 0.96
\end{align*}
\]

The sets of perfect detector scores around this point is rather dense compared to the outskirts of the feasible range. This is indicative of that it could be a rather good choice of parameters because the random input parameters to the simulation are uniformly distributed.

### 7.3.2 General detection reliability result

Using the parameters for the MPD detector decided for in Chapter 7.3.1, the detection score is then only dependent on the two UPA parameters, which makes a meaningful graphical representation of detector performance depending on those two important parameters possible, in this case a surface diagram was used where the brightness and elevation both denote the detection score. The result is given in Figure 24.

The number of parameter sets tried out were 100 in total, with $\tau_u$ spanning from 20 to 200 in steps of 20 and $\tau_p$ spanning from 1 to 10 in steps of 1.
The parameter sets that resulted in a perfect detection score (= 1) turned out to be two: \((\tau_u, \tau_p) = (60,7)\) and \((60,8)\).

7.4 Parameter selection from trial results

The final detector parameters decided for the implementation, used at the demonstration associated with this thesis project were:

\[ \tau_m = 310 \text{ accelerometer counts } (=19 \text{ mg}) \]
\[ \tau_{MPD} = 16 \text{ (dimensionless)} \]
\[ \nu = 0.96 \text{ (dimensionless)} \]
\[ \tau_u = 60 \text{ accelerometer counts } (=3,7 \text{ mg}) \]
\[ \tau_p = 7 \text{ samples } (=140 \text{ ms}) \]

Detectors used are: UPA (Chapter 5.2) and Recursive MPD (Formula set 5.7).

The parameters for the UPA detector are close to what was come to by intuition long before an actual parameter-dependent functionality trial was done. The peak positions in Figure 24 are therefore quite expected.
7.5 UPA/MPD event distinguishability

There are several ways to measure the reliability of this type of system. The reliability measure graphed in Chapter 7.3 is indicative of how well the system separates benign events from malign events, which in most cases could argued to be the most important measure of overall detection reliability.

However, another interesting measure could be how well the detectors can distinguish between supposedly “UPA-triggering events” (movement, dislodgement, lifting) and “MPD-triggering events” (chassis piercing i.e. drilling). This was also tested in MATLAB.

Using the parameters selected in Chapter 7.4, it turned out that especially the UPA detector distinguishes poorly between the two. More often than not it also detects drilling, with 10 out of the 13 cases of drilling triggering the detector. This was not expected, as the behavior was not observed using approximately the same UPA parameters during initial trials. However, detector behavior might strongly depend on the environment, type of drill, and the overall smoothness of drilling, so it is not surprising.

The MPD detector, however, only reacts on 1 out of 17 “UPA-triggering events” with the same parameters - it shows quite acceptable separation properties.

With the parameters selected, the following logic could indeed be used to distinguish between events:

- If only the UPA detector triggers, the event is most likely rather careful dislodgement, movement or lifting.
- If only the MPD detector triggers, the event is likely to be drilling.
- If both detectors trigger, the event is either drilling and/or very violent/significant dislodgement and movement.

However, using a simple Monte Carlo simulation, varying all detector parameters at random, it was also checked whether there exists parameters such that both “definite separation performance” and overall detector reliability stays reasonably high. The performance measures used were three: the overall \( \delta_{SCORE} \) as given in formula 7.1, and also, the measures explained hereafter.

First, we give the formula for the detection ratio of non-drilling events by the UPA detector (formula 7.2).

---

48 By definite event separation we mean that MPD reacts only on cases of drilling, and UPA reacts only on other malign cases (i.e. movement/dislodgement).
\[ R_{UD}(\theta) = \frac{1}{u} \sum_{k=1}^{u} \delta_{UPA}(\theta, \vec{A}U_k) \] (7.2)

\( \vec{A}U_k \) stands for malign acceleration sequences that were not generated by drilling. There are \( u \) such sequences in total. \( \delta_{UPA} \) means the detector used in this case is UPA alone.

Then, the UPA detection ratio of events generated by drilling in formula 7.3.

\[ R_{UCD}(\theta) = \frac{1}{s} \sum_{k=1}^{s} \delta_{UPA}(\theta, \vec{A}m_k) \] (7.3)

\( \vec{A}m_k \) stands for malign acceleration sequences that were generated by drilling. There are \( s \) such sequences in total.

Then the formula for the detection ratio of drilling events by the MPD detector (formula 7.4).

\[ R_{MD}(\theta) = \frac{1}{s} \sum_{k=1}^{s} \delta_{MPD}(\theta, \vec{A}m_k) \] (7.4)

\( \delta_{MPD} \) in formula 7.4 means the detector used in this case is MPD alone. Finally, we give the formula for the detection ratio of non-drilling events by the MPD detector (formula 7.5).

\[ R_{MCD}(\theta) = \frac{1}{u} \sum_{k=1}^{u} \delta_{MPD}(\theta, \vec{A}U_k) \] (7.5)

The test scores ("UPA separation" and "MPD separation") are then calculated by formulas 7.6 and 7.7, respectively. Note that the subscript "CD" stands for "Cross Detection" (which is an unwanted ratio).

\[ UPA_{sep} = \left( \frac{1 + R_{UD}(\theta)}{1 + R_{UCD}(\theta)} \right) - 1 \] (7.6)

\[ MPD_{sep} = \left( \frac{1 + R_{MD}(\theta)}{1 + R_{MCD}(\theta)} \right) - 1 \] (7.7)
Merely good separation scores is not enough, as this criterion alone allows for misdetection of benign events; it is also important that overall $\delta_{\text{SCORE}}$ stays high.

The criterion for "good parameters" stored from the Monte-Carlo simulation when encountered was that they result in $\delta_{\text{SCORE}} > 0.85$ (formula 7.1), $UPA_{\text{sep}} > 0.80$ and $MPD_{\text{sep}} > 0.80$.

After running 10000 trials using the same input parameter distribution as in Chapter 7.3.1, 28 of them fit the criteria. What the parameter sets have in common are low thresholds of $\tau_u$ (about 30 accelerometer counts), and high thresholds of $\tau_p$ (on the order of 20 samples). The parameters for the MPD detector are generally of the same order of magnitude as decided in Chapter 7.4. One parameter set did stand out as the best (if we still consider $\delta_{\text{SCORE}}$ to be of the highest importance). This parameter set was:

- $\tau_m = 230$ accelerometer counts (=14 mg)
- $\tau_{MPD} = 11.1$ (dimensionless)
- $\nu = 1.28$ (dimensionless)
- $\tau_u = 48$ accelerometer counts (= 2.8 mg)
- $\tau_p = 15$ samples (= 280 ms)

which resulted in: $\delta_{\text{SCORE}} = 0.97$, $UPA_{\text{sep}} = 0.83$ and $MPD_{\text{sep}} = 0.86$.

Focusing the Monte-Carlo simulation onto values close to these parameters, it could be noted that the parameters do have some freedom around this range - other parameter sets rather close to this one giving similarly good results can be found. With five possible degrees of parameter variability, investigating the effect of specific variables on detector separation performance is beyond the scope of this thesis.

What we conclude with here is that the MPD and UPA detectors were able to, at least to a certain degree, distinguish between the two styles of malign events even if performance is not very impressive. In case reliable malignancy detection is of higher importance, formula 7.1 alone should be used for calibration.
7.6 Alternative MPD detectors

Two alternatives to the recursive MPD detector normally used are mentioned in Chapter 5. One is windowed 3D RMS (formula 5.4) and the other is windowed MPD (formula 5.6). While these are not very suitable for use in the simple microcontroller, their performance is at least tested in this chapter.

7.6.1 Windowed 3D RMS and windowed 3D MEAN

Using formula 5.4 in place of the MPD detector, and the same method as in Chapter 7.3.1 for calculating MPD detector performance, a Monte-Carlo simulation was again run using the following distributions of input parameters:

\[ \tau_{MPD} \text{ (detector threshold) uniformly distributed from 1 to 100000.} \]
\[ k \text{ (window length) uniformly distributed from 1 to 50 samples (rounded to integers).} \]

After taking note of in what parameter ranges the score was significantly higher than 0, the Monte-Carlo simulation was rerun (still with \( k \) uniformly distributed \([1,50]\)) but now with \( \tau_{MPD} \) uniformly distributed \([1,10000]\). The detector score was then plotted over this range of parameters, as can be seen in Figure 25.

![Figure 25: Detection score for windowed 3D RMS detector plotted as a function of the two detector parameters](image)

The detector score has its peak at \( \tau_{MPD} = 1658 \) and \( k = 47 \), where the score is 0.61. This score is unacceptably low; misdetected at peak score were benign events (1,6,11), and undetected were malign events (27,28,29) - see the appendix.
Upon investigating why the detector shows so low performance, it became apparent that the squared scaling of the input magnitude is a problem. Brief large amplitudes are prone to set off the detector, while the least violent type of drilling (into the plastic sides of the siren) is prone to go undetected. Note that in the case of calculating de-facto RMS one single high-amplitude sample can easily cause a larger RMS value over a window than a chain of samples with moderately high amplitudes. This leads to that no choice of $\tau_{MPD}$ results in perfect detection in this case.

Out of interest was then to check if the result is better by dropping the “square” in RMS, i.e. using the formula 7.8 as a detector, it is essentially the 3-D amplitude mean over a window. The Monte-carlo simulation was run in the exact same fashion as with the RMS detector and the same parameter ranges. The result plot is given in Figure 26.

$$MEAN_{3D} = \left( \frac{1}{k} \right) \sum_{k=1}^{n} \sqrt{A_{xk}^2 + A_{yk}^2 + A_{zk}^2}$$

(7.8)

This result diagram shows a perfect score around $\tau_{MPD} = 671$ and $k = 47$. However, the range of parameters resulting in a perfect detection score is narrow, this can easily be seen in the diagram. In this application the 3D MEAN detector is therefore not interesting, either. The recursive MPD detector (Chapter 7.3.1) has proven much easier to calibrate.
7.6.2 Windowed MPD

The results from Chapter 7.6.1 are not impressive, the detectors are therefore not interesting to use. What the detectors tested had in common, in contrast to the recursive MPD detector, is linear or quadratic weighting of the incoming sample amplitudes. It is therefore worth checking whether a threshold weighting of the signal amplitudes is the key to good detection. The formula 7.9 was used for a “Windowed MPD” detector. The $T_S$ function in this formula is given by formula 5.5.

$$WIN_{MPD} = \left( \frac{1}{k} \right) \sum_{k=1}^{n} T_S(\sqrt{A_{yk}^2 + A_{zk}^2 + A_{rk}^2}, \tau_m)$$

(7.9)

$\tau_m$ in formula 7.9 was set to 310 accelerometer counts, just as in Chapter 7.4. Then another Monte-Carlo simulation of 10,000 samples was run, the results are presented in Figure 27. The range of $\tau_{MPD}$ input values was changed to reflect the low magnitude of output values (0 or 1) from the threshold function $T_S$.

This simulation shows a notable “plateau” of perfect detector scores. This is preferable as it means the detector can operate over a rather large range of parameters. To verify that also $\tau_m$ is variable, encountered perfect detector scores resulting from a Monte Carlo simulation were plotted as functions of the three detector parameters, similar to Chapter 7.3.1. This is presented in Figure 28.
Very short windows, especially ten samples and shorter, makes the detector reliability suffer heavily. This is expected as the detector then cannot distinguish between persistent and non-persistent high-amplitude disturbances; the short window means the detector forgets too much history.

Like the recursive MPD detector, this windowed MPD detector also shows quite a “cloud” over which the parameters are allowed to vary widely while still keeping detector score perfect. This windowed one could be argued to be slightly better; the recursive one shows a clear dependence between two of the parameters, so freedom of parameter variability is not as good (see Figure 23). In contrast, this cloud is a bit more uniform. There is a connection between amplitude threshold $\tau_m$ and the alert threshold $\tau_{MPD}$ which is expected, but it is not as apparent as the dependence between $\tau_m$ and $\nu$ for the recursive detector.

We therefore conclude that windowed MPD is probably a better detector than recursive MPD, at least if detection reliability is of the highest importance.

Instead using formula 5.6, the initially proposed windowed MPD detector (which handles the data from the different channels separately), shows similar results and parameter variability. Spending processor time to calculate $\|\vec{A}\|$ therefore seems to be unnecessary.
8 Conclusions and discussion

8.1 Conclusions

We have by this thesis demonstrated how to, supposedly reliably, use a single three-axis accelerometer as a dislodgement and sabotage detection solution in a stationary environment, and that it does not necessarily have to consume obtrusive amounts of power, either. There is the option of using no advanced mathematics in the detection equations, not even multiplication. Also, the memory requirements are very small. Even simpler and cheaper microcontrollers than the one used in this project might therefore be relevant to use.

The detectors can to a certain degree distinguish between the two significantly different types of malign disturbances. However, setting the detector parameters to values which makes this possible reduces detection reliability. For the purpose of purely distinguishing between malign and benign disturbances, the detection system did show highly satisfying performance in the limited trial study.

8.2 Discussion and future work

While the detection results are satisfying and the power consumption low enough for the given system, some ideas on improvements come to mind.

8.2.1 Changing used buffers

In this project, the accelerometer's proprietary internal FIFO buffers were not used to keep the system vendor neutral; and the processor instead offloads every sample as it is generated. However, in a commercial implementation one could resort to using them. This would mean the processor could be allowed to wake up only a few times a second to unload and process data, leading to huge power savings, perhaps more than 50%.

8.2.2 Changing microcontroller

The PIC microcontroller in this project apparently far from stands out when it comes to using little power for waking up and going to sleep. Freescale, for instance, offers a microcontroller series with a mere 6 µs wake-up time claimed\(^\text{49}\). If one does not want to resort to using the accelerometer’s FIFO buffers then this could be another approach to reduce power consumption. It would also presumably allow increasing the ODR without significantly increasing the power usage, possibly allowing for the quick transient detection which was wanted but not possible in this implementation.

\(^{49}\) Freescale Semiconductor brochure (2009).
9 Bibliography


Appendix - Functionality testing event list

Benign events

b1 Magnet refrigerator mount, hard elbowing ~1dm distance five times
b2 Magnet refrigerator mount, hard jumping onto floor
b3 Magnet refrigerator mount, slight jerk of refrigerator from side
b4 Magnet refrigerator mount, slight knocks onto door and chassis
b5 Magnet refrigerator mount, various actions (Hard floor kicking, fridge elbowing etc)
b6 Magnet refrigerator mount, elbow knocking ~2dm distance
b7 Screw mount onto wooden table, slight shaking
b8 Stereo magnet mount ~1m distance from speaker, techno music
b9 Stereo magnet mount ~1m distance from speaker, bass-heavy ethnic music
b10 Stereo magnet mount ~1m distance from speaker, bass-heavy jazz music
b11 Gypsum wall mount - knocking with rubber hammer ~1dm from mounting
b12 Gypsum wall mount - Elbow ~5dm from mounting three times
b13 Stereo magnet mount - 50Hz test tone ~1m from gypsum wall mounting
b14 Woodwall mount, Striking elbow 3x @ 2dm. distance
b15 Woodtable mount, slightly shaking table
b16 Stereo magnet mount #2 - ~1m distance from speaker; dance music
b17 Stereo magnet mount #2 - ~1m distance from speaker; trance music
b18 Stereo magnet mount #2 - ~1m distance from speaker; dance music
b19 Gypsum wall mount - knuckle knocking x 5
b20 Gypsum wall mount - knuckle knocking x 5
b21 Woodboard mount using clamp - hammer x 3 @ 70cm
b22 Woodboard mount using clamp - table elbow thumping @ 10cm
b23 Woodboard mount using clamp - strongly jerking board
b24 Woodboard mount using clamp - Beating fist moderately hard @ 10cm distance
b25 Woodboard mount using clamp - Moderate knuckle knocking (right on chassis) x 20
b26 Magnet radiator mount, stomping floor in immediate vicinity hard several times
b27 Magnet radiator mount, knuckle knicking siren (right on chassis) x 20
b28 Magnet radiator mount, playing music on compact stereo system nearby
b29 Woodboard mount, compact stereo speaker playing loud dance music at mere cm. distance
b30 Woodboard mount, slightly jerking woodboard three times
**Malign events**

m1 Loose on table, careful pitching
m2 Loose on table, careful lifting
m3 Loose on table, careful lifting
m4 Loose on table, careful indirect yawing (with screwdriver)
m5 Loose on table, very careful lifting
m6 Loose on table, careful yawing
m7 Gypsum wall mount, drilling corner w/4mm metal drill bit
m8 Gypsum wall mount, drilling front w/4mm metal drill bit
m9 Gypsum wall mount, drilling front w/5mm standard drill bit
m10 Magnet refrigerator mount; carefully opening door
m11 Magnet refrigerator mount; carefully opening door
m12 Magnet refrigerator mount; careful removal of siren
m13 Magnet refrigerator mount; careful removal of siren
m14 Loose on table, careful lifting
m15 Loose on table, careful lifting
m16 Loose on table, careful lifting
m17 Loose on table, careful lifting
m18 Loose on table, careful lifting
m19 Loose on table, careful lifting
m20 Loose on table, careful lifting
m21 Drilling plastic side of siren, 5mm standard drill
m22 Drilling plastic side of siren, 5mm standard drill
m23 Drilling plastic side of siren, 4mm metal drill
m24 Drilling front of siren, 4mm metal drill
m25 Drilling plastic side of siren, 5mm standard drill
m26 Drilling front of siren, 5mm standard drill
m27 Drilling plastic side of siren, 4mm metal drill
m28 Drilling plastic side of siren, 4mm metal drill
m29 Drilling plastic side of siren, 5mm standard drill
m30 Drilling front of siren, 5mm standard drill