



FACULTY OF SCIENCE AND TECHNOLOGY

## MASTER THESIS

Study programme/specialisation:	The spring semester, 2023
Master of Science / Robot Technology and Signal Processing	Open
Author: Anders Espedal	
Supervisor at UiS: Kristian Thorsen External supervisor: Fredrik Sævland	
Thesis title: Implementation of a acceleration estimator based compensation scheme to increase load data accuracy for a robotic testing system for CPR-manikins	
Credits (ECTS): 30	
Keywords: Product testing, calibration, accuracy, robotics, estimation, compensation	Number of pages: 95 + appendix/other: 126  Stavanger 15th June 2023

# Acknowledgement

First of all I would like to thank my supervisor Kristian Thorsen for his guidance and assistance from shaping the draft for this thesis and his aid through the process since that stage.

I would also like to extend my gratitude to Wenche Litlehei and Eivind Tønnessen for facilitating the cooperation between Laerdal Medical and the Department of Electrical Engineering and Computer Science (IDE) at the University of Stavanger. Also i would like to thank Eivind Tønnessen for his practical and technical guidance and supply of equipment documentation.

Also, my thanks goes to everyone i have encountered at Laerdal Medical for their support, guidance and general interest in this project.

I also want to thank my fellow students and the student environment at IDE in general. My peers are incredibly knowledgeable, helpful and nice people.

As the first student writing my master thesis as part of the development work at Laerdal Medical's Test Center, i hope that this work serves as the spark of a prosperous relationship between the Test Center and IDE.

# Abstract

Laerdal Medical is a producer of Cardiopulmonary Resuscitation (CPR) training manikins, all of which undergo rigorous endurance and accuracy testing. This work proposes an acceleration estimator based compensation scheme for a industrial robot manipulator product testing system with the intention of increasing load data accuracy for the purpose of product review and calibration. As part of the compensation scheme four different acceleration estimators are implemented and compared. Results indicate that the compensation scheme increases the load data accuracy by 1.5 - 6 % of the reference value depending on compression depth and spring rate. However the accuracy goal of 0.4 [kg] is not reached. The work has also uncovered the presence of position error in the robot. Thus, further improvement to the compensation scheme and positional error compensation is required.

**Note:** All source code for implementation, data-processing and raw data can be found in the thesis repository [https://github.com/espedalen/Robot\\_CM.git](https://github.com/espedalen/Robot_CM.git).

---

## Abbreviations

<b>CPR</b>	Cardiopulmonary Resuscitation
<b>TCP</b>	Tool center point
<b>ERC</b>	European resuscitation council
<b>cpm</b>	compressions per minute
<b>BFD</b>	Backwards finite difference
<b>SGF</b>	Savitzky-Golay filter
<b>KF</b>	Kalman filter
<b>EAW</b>	Enhanced adaptive windowing filter
<b>ABS</b>	Anti lock braking system
<b>EKF</b>	Extended Kalman filter
<b>LD</b>	Linear differentiator
<b>TD</b>	Tracking differentiator
<b>END</b>	Enhanced nonlinear differentiator
<b>FCFVE</b>	Fast curve fitting velocity estimator
<b>LSVE</b>	Least square velocity estimator
<b>NVE</b>	Nonlinear velocity estimator
<b>LDD</b>	Linear double differentiator
<b>LSCF</b>	Least square curve fitting
<b>FCFE</b>	Fast curve fitting estimator
<b>RMSE</b>	Root mean square error
<b>GUM</b>	Guide to the expression of uncertainty in measurement
<b>GUI</b>	Graphical user interface
<b>EGM</b>	Externally guided motion
<b>RDT</b>	Raw data transfer
<b>RWS</b>	Robot web services
<b>TCP (2)</b>	Transmission control protocol
<b>UDP</b>	User datagram protocol
<b>API</b>	Application programming interface
<b>REST</b>	Representational state transfer
<b>HTTP</b>	Hypertext transfer protocol
<b>Protobuf</b>	Google protocol buffers
<b>IP</b>	Internet protocol
<b>ECU</b>	Environmental compensation unit
<b>SNR</b>	Signal to noise ratio
<b>CI</b>	Confidence interval



# Contents

<b>Acknowledgement</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Abbreviations</b>	<b>iii</b>
<b>Content</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem description . . . . .	2
1.2 Compression testing . . . . .	3
1.3 Approach . . . . .	5
1.4 Scope . . . . .	6
1.5 Outline . . . . .	6
<b>2 Theory and Method</b>	<b>8</b>

## CONTENTS

---

2.1	Dynamic load compensation . . . . .	8
2.1.1	Related work . . . . .	11
2.2	Compensation scheme . . . . .	13
2.3	Acceleration estimation . . . . .	14
2.3.1	Numeric differentiation by Second Order Backwards Finite Difference . . . . .	17
2.3.2	Linear Tracking Differentiator . . . . .	17
2.3.3	Fast Curve Fitting Estimator . . . . .	18
2.3.4	Kalman Filter . . . . .	21
2.4	Performance evaluation . . . . .	24
2.4.1	Performance metrics . . . . .	25
2.4.2	Uncertainty of measurement . . . . .	26
2.5	Robot error sources . . . . .	30
<b>3</b>	<b>Description of hardware and software</b>	<b>32</b>
3.1	Robot cell and testing environment . . . . .	32
3.2	Robot Compression Testing Application . . . . .	33
3.2.1	Program structure . . . . .	34
3.3	Load Compensation Application . . . . .	37
3.3.1	EGM Implementation . . . . .	38
3.3.2	Net F/T Implementation . . . . .	38

## CONTENTS

---

3.4	ABB Resources . . . . .	40
3.4.1	RAPID . . . . .	40
3.4.2	Robot Web Services . . . . .	41
3.4.3	Externally Guided Motion . . . . .	42
3.5	Robot/Sensor-Computer communication protocols . . . . .	42
3.5.1	Google Protocol Buffers . . . . .	43
3.6	External equipment . . . . .	44
3.6.1	Load cell . . . . .	44
3.6.2	Interferometer . . . . .	44
3.6.3	Accelerometer . . . . .	45
3.6.4	Compression spring . . . . .	46
3.6.5	Scale AH46 . . . . .	47
<b>4</b>	<b>Simulation</b>	<b>48</b>
4.1	Generation of compression signal . . . . .	48
4.1.1	Noise . . . . .	49
4.2	Simulation Data . . . . .	50
4.3	Simulation results . . . . .	51
4.3.1	Peak acceleration deviation . . . . .	54
4.3.2	RMSE performance . . . . .	55

## CONTENTS

---

4.3.3	Time performance . . . . .	58
<b>5</b>	<b>Results</b>	<b>60</b>
5.1	Test procedure . . . . .	61
5.2	Estimate-accelerometer comparison . . . . .	61
5.3	Compression test without resistance . . . . .	64
5.4	Static measurement for determination of spring rate . . . . .	66
5.4.1	Spring TA22 . . . . .	67
5.4.2	Spring TA48 . . . . .	69
5.5	Spring compression test . . . . .	70
<b>6</b>	<b>Discussion and future work</b>	<b>74</b>
6.1	Discussion . . . . .	74
6.1.1	Summary of findings . . . . .	74
6.1.2	Differentiation Estimator . . . . .	76
6.1.3	LDD Estimator . . . . .	76
6.1.4	FCFE Estimator . . . . .	76
6.1.5	KF Estimator . . . . .	77
6.1.6	Previous data acquisition implementations . . . . .	78
6.1.7	Acceleration estimate comparison with simulation . . . . .	79
6.1.8	Measurement deviation . . . . .	79

## CONTENTS

---

6.1.9	Positional error sources . . . . .	82
6.1.10	Impact of pre-load on deviation . . . . .	85
6.2	Further work . . . . .	85
6.2.1	Compliance error compensation scheme using Interferometer . . . . .	85
6.2.2	Compliance error compensation scheme using load cell	86
6.2.3	Improvement of acceleration estimation . . . . .	87
6.2.4	Dynamic load cell performance/time response . . . . .	87
<b>7</b>	<b>Conclusion</b>	<b>89</b>
	<b>Bibliography</b>	<b>95</b>
	<b>Appendix</b>	<b>95</b>
<b>A</b>	<b>Protocol overview</b>	<b>96</b>
A.1	User Datagram Protocol . . . . .	96
A.2	Transmission Control Protocol . . . . .	97
A.3	Hypertext Transfer Protocol . . . . .	98
<b>B</b>	<b>Hardware documentation</b>	<b>99</b>
<b>C</b>	<b>Poster presentation</b>	<b>122</b>

# Chapter 1

## Introduction

CPR-training manikins are essential tools in education and training of both medical and non-medical personnel. In order to perform correct Cardiopulmonary Resuscitation (CPR) quality training is important. One manufacturer of such training products is Laerdal Medical AS, whose main office and development department are situated in the city of Stavanger, Norway.

To ensure quality and longevity of the CPR-training manikins, they are put through extensive testing. Laerdal employs several tools and machines to perform this testing. These machines must be able to emulate human-like compression movements, so that the products are tested in a manner that is as similar as possible to what they would experience during their expected lifetime. The typical lifetime of the manikins are in the hundreds of thousands of compressions.

This work investigates the utilization of a compensation scheme with the aim of increasing the accuracy of load data gathered during compression testing. This data is used for product performance review and internal instrument calibration.

## 1.1 Problem description

---

### 1.1 Problem description

Currently there are two machines at the company's test facilities in Stavanger used to perform compression testing. The first is a compression machine shown in figure 1.1, purpose-built for performing repeated compressions on a test object for hours on end. The machine ensures a high level of accuracy during testing, and has a user friendly interface which allows users to quickly set up their tests. In addition to endurance testing, the compression machine can be used to validate product performance and function as a calibration tool for position and force sensors on a manikin. According to its calibration certificate the compression machine has a dynamic load measurement accuracy of 0.4 [kg] up to a maximum load of 140 [kg] (see appendix B).

However, this machine has its issues. The source code for both machine and interface was written in Labview many years ago. The source code has not been maintained and is in a near non-modifiable state. In addition, the machine can only handle one test object at a time, and some of the machine's parts have become obsolete. This implies that if these parts need replacing, the source-code would need to be modified. Being basically a black-box, the machine and its system lives on the mercy of its obsolete parts not malfunctioning.

The second machine employed by Laerdal is an ABB IRB 6620 Industrial Robot Manipulator shown in figure 1.2. The robot was originally purchased as a flexible system able to perform any test required of it, but it has also seen extensive use for endurance compression testing. Previously this required knowledge of RAPID, the programming language used for making executable procedures for the robot. To eliminate this need and increase user-friendliness, an application to set up and control automated tests was made.

The robot presents itself as a more viable future option for performing this type of testing, being serviceable, highly customizable and being able to perform testing of multiple test objects at a time. However in its current configuration, the robot can not be utilized in the same manner as the compression machine. The biggest hindrance to this is that the robot does not possess the load and position measurement accuracy of the compression

## 1.2 Compression testing

---

machine. This capability is highly desirable to obtain on the robot as it is required to perform product validation.

For the measurement of position data, the robot controller's calculated Tool Center Point (TCP) position is used. For measurement of load data a load cell mounted on the mechanical interface of the robot end effector is used. Regarding both position and load data, only vertical measurements are of interest.

Both position and load data accuracy improvement are required areas of work for the robot to be approved to perform validation testing. However, the main focus of this work is on the improvement of load data accuracy. Since the system as a whole has to be considered, this work also features analysis of the position measurement and its effect on the load data accuracy.

## 1.2 Compression testing

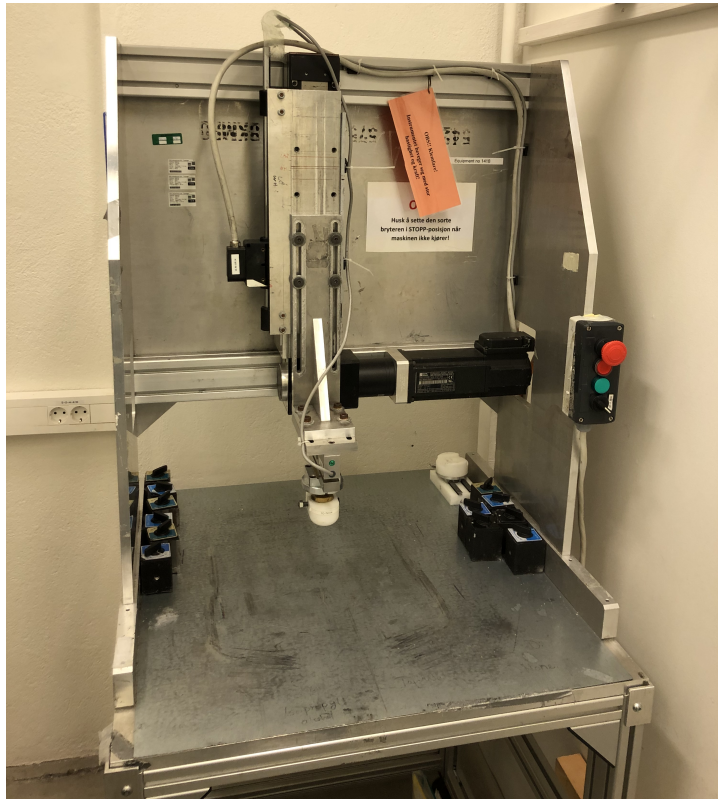
A compression test consist of the robot performing a motion simulating a chest compression at the TCP. According to the European Resuscitation Council (ERC) [1] a high quality chest compression consist of compressions of at least 50 [mm] but no more than 60 [mm] depth and a rate of 100 - 120 compressions per minute (cpm). These values are most commonly used in all types of compression testing, however other values can be used examine performance in border cases. A compression test can serve several different purposes, some of which are listed below, to provide some insight as to why the specified improvements are of importance:

- **Endurance compression test:** Usually an objective of testing is to have a test manikin undergo a specific amount of compressions to simulate lifetime use. These tests typically do not demand high accuracy data, and the robot is currently used for this purpose.
- **Evaluation of product characteristics:** Typically a test is performed between endurance tests to establish changes in mechanical characteristics and internal sensor accuracy, as the product "ages".



## 1.2 Compression testing

---



**Figure 1.1:** Compression machine used at Laerdal Medical. It consists of a stiff steel frame, and an actuator moving the compression tool (white object in the middle of the picture).

The test is typically performed towards the end of product development before making any final adjustments to components. This type of testing requires high accuracy data, and is something the robot currently is not capable of.

- **Validation of sensors and mechanical characteristics:** Once a product has finished its development, it is tested to validate that it fulfills its specification of sensor accuracy and mechanical characteristics. This type of testing also demands high accuracy data and thus, the robot is not capable of this either.

When considering these test cases, the current limitations of the robot become quite clear, thus highlighting the need for the specified improvements.

### 1.3 Approach

---



**Figure 1.2:** ABB IRB 6620 Industrial Robot Manipulator used at Laerdal Medical. The robot is in the testing configuration using a spring in place of a test object.

### 1.3 Approach

The approach is briefly explained in order to clarify the research objectives stated in section 1.4. The hypothesis of what is currently contributing to a significant deviation in measured load data, is the force contribution of the accelerating mass of the compression tool mounted below the load cell. This is analyzed and explained in detail in section 2.1. To minimize the deviation contributed by this force, a simple acceleration estimator based load compensation scheme is proposed. The acceleration estimator uses the measured position of the robot controller. Four acceleration estimators of varying complexity are implemented and compared.

## 1.4 Scope

The research objectives of this task revolve around the performance evaluation of the compensation scheme. The objectives are listed below:

- One objective is to prove the hypothesis described in section 1.3 and examine the degree to which the tool mass contributes to the load data deviation, and whether the compensation scheme can reduce this deviation by a significant amount.
- Another research objective is to establish if any configuration of the compensation scheme can reduce the deviations to within the load accuracy specifications of the compression machine as described in 1.1.
- A third objective is to establish which acceleration estimator provides the best performance, considering performance in isolation and integrated with the compensation scheme.

The method of research of this study is a combination of visual analysis and quantitative evaluation of experimental and simulated results. The combination aims to provide the best possible basis of determining the accomplishment of the stated research objectives.

With respect to the overarching objectives of improvement of both load- and position data accuracy, the elimination of the position data as an objective of accuracy improvement was done during the course of this work to focus resources on providing conclusive results for the improvement of load data accuracy.

## 1.5 Outline

This thesis consists of 7 chapters. Chapter 2 describes the methods used as part of this work from a theoretical viewpoint. This chapter will also attempt to highlight why the methods are relevant in order to achieve the end result, and how the results will be evaluated.

## 1.5 Outline

---

In order to gain additional insight in the general setup in regards to practical experiments, section 3 describes the design and function of the testing environment and robot cell, implementation in software, third party technology, communication protocols, and technical description of measurement tools.

Chapter 4, presents an initial review of the acceleration estimator performance, based on simulated data, with the aim of confirming their proper functionality. In this section, the comparative time performance of calculations are also simulated.

Chapter 5 includes the presentation of the results from the practical experiments performed on the robot, including initial comments and comparison on their performance.

Chapter 6 contains discussion on overall findings and topics that are relevant to consider in regards to results. Topics of improvement and further work are also found here.

Finally, chapter 7 presents a short review of the work that has been carried out and concludes upon the results of both simulation and practical experiments.

## Chapter 2

# Theory and Method

This section provides an explanation of the proposed method for dynamic load compensation. First, the theoretical basis on which the compensation scheme is built on is explained. An overview on the fields of load compensation and acceleration estimation are also provided. A description for each of the acceleration estimators that are evaluated as part of the load compensation scheme is provided. Finally, some remarks on error sources in industrial robots and how they occur, specifically relating to position error, which will also need to be taken into account when evaluating the results in chapter 5.

### 2.1 Dynamic load compensation

The aim of this thesis is to design a dynamic load compensator without direct information on TCP acceleration, instead using an estimation of this parameter. The purpose is to examine whether peak load accuracy can be improved without using additional sensors, such as accelerometers or gyroscopes.

In principle, the dynamic system under consideration consist of the following:

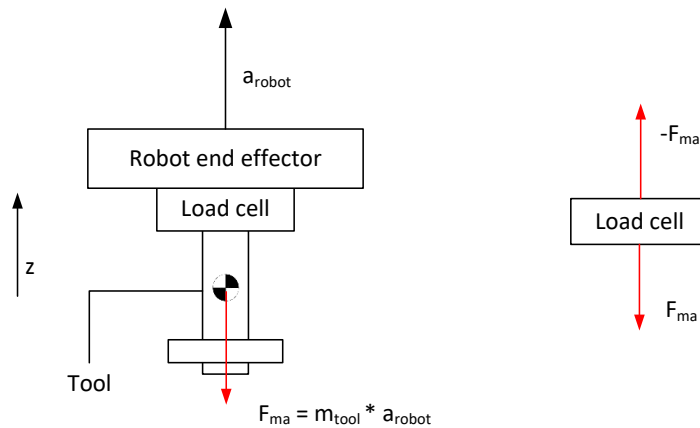
## 2.1 Dynamic load compensation

---

1. The robot
2. The load cell, mounted onto the end of the robot arm
3. A tool (with an unknown mass) mounted onto the load cell tool interface
4. A compression test object (usually a CPR-manikin). In this work, linear springs are used in experimental tests.

Linear springs are used as their spring rate can be found experimentally. Using the position measurement it is possible to calculate the expected spring force to use as a reference for evaluation of results.

Consider first the system with no spring when the robot is at the bottom of a compression accelerating upwards, as shown in figure 2.1. Since the load cell is not moving relative to the robot and the compression tool, the sum of forces acting on it must be zero. However, since there is an accelerating mass mounted on the measurement surface of the load cell, the load cell will output a force equal to the mass multiplied by the negative robot acceleration. This condition is tested in section 5.3.



**Figure 2.1:** Forces acting on the system without spring resistance.

Now consider the system with the spring as shown in figure 2.2, and the following key points in one compression cycle:

## 2.1 Dynamic load compensation

---

1. Starting point: The compression tool and spring are firmly in contact, with a small force applied (typically 5 [N]).
2. Robot accelerates and moves downwards.
3. Robot accelerates upwards while still moving downwards before reaching its bottom point.
4. Compression has reached the bottom. Momentarily the system can be considered as not being in motion. This (or at least very close) is where peak upwards acceleration is expected to occur.
5. Robot accelerates and moves upwards.
6. Robot accelerates downwards while moving upwards before reaching top point.

In the data presented in chapters 4 and 5, upwards acceleration are represented by positive values and vice versa.

The only difference in the cases of figures 2.1 and 2.2 is that the latter also includes the spring force  $F_{ref}$  which is the force that we want to measure. However, like in the case of figure 2.1 the force  $F_{tool}$  is also measured by the load cell.

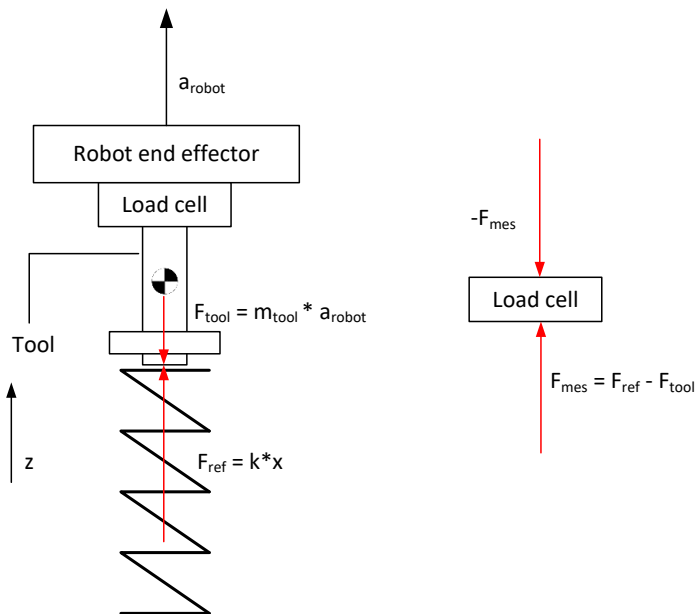
In order to know how much error the acceleration of the tool is contributing to the measured load, two parameters must be known:

- The mass of the currently mounted tool. (Meaning all mass mounted below the load cell that is connected to the tool assembly.)
- The current acceleration of the robot TCP.

The mass of the tool is given by the stationary reading of the load cell mounted on the robot. This measurement is performed at the start of each test. The robot does not currently have any measurement instruments for acceleration, which means it must be estimated from the available position data. Requirements for the proposed compensation scheme are listed below:

## 2.1 Dynamic load compensation

---



**Figure 2.2:** Forces acting on the system under dynamic conditions from the perspective of the load cell.

- The force sampling rate should be high enough to capture peak force.
- The sampling rate of the acceleration should be the same as the force, to ensure that compensation can be done at every time-step.
- The estimation calculation time should be within one sample interval, so that estimation can be performed in real-time.
- There should be minimum latency of the estimation as not to compromise the synchronicity of the compensation scheme.

### 2.1.1 Related work

The problem of end-effector dynamics acting as disturbances on a wrist-mounted load cell is well described in chapter 2 of [2]. Typically the external loads are of interest, but cannot be detected directly because of the load cell



## 2.1 Dynamic load compensation

---

not being mounted at the point of contact. The problem is well established given the publication year of this article.

Collaborative robots have become increasingly popular for use in the industry in later years, due to their internal safety mechanisms making them easier to implement in areas where humans are also required to work. Thus, in recent years there has been work focusing on techniques to detect collisions in collaborative environments. Although collision detection is not the aim of this thesis, it has some similarities as it deals with dynamic load compensation.

For example one study on dynamic load compensation using a quaternion-based kalman filter for admittance control has been done [3]. This work describes a control architecture for industrial robotic applications allowing human/robot interactions, using an admittance control scheme and direct sensing of human inputs.

A similar approach was taken in [4] with the aim of separating contact and non-contact forces using a load cell mounted on the end-effector. This work also considers force and torques on all directions. The non-contact forces were compensated for by the use of velocity and acceleration estimation based measurement on joint encoder measurements.

This work is similar in the sense that a load cell is mounted on the robot TCP to measure force input, however the control architecture is intended to sense forces and torques in all directions, which is not a requirement in this work. Also, the focus in this thesis is not on control but on observation of a single axis, thus making the estimators described in these papers unnecessarily complex.

Other examples include works such as [5] and [6] however the research is targeted at a different goal than the work in this thesis. Thus, this thesis bases its compensation scheme on its own aim and is thus kept much simpler than the aforementioned examples.

## 2.2 Compensation scheme

---

## 2.2 Compensation scheme

Consider the equation shown in figure 2.2 expressing the combined force measured by the load cell:

$$F_{mes} = F_{ref} - F_{tool} \quad (2.1)$$

Moving  $F_{tool}$  to the other side of the expression leaves a basic model of the force exerted on the tool by the spring in dynamic conditions, as explained in section 2.1:

$$F_{ref} = F_{mes} + F_{tool} \quad (2.2)$$

Where  $F_{ref}$  is the spring force on the compression object. This force can also be expressed as:

$$F_{ref} = kx \quad (2.3)$$

Where  $k$  is the spring rate and  $x$  is the spring displacement. However this cannot be used in the compensation scheme as usually the spring rate is not known.  $F_{ref}$  is the value it is desirable to obtain by compensation.  $F_{mes}$  is the force output read from the load cell, and  $F_{tool}$  is the force contribution from the accelerating mass between the test object and load cell.

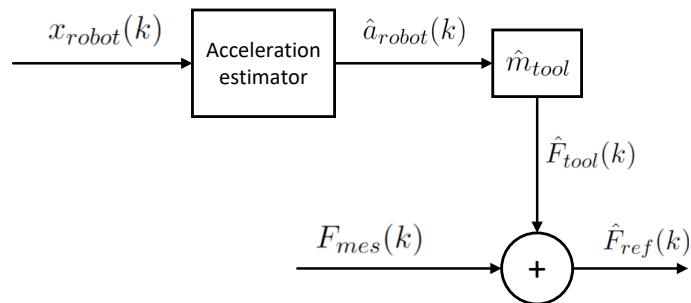
Expanding the expression for the dynamic contribution according to figure 2.2, the equation looks like the following:

$$F_{ref} = F_{mes} + a_{robot} \cdot m_{tool} \quad (2.4)$$

Where  $a_{robot}$  is the acceleration of the robot TCP and  $m_{tool}$  is the mass of the end-effector tool. However, the exact acceleration of the robot is not known. Neither is the exact mass, however as mentioned these parameters can be estimated, thus yielding an estimate of the spring force:

## 2.3 Acceleration estimation

---



**Figure 2.3:** Overview of compensation scheme.

$$\hat{F}_{ref} = F_{mes} + \hat{a}_{robot} \cdot \hat{m}_{tool} \quad (2.5)$$

Where  $\hat{F}_{ref}$ ,  $\hat{a}_{robot}$  and  $\hat{m}_{tool}$  are the estimates for spring force, robot TCP acceleration and mass of the end-effector tool respectively. Thus the the simple acceleration estimator based compensation scheme can be expressed as a discrete sample-wise calculation:

$$\hat{F}_{ref}(k) = F_{mes}(k) + \hat{a}_{robot}(k) \cdot \hat{m}_{tool} \quad (2.6)$$

Due to the filtering undergone by the acceleration estimators and the load cell, it should not be necessary to perform further filtering as part of the compensation scheme. As the performance at maximum compression depth is of most interest, any dampening of the signal response should be avoided if possible. An overview of the load compensation scheme can be seen in figure 2.3.

## 2.3 Acceleration estimation

Much work has been done on the estimation of velocity and acceleration based on position measurements. Through the design and evaluation of a

## 2.3 Acceleration estimation

---

large variety of methods, many robust, accurate and effective methods have been proposed.

A comparison between several estimation methods were made on a servopneumatic system in [7], based on digital encoder measurements. The estimators were implemented for both velocity and acceleration. The authors evaluated and compared five fixed-time methods; Backwards Finite Difference (BFD), Savitzky-Golay Filter (SGF),  $\alpha\beta\gamma$  Filter, Kalman Filter(KF) and Enhanced Adaptive Windowing Filter (EAW). The methods were compared using combinations of sampling rates and encoder resolutions. The work concluded that KF had the best performance for both velocity and acceleration predictions across different position and time resolutions.

In [8], the authors applied the Kalman filter as an observer in order to estimate velocity and acceleration based on measuring the output from the position. However, the method was based on a dynamical frequency model to increase operation speed range. The experiments in [8] used a single motor for the experimental data and thus this frequency model, does likely not apply to the robot.

The author of [9] considers the estimation of encoder angular velocity and acceleration in with respect to robotic motion by use of the KF method. The work includes models of the KF for both acceleration and velocity estimation. However, the model is based on angular motion which means that the model does not directly apply to the case of estimating the robot TCP. In [10] an embedded implementation of the KF is optimized for use in vehicle anti lock braking systems (ABS). The implemented model is similar to the KF model used in cite [7] although also here, the model parameters are angular instead of linear.

The method presented in [11] is aimed at estimation in regards to robotic manipulators. The authors propose a method to fuse information from joint position measurement with link accelerometers to estimate joint velocity and acceleration. The method employs an Extended Kalman Filter (EKF) to linearize the proposed nonlinear model around the estimated state.

As the KF is a widely used method of acceleration estimation, an implementation will be considered in this work. The method and implementation is described in more detail in section 2.3.4. The KF has also seen use as

## 2.3 Acceleration estimation

---

integration estimator such as in [12] where the authors used gyroscope and magnetometer measurements to estimate velocity.

In [13] the combination of position and acceleration measurements are used to obtain robust velocity estimates. As this method employs measurement of acceleration using sensors, it will not be considered for this work. A high-order disturbance observer to obtain joint acceleration estimation is presented in [14]. Also here, the observer is used to estimate robot joints. In [15] a method for estimating velocity and acceleration in optical encoders is proposed.

In [16] an enhanced linear differentiator (LD) was compared with a nonlinear tracking differentiator (TD) and an enhanced nonlinear differentiator (END), and was found to have outperform the TD, while having similar performance as the END despite being a simpler implementation. The LD is described in detail in section 2.3.2, as it is considered in this work.

As stated, much work has been done on this topic of research, and although not all methods can be put to the test as part of this work, an attempt has been made to choose a selection of methods which represent variety both in complexity, robustness, ease of implementation and computational efficiency. Numeric differentiation by means of the Backwards Finite Difference method was evaluated as part of [7], and will be taken into consideration.

The fast curve fitting velocity estimator (FCFVE) was presented as a more efficient alternative to least-square curve fitting velocity estimators (LSVE) in [17] and will therefore be considered in this work. The implementation of this method is described further in 2.3.3.

Each estimation method is tested on simulation data, and on compression tests performed on the robot. These results are presented in chapter 4 and 5 of this thesis, respectively.

## 2.3 Acceleration estimation

---

### 2.3.1 Numeric differentiation by Second Order Backwards Finite Difference

A common technique for numeric approximation of derivatives is the finite difference method. In this work, position measurements are used and the second order backwards finite difference approximation is employed to find an estimate of acceleration. Further in this thesis the method is simply called differentiation. It is a very simple method of obtaining acceleration estimates. It does not require complex computation, however the estimates can be affected by resolution and sampling rate. According to [9] estimates can suffer degradation as sampling rates and quantization levels increase. Additionally it suffers from noise sensitivity, especially for acceleration estimates, and from poor low-speed performance [9]. Equation (2.7) describes the method for acceleration estimation in continuous time domain.

$$\hat{a}(t) = \frac{y(t) - 2y(t - T_s) + y(t - 2T_s)}{T_s^2} \quad (2.7)$$

Where  $\hat{a}(t)$  is the acceleration estimate,  $y(t)$  represent position measurement at time  $t$ , and  $T_s$  is the sampling period. In order to illustrate the implementation more clearly, the equation is also presented in the discrete time domain, where the estimation is done for sample  $k$ .

$$\hat{a}(k) = \frac{y(k) - 2y(k - 1) + y(k - 2)}{T_s^2} \quad (2.8)$$

### 2.3.2 Linear Tracking Differentiator

The linear tracking differentiator (LD) presented in [16] is constructed based on the fact that velocity belongs to the chain of kinematic quantities: position and velocity, and that numerical integration can provide more stable and accurate results than numerical differentiation in the presence of noise. The method among other works based based on a nonlinear velocity estimator (NVE) proposed in [18]. The LD is model-free as opposed to KF's or

## 2.3 Acceleration estimation

---

other observer based approaches. It does have some similarities to the KF, such as improving estimates based on measurement-estimate errors. Difference being that its design parameters are not based on a system model. Instead, the parameters are tuned experimentally. The method is fairly easy to implement and uses simple calculations for each estimate update. For example, there are no matrices that need to be solved.

On the same basis of reasoning as presented in chapter 2 of [16], the established chain of kinematic quantities are followed by a third parameter, which is acceleration. It is also assumed that the acceleration is constant between measurement intervals. Consider that  $x_1$  is the position measurement. Thus, a linear tracking double differentiator (LDD) is proposed:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -\beta_1 R^3(x_1 - r) - \beta_2 R^2 x_2 - \beta_3 R x_3 \end{cases} \quad (2.9)$$

Where the parameter  $r$  is defined as the previous estimate of the position. This is further clarified as done in [16], by applying the Euler discretization method to obtain discrete LDD for implementation:

$$\begin{cases} \hat{p}(k+1) = \hat{p}(k) + T\hat{v}(k) \\ \hat{v}(k+1) = \hat{v}(k) + T\hat{a}(k) \\ \hat{a}(k+1) = \hat{a}(k) + T(-\beta_1 R^3 \epsilon(k) - \beta_2 R^2 \hat{v}(k) - \beta_3 R \hat{a}(k)) \end{cases} \quad (2.10)$$

where  $\hat{p}$ ,  $\hat{v}$ , and  $\hat{a}$ , are the estimated position, velocity and acceleration, respectively,  $\epsilon(k) = \hat{p}(k) - p(k)$  is the position estimation error,  $p$  is the measured position,  $T$  is the sampling period,  $k$  denotes the  $k$ th sampling instant, and  $R > 0$  and  $\beta_1, \beta_2, \beta_3 > 0$  are design parameters.

### 2.3.3 Fast Curve Fitting Estimator

Traditional Least Square Curve Fitting (LSCF) use least square calculation to approximate a n-th degree polynomial to a data set. This concept is

## 2.3 Acceleration estimation

---

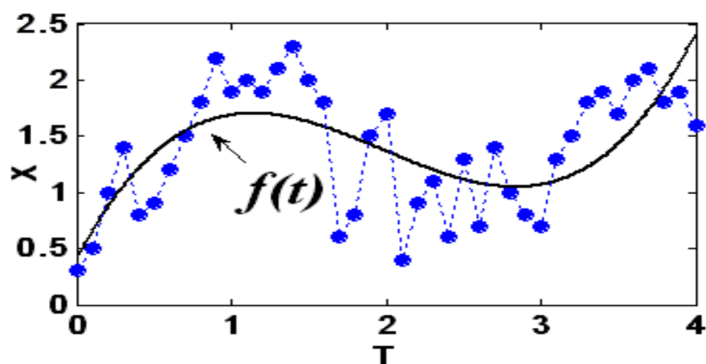


Figure 2.4: Curve fitting of a polynomial to a data set. Figure is from [17]

illustrated in figure 2.4. In an attempt to improve on the widely used LSVE [19], the authors of [17] designed a more computationally efficient method of velocity estimation by curve fitting, called FCFVE. The technique was compared to a traditional LSVE and a LD <sup>1</sup> and was found to have better performance than both other techniques when tested on a sinusoidal input reference signal with frequency of 20[Hz] and 1[Hz]. A compression test will typically operate with sinusoidal movements of around 2[Hz], making this estimator a valid candidate for evaluation.

As with the LD, the work presented in [17] only covers a velocity estimator, and for the estimator to be useful in this case it must be extended to an acceleration estimator. This extension will be presented as a fast curve fitting estimator (FCFE).

The technique is based on the approximation of a theoretical function  $f(t)$  in a small region of samples. When the function  $f(t)$  has been approximated, the derivative  $\frac{df(t)}{dt}$  is computed, thus providing a velocity estimate at the point of evaluation on the curve. The nature of this method makes it rather simple to extend into an acceleration estimator. The acceleration is found by finding the double derivative  $\frac{d^2f(t)}{dt^2}$  of the approximated function  $f(t)$ .

The curve fitting is done by solving an  $n^{th}$ -order matrix equation. For the velocity estimator in [17] a  $3^{rd}$  order equation is used. The computational

---

<sup>1</sup>Note that even though the methods LD and FCFVE have been compared previously, it is still of interest to make the comparison again, given the novelty introduced to each method by expanding them from VE to VAE.



### 2.3 Acceleration estimation

---

advantage comes from restricting the algorithm to perform polynomial fitting of the 3<sup>rd</sup> order, as opposed to LSVE where the algorithm attempts to find the best fitting  $n^{\text{th}}$  degree polynomial [19], which in most cases will increase computational demand per sample.

Since the second order derivative is to be used in this work, the order of the fitting equation is increased to 4. This is done to ensure adequate accuracy, but comes at the cost of computational efficiency. Still, the order limitation is kept as with the the FCFVE, in contrast to the LS method. Equations 2.11 - 2.17 shows the steps of performing curve fitting for a single sample.

The fifth order polynomial for the approximation of the function around the point  $x_0$  is expressed as:

$$f'(t) = x' = x_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 \quad (2.11)$$

Where  $f'(t)$  denotes the time translation  $f(t - \Delta t)$  of  $f(t)$ , meaning that the function can be estimated at all samples by shifting with  $\Delta t$ . The zero order term is transposed, resulting in the following equation:

$$\Delta x = x' - x_0 = a_1t + a_2t^2 + a_3t^3 + a_4t^4 \quad (2.12)$$

The 4<sup>th</sup>-order matrix equation is expressed as:

$$\begin{bmatrix} x_1 - x_0 \\ x_2 - x_0 \\ x_3 - x_0 \\ x_4 - x_0 \end{bmatrix} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta x_4 \end{bmatrix} = \begin{bmatrix} t_1 & t_1^2 & t_1^3 & t_1^4 \\ t_2 & t_2^2 & t_2^3 & t_2^4 \\ t_3 & t_3^2 & t_3^3 & t_3^4 \\ t_4 & t_4^2 & t_4^3 & t_4^4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad (2.13)$$

Solving as done in [19]:

$$m_4 = \begin{bmatrix} 1 & 1^2 & 1^3 & 1^4 \\ 2 & 2^2 & 2^3 & 2^4 \\ 3 & 3^2 & 3^3 & 3^4 \\ 4 & 4^2 & 4^3 & 4^4 \end{bmatrix}^{-1} \quad (2.14)$$

## 2.3 Acceleration estimation

---

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = m_4 \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta x_4 \end{bmatrix} = \begin{bmatrix} t_1 & 0 & 0 & 0 \\ 0 & t_1^2 & 0 & 0 \\ 0 & 0 & t_1^3 & 0 \\ 0 & 0 & 0 & t_1^4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad (2.15)$$

Yields the resulting curve fitting coefficient vector:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} b_1 t_1^{-1} \\ b_2 t_1^{-2} \\ b_3 t_1^{-3} \\ b_4 t_1^{-4} \end{bmatrix} \quad (2.16)$$

The double derivative of equation 2.11 yields the following:

$$\frac{d^2 f(t)}{dt^2} = 2a_2 + 6a_3 t + 12a_4 t^2 \quad (2.17)$$

### 2.3.4 Kalman Filter

The Kalman Filter (KF), first described over 60 years ago in [20], has seen many applications of which a review was presented in [21]. The KF is an optimal recursive estimator of the state of an uncertain dynamic system. This method is the only model-based approach considered as part of this work. The following equations describe a generalized discrete system which can be used as the basis of a KF:

$$x(k+1) = Ax(k) + Bu(k+1) + Gw(k) \quad (2.18)$$

$$y(k) = Cx(k) + v(k) \quad (2.19)$$

where  $A$  is the state matrix,  $B$  is the system input matrix,  $C$  is the measurement matrix,  $G$  is the process noise matrix,  $u(k)$  is the input vector,

### 2.3 Acceleration estimation

---

$w(k)$  is the process noise vector,  $v(k)$  is the measurement noise vector and  $y(k)$  is the measurement vector.

The KF uses a prediction-update algorithm described by equations (2.20) - (2.24):

Prediction:

$$\bar{x}(k+1) = A\hat{x}(k) + Bu(k) \quad (2.20)$$

$$M(k+1) = AP(k)A^T + Q \quad (2.21)$$

Where  $\bar{x}(k+1)$  is the predicted state estimate,  $\hat{x}(k)$  is the previous updated state estimate,  $M(k+1)$  is the estimate covariance and  $Q$  is the process noise covariance. Update:

$$P(k) = M(k) - M(k)C^T(CM(k)C^T + R)^{-1}CM(k) \quad (2.22)$$

$$L(k) = P(k)C^TR^{-1} \quad (2.23)$$

$$\hat{x}(k) = \bar{x}(k) + L(k)(y(k) - C\bar{x}(k)) \quad (2.24)$$

Where  $P(k)$  is the updated estimate covariance,  $L(k)$  is the optimal Kalman gain and,  $\hat{x}(k)$  is the update state estimate. For this application the process and measurement noise covariance are considered time-invariant.

The physical system under consideration is a 6-DOF Robot Manipulator, which would lead to a complex model if all the links and joints were considered. For this work the observation of the TCP movement along a single linear axis is what is of interest and thus, the model will follow this approach. I.e. the robot TCP is modeled as an autonomous uncertain system with states position, velocity and acceleration as done in [7]:

### 2.3 Acceleration estimation

---

$$\mathbf{x} = [x \quad \dot{x} \quad \ddot{x}]^T \quad (2.25)$$

$$\mathbf{A} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

$$\mathbf{B} = [0 \quad 0 \quad 0]^T \quad (2.27)$$

$$\mathbf{G} = \left[ \frac{T_s^2}{2} \quad T_s \quad 1 \right]^T \quad (2.28)$$

$$\mathbf{C} = [1 \quad 0 \quad 0] \quad (2.29)$$

Where  $T_s$  is the sampling period. The process and measurement noise are assumed to be white noise with a gaussian distribution according to  $p(w(t)) \sim N(0, Q)$  and  $p(v(t)) \sim N(0, R)$  where  $Q$  and  $R$  are described as:

$$\mathbf{Q} = E[ww^T] = GG^T \sigma_w^2 = \begin{bmatrix} \frac{T_s^4}{4} & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^3}{2} & T_s^2 & T_s \\ \frac{T_s^2}{2} & T_s & 1 \end{bmatrix} \cdot \sigma_w^2 \quad (2.30)$$

$$\mathbf{R} = [\sigma_{encoder}^2] = \left[ \frac{\Delta y_{min}^2}{12} \right] \quad (2.31)$$

The process noise  $\sigma_w$  is decided according to the equation (60) in [7] as shown in equation 2.32:

$$\sigma_w = \frac{T_s \times \hat{a}_{max}}{K} \quad (2.32)$$

## 2.4 Performance evaluation

---

Where  $T_s$  is the sampling period,  $\hat{a}_{max}$  is the maximum jerk value and  $K$  a constant. The value for  $\hat{a}_{max}$  was decided by simulation using the FCFE method to be  $17000[mm/s^3]$  while the value for  $K$  was set to 3 according to [7].

Strictly speaking, this implementation of the KF can be said to not be model-based, since the system is not modeled on any other physical parameters than the ones that are used to describe the motion of the system. This is further discussed in chapter 6.

## 2.4 Performance evaluation

The load compensation performance will be evaluated by a set of tests to determine the accuracy and robustness of the estimators themselves and for each estimator as part of the compensation scheme. The following tests are performed in chapter 4 and 5:

- Simulation comparison with analytical double derivative of optimal position path, the path is described in chapter 4.
- Comparison between acceleration estimates and accelerometer data.
- Test with robot performing unloaded compressions, meaning that the robot is compressing in free-air.
- Dynamic compression trial, with compressions against a linear compression spring.

The first simulation trial is to verify that the estimators are behaving as expected in a simulated environment, and that they are performing with reasonable accuracy. The performance will be evaluated based on quantitative measures such as Root-mean-square Error (RMSE) and average peak accuracy, as well as visual evaluation of estimator plots.

An accelerometer is mounted on the robot tool used during tests, and this will provide the opportunity to directly evaluate the performance of the estimators in the actual testing environment. The acceleration estimate is

## 2.4 Performance evaluation

---

measured against the accelerometer data to determine the accuracy of the estimates.

The unloaded compression tests will provide a direct measure of the load compensation scheme's ability to compensate for the dynamic error generated by the robot in motion. The compensated force output will be measured against the ideal compensation value, which is always zero.

The final test using a compression spring will give a tangible measure of the actual compensation performance during a test. This experiment reflects the typical conditions for an actual compression test, and thus is the most relevant for evaluation of overall performance. The compensated output will be measured against the spring reference value found by equation 2.3. A separate test to determine spring rate  $k$  is done in section 5.4.

### 2.4.1 Performance metrics

**Average deviation/error:** For the evaluation of peak accuracy, the estimates are in this thesis measured against peak value references. This is done both with respect to peak acceleration and peak load in chapters 4 and 5.

Let  $x_i$  be the reference of a given value and  $\hat{x}_i$  its estimate. For a sample of size  $n$  the following expression for the average deviation is used in this work:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i) \quad (2.33)$$

Where  $\bar{x}$  is the resulting average deviation.

**Root mean square error (RMSE):** Applied to evaluate the overall accuracy of performance estimators either with reference to an analytical function in 4 or to an accelerometer in 5. Using the same parameters for reference value, estimate and population size, the following expression is used:

## 2.4 Performance evaluation

---

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (2.34)$$

### 2.4.2 Uncertainty of measurement

The estimate of the combined uncertainty of results in section 5.5 is found in accordance with the Guide to the expression of uncertainty in measurement (GUM) [22]. For other results presented, either standard deviation or confidence interval is used as deemed fitting to illustrate the uncertainty of the data.

An objective of the robot is to make a calibration certificate for load and depth, and a calibration certificate requires the combined uncertainty to be identified. Thus the results presented as part of this work will take this into consideration.

For the determination of the uncertainties of the measurement first, the uncertainty of the spring rate  $k$  for the springs used in section 5.5 must be found. The formula for the spring force is expressed as:

$$F = kx = mg \quad (2.35)$$

Where  $k$  is the spring rate,  $x$  is the depth,  $m$  is the load from the scale, measured in [kg] that the spring is exposed to, and  $g$  is the gravity constant. Rearranging this equation gives the expression for the spring rate:

$$k = \frac{mg}{x} \quad (2.36)$$

According to section 5.1 of [22], *Determining combined standard uncertainty - Uncorrelated input quantities* the uncertainty coefficients are found by finding the partial derivatives of this expression:

$$c_{km} = \frac{\delta k}{\delta m} = \frac{g}{x} \quad (2.37)$$

## 2.4 Performance evaluation

---

$$c_{kx} = \frac{\delta k}{\delta x} = \frac{-mg}{x^2} \quad (2.38)$$

The numerical values of the uncertainty coefficients are determined by the average values with regards to  $m$  and  $x$  and while  $g$  has the value 9.81.

According to [22], section 4, uncertainties are divided into to categories; Type A and Type B standard uncertainty. Type A standard uncertainty is estimated by the variance of the mean of a data set:

$$s^2(\bar{q}) = \frac{s^2(q_k)}{n} \quad (2.39)$$

Where  $s^2(q_k)$  is the estimate of the variance and  $n$  is the number of independent observations. Type B standard uncertainties represent all uncertainties that are not obtained by repeated observations. In this case these uncertainties are found in equipment calibration certificates (see appendix B).

For the spring rate the following uncertainties are considered:

Spring rate uncertainty					
Type	Input quantity	Distribution	Value	Standard uncertainty	Sensitivity
A	Weight from scale	Normal	$m$	$u(m)$	$c_{km}$
B	Scale deviation	Square	$dev_S$	$u(dev_S)$	$c_{km}$
B	Scale resolution	Square	–	$u(res_S)$	$c_{km}$
A	Spring comp. depth	Normal	$x$	$u(x)$	$c_{kx}$
B	Interferometer dev.	Square	–	$u(dev_I)$	$c_{kx}$
B	Interferometer res.	Square	–	$u(res_I)$	$c_{kx}$

**Table 2.1:** Spring rate uncertainty

For the determination of the spring rate done in section 5.4, a scale is used as a reference measurement. The uncertainty parameters of this is specified in table 2.1. Once the sources of uncertainty are identified, the combined standard uncertainty can be calculated according to section 5.1 of [22]:



## 2.4 Performance evaluation

---

$$u(k)^2 = [c_{km}u(m)]^2 + [c_{km}u(dev_S)]^2 + [c_{km}u(res_S)]^2 + [c_{kx}u(x)]^2 + [c_{kx}u(dev_I)]^2 + [c_{kx}u(res_I)]^2 \quad (2.40)$$

Thus, the uncertainty for the spring rate can be identified by taking the root of equation 2.40. The results presented in section 5.5 are measured against the reference force at the given depth of the compressions given as:

$$F_{ref} = kx \quad (2.41)$$

To find the uncertainty of this reference load, the same procedure is repeated as when the spring rate uncertainty was found. The respective uncertainty coefficients are shown in equations 2.42 and 2.43:

$$c_{Fk} = \frac{\delta F_{ref}}{\delta k} = x \quad (2.42)$$

$$c_{Fx} = \frac{\delta F_{ref}}{\delta x} = k \quad (2.43)$$

<i>F<sub>ref</sub></i> uncertainty					
Type	Input quantity	Distribution	Value	Standard uncertainty	Sensitivity
A	Spring rate	Normal	<i>k</i>	<i>u(k)</i>	<i>c<sub>Fk</sub></i>
A	Recorded depth avg.	Normal	<i>x</i>	<i>u(x)</i>	<i>c<sub>Fx</sub></i>

**Table 2.2:** Spring rate uncertainty

Note that the value *x* in table 2.2 is not the same value as in table 2.1. The spring rate is determined by static measurements as presented in section 5.4. This spring rate is then used together with the average recorded dynamic depth measurement to determine the reference value in section 5.5. The combined uncertainty for the reference load *F<sub>ref</sub>* can be expressed as:

$$u(F_{ref})^2 = [c_{Fk}u(k)]^2 + [c_{Fx}u(x)]^2 \quad (2.44)$$

## 2.4 Performance evaluation

---

Finally, the complete uncertainty can be calculated based on the reference and compensated output estimate. The error presented in section 5.5 is expressed as:

$$F_{err} = F_{ref} - \hat{F}_{ref} \quad (2.45)$$

Since both parameters have unit [N], the uncertainty coefficients are equal to 1. In addition to the reference and compensated estimate, static load cell calibration deviation is included in the uncertainty table

$F_{err}$ uncertainty					
Type	Input quantity	Distribution	Value	Standard uncertainty	Sensitivity
A	Reference load	Normal	$F_{ref}$	$u(F_{ref})$	1
A	Load comp. est.	Normal	$F_{comp}$	$u(F_{comp})$	1
A	Static load cell cal.	Normal	$dev_{LC}$	$u(dev_{LC})$	1

**Table 2.3:** Spring rate uncertainty

The combined uncertainty of the load compensation error can be expressed as:

$$u(F_{err})^2 = [u(F_{ref})]^2 + [u(F_{comp})]^2 + [u(dev_{LC})]^2 \quad (2.46)$$

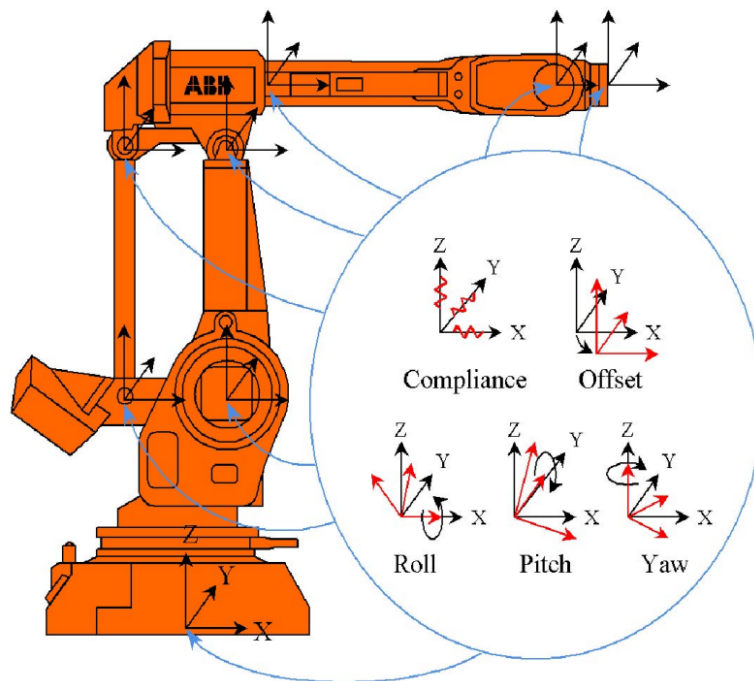
This combined uncertainty is shown as error caps for the results presented in section 5.5. According to [22], for a confidence interval of 95 % the resulting uncertainty is multiplied by a factor  $k$  which is equal to 2. Note, that the results in section 5.5 are presented as percentage of the reference value. Naturally the same is true for the uncertainty. As stated, the identification of this uncertainty has been a vital part of the performance evaluation done in this thesis. It should be stated that the uncertainty estimate is not absolute, but the most influential sources of uncertainty have been included.

### 2.5 Robot error sources

It has been discovered during previous attempts to perform a compression depth calibration of the robot, that a non-trivial internal position error arises when performing compressions at typical compression depths, which are around 50 to 60 [mm].

The ABB IRC5 application manual [23], separates the main error sources that can be compensated for in the controller into two main categories: Compliance errors and kinematic errors. An overview can be seen in figure 2.5.

Compliance errors occur due to the effect of the robots own weight and the weight of the current payload. These errors depend on gravity and the characteristics of the load. Kinematic errors are caused by position or orientational errors in the robot axes, and are independent of load [23].



**Figure 2.5:** Overview of error sources specified by ABB from [23].

## 2.5 Robot error sources

---

One type of mechanical error present in all industrial robots is backlash. Backlash is caused by mechanical tolerances in joint gears and occur when the direction of motion of an actuated joint is reversed, which happens twice during one compression cycle. In [24] the authors propose an experimental approach for evaluating the backlash error of an ABB industrial robot. It is stated by the authors that the focus of their work is measurement along a linear path. This makes their findings relevant for consideration in this thesis. Their results show that backlash error strongly affects the repeatability of the robot. The error is dependent on robot configuration and TCP speed, however it is nearly unaffected by load. For the case of compression tests, the effects of backlash error should be considered when increasing compression rate.

As the robot system in question will be exposed to high loads, compliance errors are expected to occur. The field of compliance error compensation within industrial robotics has seen several contributions. Online compensation of compliance error is not in the scope of this thesis, however it is of value to consider the work that has been done in this field, to gain insight as to how the errors occur. In [25] the authors propose a compensation system for industrial robots in contact applications. They use an elasto-geometric robot model combined with a load sensor mounted between the robot mechanical interface and end effector similarly to the robot system used in this thesis. The authors of [26] perform similar compensation based on use of a laser tracker. During testing done as part of this thesis an external measuring tool called an interferometer is used to measure absolute compression depth. Its function is further explained in chapter 3.

As mentioned, the compensation of compliance and kinematic errors will not be considered in this work. However it is useful to have some background information on the topic. The topic is also further discussed in chapter 6, with emphasis on the measured results.

## Chapter 3

# Description of hardware and software

This chapter describes the structure and design of hardware and software used and/or developed as part of this thesis. An explanation of the robot cell and test setup will be given, followed by a walk-through of the structure of the compression test application which is the end user product. Following this, an explanation of the load compensation application that has been developed is given. Furthermore, a description of the protocols that have been used is provided. Lastly, an overview of the external measuring equipment is provided.

### 3.1 Robot cell and testing environment

The robot cell installation at Laerdal Medical's test facility which can be seen in figure 3.1, consist of an ABB IRB 6620 Robot, enclosed by three solid walls, and one wall with sliding glass doors. A security lock ensures that the door stays locked when the robot is in automatic mode. A large portion of the floor is fitted with a steel sheet, which enables test objects to be locked in place by magnets. This can be seen in figure 3.2. A variety of compression tools are available, tailored to specific test requirements and the characteristics of the test objects under examination. Thus, the use

### 3.2 Robot Compression Testing Application

---

of the robot is based on quick manual interchange between different tools, which means that the compensation algorithm needs to be adaptable to different tool masses. The controller and the connected computer is located just outside the glass doors, providing users with a good view of the test area.



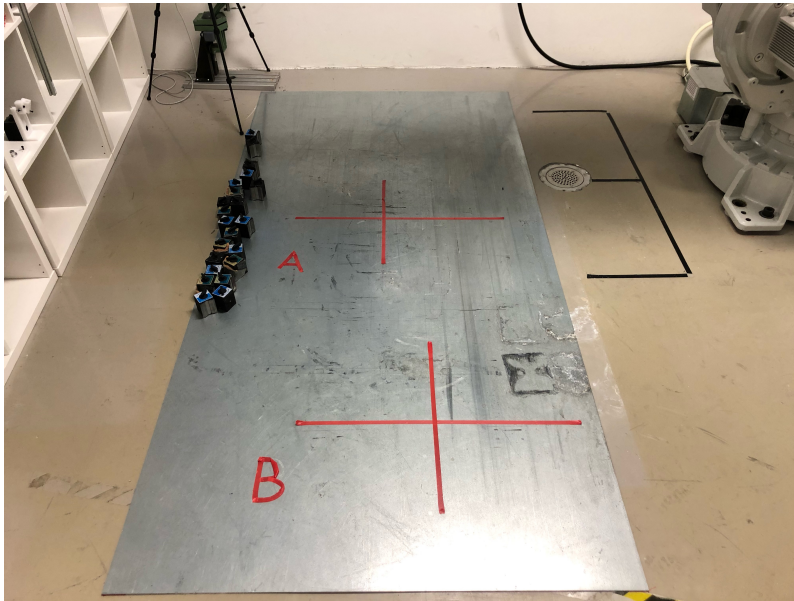
**Figure 3.1:** Robot cell and controller station.

### 3.2 Robot Compression Testing Application

The compression test application described in chapter 1 was made to lower the user-threshold for the robot, and to ensure that compression testing could be done in a similar manner to that of the compression machine also mentioned in chapter 1. The application consists of a Graphical User Interface (GUI) and its functionality, and a backend handling communication between the robot and the controller.

## 3.2 Robot Compression Testing Application

---



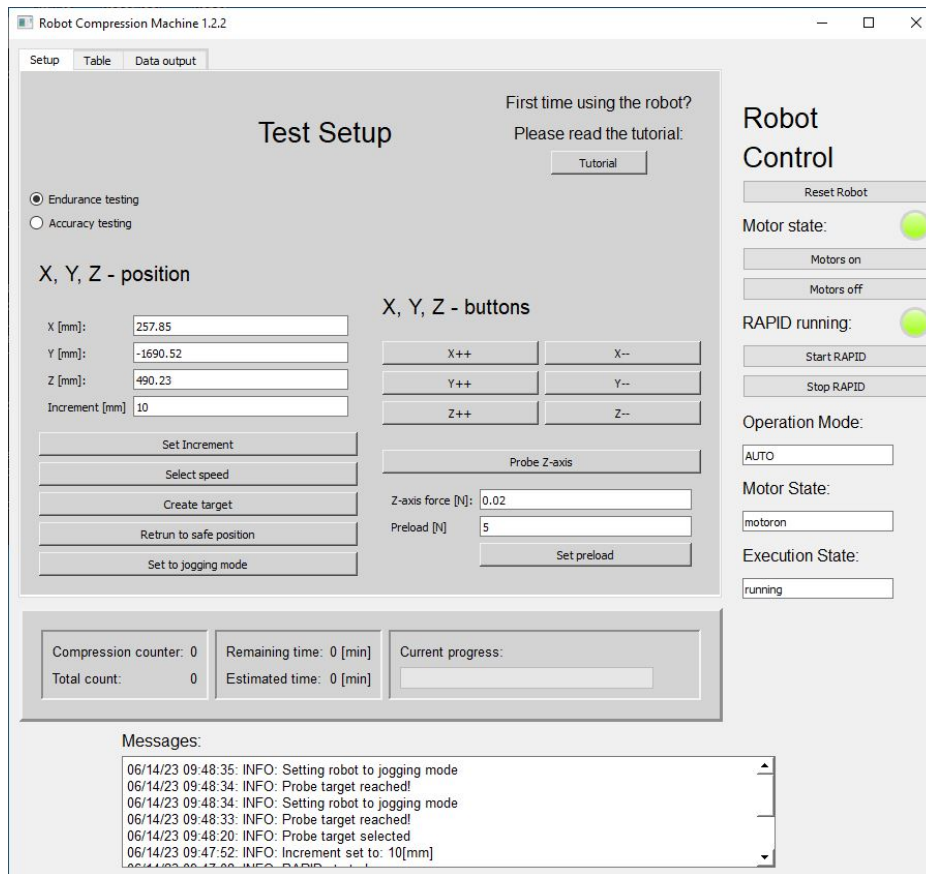
**Figure 3.2:** The steel plate marks the designated test area, the two crosses mark pre-defined targets for compression. The user can choose to assign up to 6 targets anywhere in the test area when using the compression test application.

### 3.2.1 Program structure

The GUI consists of three main tabs, a message console, and a robot control panel, controlling basic robot operations such as turning motors on or off, resetting the program pointer and starting or stopping RAPID execution. The main tabs are for performing test setup and enable the user to write simple scripts for test execution.

The *Setup*-tab provides the user with jogging buttons, to move the robot linearly to the position of a test object. The user can then chose to adjust the robot TCP to the exact location of a compression zero-point. This means that when the robot performs a compression, it will start at the assigned point and move downwards to the desired compression depth. Alternatively the user can utilize a *probe*-function, that will move slowly downwards and set the compression zero-point at a force value assigned by the user. The program currently supports the assignment of six independent compression targets.

## 3.2 Robot Compression Testing Application



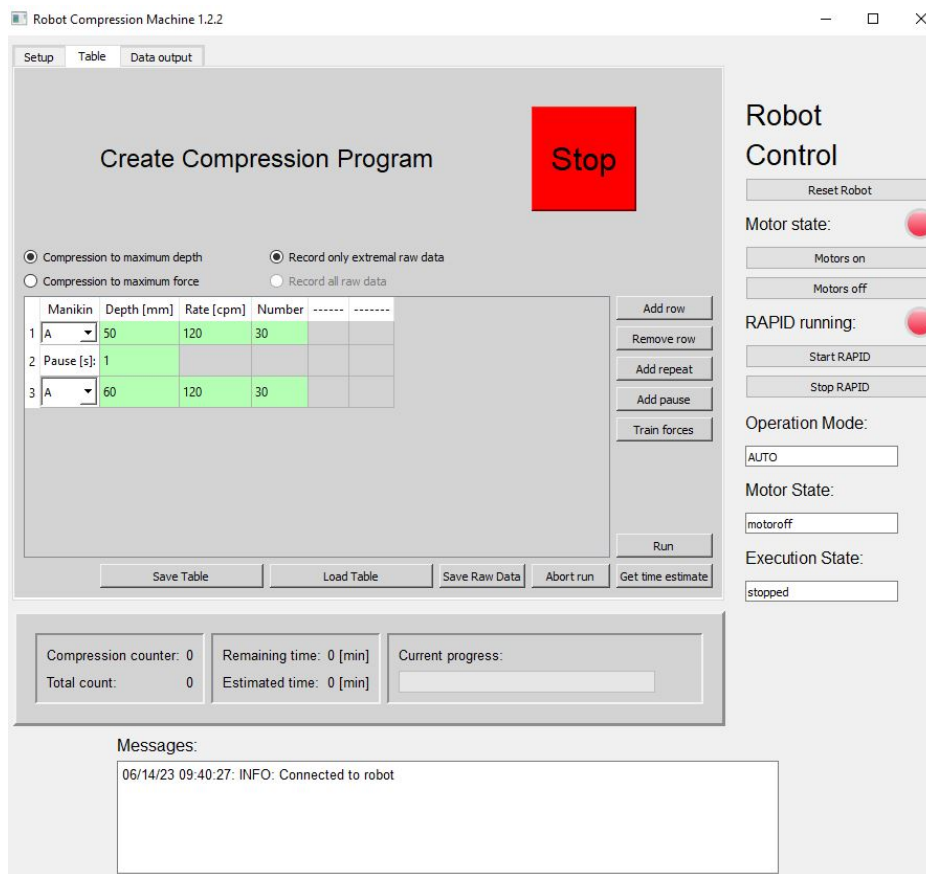
**Figure 3.3:** Compression test application setup-tab. The tab allows users to configure the setup for a test.

Once the user has performed the required setup, the *table*-tab is used to construct the test procedure. Here, the user can specify compression depth, rate and number of compressions to be performed on the targets they have previously defined. Also, the possibility of loops and pauses are available. The functionality enables the creation of a highly customizable compression test procedure according to user needs.

The user is also given the opportunity to save a procedure for later use, and to save a raw-data file after the test has been performed. The raw-data in this file, specifically the force, is what is sought to be improved in this work. This data is used for reviewing product performance at intervals across the



### 3.2 Robot Compression Testing Application



**Figure 3.4:** Compression test application table-tab. Allows users to configure the test procedure.

simulated lifetime. When deemed accurate enough, it can also be used for product specific sensor calibration.

Both the graphical and functional core of the application was written in python [27]. For the graphical interface design the QT library [28] was used, specifically the framework *pyqt5*. For the backend and communication a selection of protocols and third party packages have also been used. The most relevant of these are discussed later in this section.

## 3.3 Load Compensation Application

The load compensation application was created as a separate program, intended to run in the background during a test. After the test is complete, a file is generated for the user to name and store where desired. Originally the intention was to implement the load compensation application in *python*, to minimize the complexity of the project and to ease the integration into the main compression application. Due to issues with protocol compatibility which will be explained more in depth later in this section, it was decided to implement the compensation scheme in *C#* [29].

The core method of the load compensation application consist of three threads which separately communicate with their respective systems in order to gather data when instructed to do so. The threads and their function is listed below:

- *EGM Sensor* thread: Receives position data from the robot controller through ABB's Externally Guided Motion (EGM) protocol.
- *RunCheck* thread: Monitors the state of the RAPID program running through the main compression application. If a test is started, the thread instructs the compensation application to start collecting data.
- *NetFT* thread: Receives force data directly from the load cell through Raw Data Transfer (RDT).

The *RunCheck* thread is based on a Robot Web Services (RWS) client which communicates with the robot using Transmission Control Protocol (TCP) messages. The client monitors a boolean variable in the RAPID module which is controlled by the compression testing application. When a test is running this variable is set to *TRUE* and the data sampling in the load compensation application is started.

When the data sampling is started, the acceleration estimator and compensation scheme is run in real-time. When a test is finished, a text file containing time, position, force, acceleration estimates, and compensated force data is generated.

### 3.3 Load Compensation Application

---

#### 3.3.1 EGM Implementation

Position data of the robot TCP is extracted from the robot controller using the EGM sensor protocol from ABB [30]. The protocol allows access to low level data such as the TCP-position (not to be confused with Transmission Control Protocol), and is explained further in section 3.4.3. Previous data-collection implementations were based on reading data directly from RAPID. This method was determined to be neither adequately efficient, nor able to provide a consistent sample interval.

The choice of implementing the compensation scheme in *C#* was largely based on the EGM protocol's support for this language, combined with its lack of support for other languages. EGM uses Google Protocol Buffers (protobuf) to serialize structured data for transfer. This protocol involves the language specific compilation of a file called a *proto*-file. It was originally attempted using *python*, but the *proto*-file did not compile correctly, even though the Protobuf documentation [31] states that *python* is a supported language. The *proto*-file provided by ABB required some minor syntax modifications in order to successfully compile into a *C#* namespace. The modification was necessary to convert the file from *proto2* (earlier version) to *proto3* (current version) as the former is no longer supported. The resulting namespace is used to communicate using the EGM protocol. The communication is possible due to Protobuf wire-independence, meaning that one communication endpoint can use one version of *proto*-files while the other endpoint can use another version. Protocol Buffers are further explained in 3.5.1.

#### 3.3.2 Net F/T Implementation

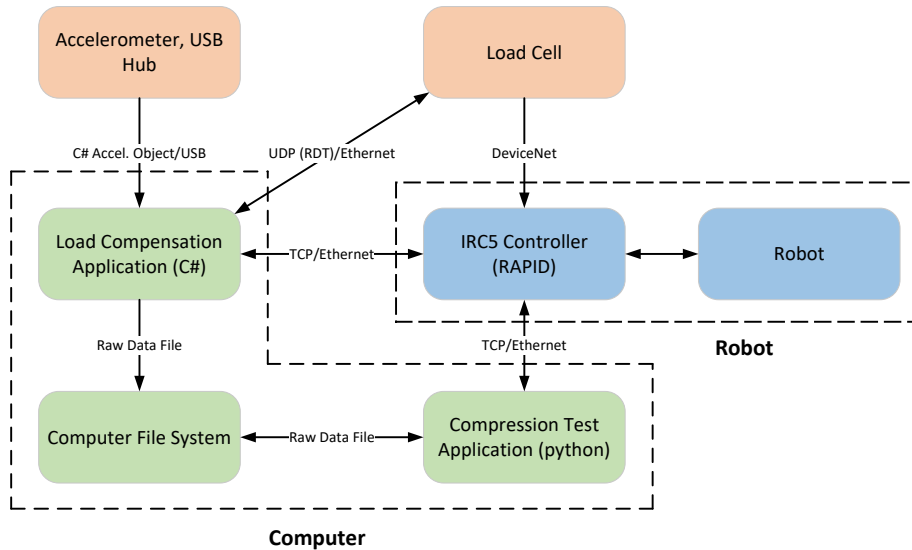
To acquire force data from the load cell, it was first attempted to retrieve the data through Robot Web Services (RWS) Subscriptions using a websocket implementation. This solution was deemed not satisfactory due to lack of speed and consistency of data retrieval. Meaning that the value did not update quickly enough and with an inconsistent rate. This is also described further in appendix A.2.

It was therefore decided to make an implementation where the computer

### 3.3 Load Compensation Application

would communicate directly with the ATI NetBox, which is the load cell's communication interface. This would evade any latency and unreliability introduced by the robot controller, as well as give a large performance improvement in speed. The NetBox is capable of streaming force/torque signals on Ethernet at a rate of 7 [kHz]. However the chosen sample rate was 1000 [Hz] as to not unnecessarily overcrowd the communication pipeline with data. This is important as all communication between the robot and computer applications are sent over the same physical layer. Also, the rate is more than high enough to work with the EGM signal which operates at 250 [Hz].

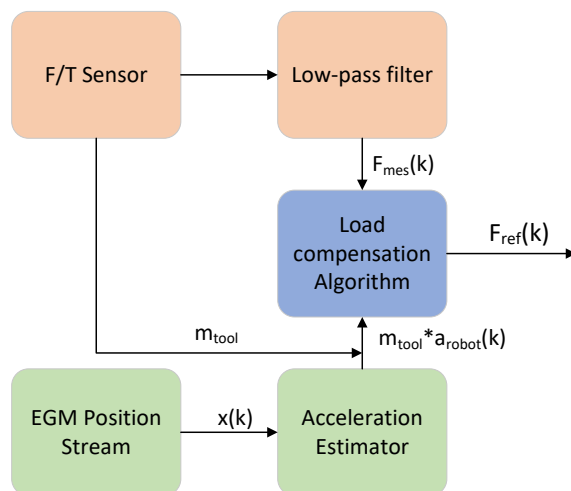
Data packets from the load cell are sent by "Raw Data Transfer" meaning that the NetBox sends data packets containing *counts* using User Datagram Protocol (UDP). These *count* values can then be transformed into the force values upon arrival. The UDP transmission protocol provides favourable speed and efficiency compared to TCP which the previous websocket implementation was based on. UDP is further explained in section A.1. A full overview of all systems and communication pathways can be seen in figure 3.5, while the compensation scheme implementation is shown in figure 3.6.



**Figure 3.5:** Overview of system setup and communication. See main text for explanation.

### 3.4 ABB Resources

---



**Figure 3.6:** Compensation scheme implementation overview. The data from the load cell is passed through a low-pass filter from the NetBox.

## 3.4 ABB Resources

Since this work is based on work done with an industrial robot manipulator produced by ABB, it can be useful to have some background knowledge of the systems used as part of the implementation. The following section provides a brief explanation of the programming language RAPID, the RWS application programming interface (API), and the EGM sensor protocol, all used as part of this work.

### 3.4.1 RAPID

RAPID is a high-level programming language for programming and control of ABB Robots [32]. It supports a leveled programming concept [33], where new routines (similar to functions or methods in other languages), and data types can be installed for a specific robot system. This concept makes it possible to customize the programming environment.

RAPID includes a number of features often found in other high-level programming languages such as modules, procedures, type definitions, arith-

### 3.4 ABB Resources

---

metic and logical operators, control structures, error recovery, and interrupt handling [33].

A program is typically created on a computer through *RobotStudio* or on the *FlexPendant*, and loaded into the robot controller. There is also functionality to save and load programs from the controller memory.

*RobotStudio* is ABB's programming and simulation tool for robotic applications [34]. The *FlexPendant* is a handheld computer connected and integrated with the robot IRC5 controller. It can be used to create, load and save programs, jog the robot, perform calibration and much more.

#### 3.4.2 Robot Web Services

RWS is a platform that enables developers to create their own custom applications to interact with the robot controller.

RWS is designed after the network application architectural style of Representational State Transfer (REST) API, which in turn leverages the HTTP protocol. RWS messages are composed of XHTML and JSON. Robot Web Services facilitates platform independent and language independent communication with the robot controller [35].

In REST, a URL identifies a resource. The representation of application data sent from the robot controller can be either in XHTML or JSON format. Formatting information on how the data should be displayed in a web browser, is not provided through the Robot Web Services. A URL can contain query parameters which are identified with the character '?' [35].

The platform enables communication with applications on a computer and has been extensively employed as part of this work. Both the python compression test application and the C# load compensation application, communicate with the robot controller using Robot Web Services.

## 3.5 Robot/Sensor-Computer communication protocols

---

### 3.4.3 Externally Guided Motion

The EGM sensor protocol is designed for high speed communication between a robot controller and a communication endpoint with minimum overhead. The EGM sensor protocol uses Google Protocol Buffers for encoding and UDP as transport-layer protocol [30].

According to the documentation [30], Google Protocol Buffers was selected due to its speed and language-neutrality. UDP was chosen as the transport protocol since the data is sent in real-time, with high frequency, and because if data is lost it is anyway useless to re-send.

Specifically, this work employs the *Position Stream* functionality of the protocol. When instructed by RAPID, *Position Stream* sends data packets containing planned and actual robot positions at 250 [Hz] until instructed by RAPID to stop.

The EGM sensor protocol data structures are defined by the EGM proto file. Sensor name, IP-address and port number of sensors are configured in the system parameters of the controller.

## 3.5 Robot/Sensor-Computer communication protocols

Several communication protocols are in this work employed to broadcast and receive messages, data and instructions from robot-controller to computer applications. The protocols used have different strengths and are thus used in order to achieve adequate performance based on the application, and the available means. Also the choices are limited due to the compatibility of the robot controller communication which ultimately are ABB developer decisions. This section provides a description of the protocols that have been used and what advantages they offer.

The inclusion of this explanation is based on the fact that the respective protocols mentioned are quite central as part of the application development of this work. Some design choices, such as programming languages and

## 3.5 Robot/Sensor-Computer communication protocols

---

implementation methods, have been directly related to their compatibility with these protocols.

### 3.5.1 Google Protocol Buffers

Google Protocol Buffers (Protobuf) provide a language-neutral, platform-neutral mechanism for serializing structured data in a forward- and backward-compatible way. Similar to JSON or XML, however the data structures are typically much smaller and contain minimal amounts of overhead, significantly improving speed. The data structure is user defined, and interpreted using purpose built code. Meaning that some development time is sacrificed in order to achieve much faster data transfer than what would be possible with a standard form [31].

Developed by Google, Protobuf is their most commonly-used data format for inter-server communication and archival storage. Protocol buffer messages are described by specially authored *.proto* files.

Advantages of using protocol buffers:

- Compact data storage
- Fast parsing
- Availability in many programming languages
- Optimized functionality through automatically-generated classes

This protocol have been used to receive and parse data from the EGM protocol, as part of the implementation of the *Position Stream* functionality. As it is of interest to receive these messages as fast as possible and there being no need to provide a response for each message, in addition to the customizable serialization, this protocol provides better performance compared to standard TCP communication using XML/JSON.

The protocols UDP, TCP and HTTP which are familiar terms are explained in appendix A.



## 3.6 External equipment

---

### 3.6 External equipment

#### 3.6.1 Load cell

The load cell is used to measure the force exerted on a test object during a test. In addition to being used for data analysis the force measurement is also used as a safety/stop switch which will halt a test if a force/torque threshold is breached. This is an additional safety barrier intended to protect the robot, test object and other instruments/objects in close vicinity to where the test is being performed.

The load cell mounted on the robot is a Schunk Delta FTN sensor. The complete system shown in figure 3.7 consist of a sensor/transducer, transducer cable, and Net Box. The Net Box (Named NETB Box in figure 3.7) processes and communicates the transducer readings to the robot controller and computer. The Net F/T sensor system is a multi-axis force and torque sensor system that simultaneously measures forces  $F_x$ ,  $F_y$ ,  $F_z$  and torques  $T_x$ ,  $T_y$ , and  $T_z$ . The system provides Ethernet/IP, CAN bus and Ethernet communication interfaces and is compatible with DeviceNet. DeviceNet is used to transmit data to the robot controller I/O-system. To communicate with the computer, the current setup employs an Ethernet/IP configuration [36].

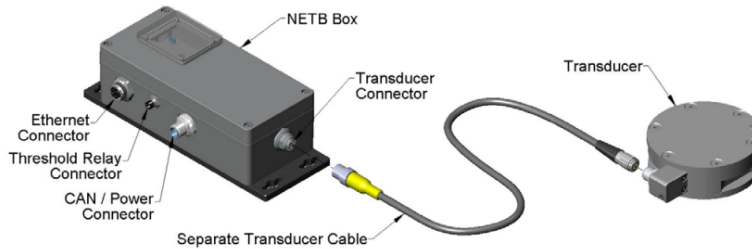
The load has a measurement range of 0 - 200 [kg] at its current calibration. According to the calibration certificate (see appendix B) the load cell has an error of 0.10 [N] when 1112.055 [N] compression force was applied. The load cell is calibrated statically, but the intended use for compression testing is a dynamic environment. The dynamic performance of the sensor is not guaranteed, and the effect on results is discussed in chapter 6.

#### 3.6.2 Interferometer

In order to determine true compression depth, not influenced by robot compliance error, an external position measuring device is employed. This measuring device is an IDS 3010 Displacement Measuring Interferometer System. This system is specifically a low-finesse fiber-based Fabry-Perot

### 3.6 External equipment

---



**Figure 3.7:** Net F/T System

Interferometer. It provides the advantage of an electronic-free sensor head, which is ultra-compact and allows for flexible alignment and multiaxis measurements [37].

The system in use, includes two sensor heads of type M12/C7.6 with fiber-cables, the main unit, and an Environmental Compensation Unit (ECU). The sensor heads are placed on either side of a boom which is part of the compression tool assembly. This can be seen in figure 3.8. The sensor head requires reflectors to register signals, which are placed on the floor below the test assembly. Currently the measurement data is not integrated with the other measurement devices, due to this device not being intended for use during conventional product testing. The standalone application for recording data has thus been used for tests done in chapter 5.

According to its calibration certificate the interferometer has a measurement accuracy of  $\pm 1$  [ppm] for distances up to 5 [m] (see appendix B). Maximum update frequency is 500 [kHz].

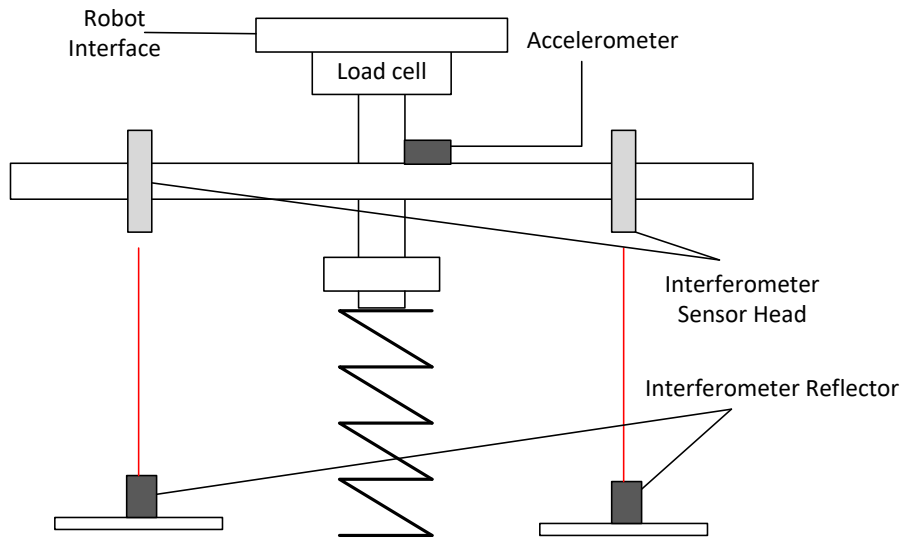
#### 3.6.3 Accelerometer

In order to evaluate the accuracy of the acceleration estimators, the estimator data is compared with the data from an accelerometer. The accelerometer is of the type Phidgets MOT1100\_0. It is capable of measuring accelerations of up to 8 [g] in three-axes, with a maximum frequency of 100 [Hz] [38]. Although this frequency is not as high as the estimator update frequency (250 [Hz]), it is still useful for comparison of the data. The accelerometer is mounted onto the compression tool (see figure 3.8) and connected to a

### 3.6 External equipment

---

Phidgets VINT-hub which connects by USB-cable to the robot computer.



**Figure 3.8:** Overview of external equipment for testing.

#### 3.6.4 Compression spring

When performing tests that require resistance simulating a test object, a linear spring is used. Although its stiffness is typically a little harder than most test objects, it is preferred due to its linearity. Springs are typically used when performing equipment calibration or measurements requiring high accuracy and consistency, including several of the tests performed as part of this thesis.

For testing, two springs with different spring rates have been used. They can be seen in figure 3.9. The softest spring is named TA48 (brushed steel) and the stiffest spring is named TA22 (black). They do not have calibration certificates, so their spring rate is determined experimentally in chapter 5.

### 3.6 External equipment

---



**Figure 3.9:** Springs used for compression testing.

#### 3.6.5 Scale AH46

The scale was used as an external reference in the static measurement test in section 5.4. As it has small measurement error and uncertainty, it helps to minimize the error and uncertainty in the spring rate estimates. According to the calibration certificate (see appendix B), it has a measurement area of 0 - 150 [kg] and a measurement error of -0.043 [kg] for the maximum load. The spring can be seen in figure 5.7.

# Chapter 4

## Simulation

To obtain insight in their performance and accuracy, the estimation methods described in section 2 will be tested on a set of artificial signals that represent a selection of test cases for the robot. These cases will be the same as in section 5.5. The test cases are selected with the intention of evaluating the performance in typical testing conditions as well as edge cases, for example using high compression rate and large depth. A time analysis is performed in order to establish how computationally efficient the methods are, compared to each other. It is expected that the simpler methods, especially the Differentiation method should have the best time efficiency.

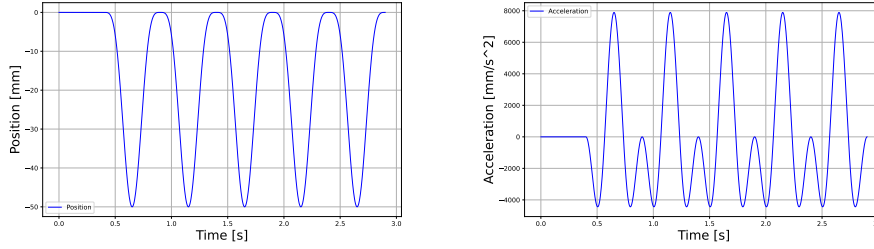
### 4.1 Generation of compression signal

The signals are generated by an algorithm based on the same theoretical compression function that the RAPID module used as part of the compression test application is based on. This ensures that the artificial signals are as close to the real signals as possible. However it should be mentioned that the robot does not perfectly produce these signals during testing. The function is a result of internal research at Laerdal with the aim of discovering a repeatable and accurate function to describe the motion profile of a heart compression. The advantage of using the theoretical function is that the respective derivatives that describe velocity and acceleration can

## 4.1 Generation of compression signal

---

be easily found by analytical differentiation of the original position function. To evaluate the estimators, these differentiated signals are considered the true values that will serve as the reference point for estimation accuracy. In figure 4.1, the motion profile for position and acceleration are shown for 5 compressions with depth 50 [mm] and rate of 120 [cpm]:



**Figure 4.1:** Position and Acceleration profile for the theoretical function

The output signals seen in figure 4.1 are based on the following functions in continuous time:

$$x_{sim}(t) = -\frac{d}{4}(1 - \cos(2\pi ft))^2 \quad (4.1)$$

$$a_{sim}(t) = 2df^2\pi^2(-\cos(2\pi ft) + \cos^2(2\pi ft) - \sin^2(2\pi ft)) \quad (4.2)$$

Where  $x_{sim}(t)$  and  $a_{sim}(t)$  represents position and acceleration respectively, while  $d$  represents compression depth and  $f$  (the compression frequency) is given as:

$$f = \frac{\text{compressions per minute [cpm]}}{60} \quad (4.3)$$

### 4.1.1 Noise

The signals obtained from experimental testing will have a certain level of noise. Although the level of noise is unknown, previous work with the robot's position data indicate a relatively low noise level. Due to this fact, multiple simulations will be done using noise levels with signal to noise ratio

## 4.2 Simulation Data

---

(SNR) values of 100, 80 and 60 [dB]. Although the simulated signals has a high SNR, it is expected that the results of some estimators will be compromised as noise levels increase. Potentially from the noise amplification which is inherent in differentiation.

Providing a quantitative measure of the performance on the artificial signal will provide an indication of the comparative performance. However, these results only serve as an indication of the performance of the estimators.

## 4.2 Simulation Data

Results presented consist of estimator output, deviation at peak acceleration average and RMSE of the same simulation run using the analytical result as benchmark for both metrics. Each simulation consists of 30 compressions, with a small idle section at the start. The simulation runs consist of combinations of compression depth of 50 and 80 [mm] and compression rates of 80, 100 and 120 [cpm], similarly to the tests performed in section 5. These simulations are also performed at three levels of SNR as described in the paragraph above.

Compression depth of 50 [mm] is chosen as it is the recommended target depth for chest compressions according to the ERC recommendations [1], which is the most used depth for compression testing and calibration.

In order to evaluate product performance when compressions are performed deeper than recommended, it is sometimes necessary to perform testing at depths of 60 and even 70 [mm]. In order to gain insight in the performance of the estimators in these cases, 80 [mm] compression depth is chosen as this is a definite edge case. If performance in this depth interval is acceptable then it is also safe to assume that it is acceptable within it.

Both the data generation and the estimators have been implemented in *python 3.9.1* for the simulation. See attached zip file or github repository: Robot\_CM/py\_est/Simulation.py.

The following estimator parameters used for the simulation tests in this section and also for the experimental tests in section 5:

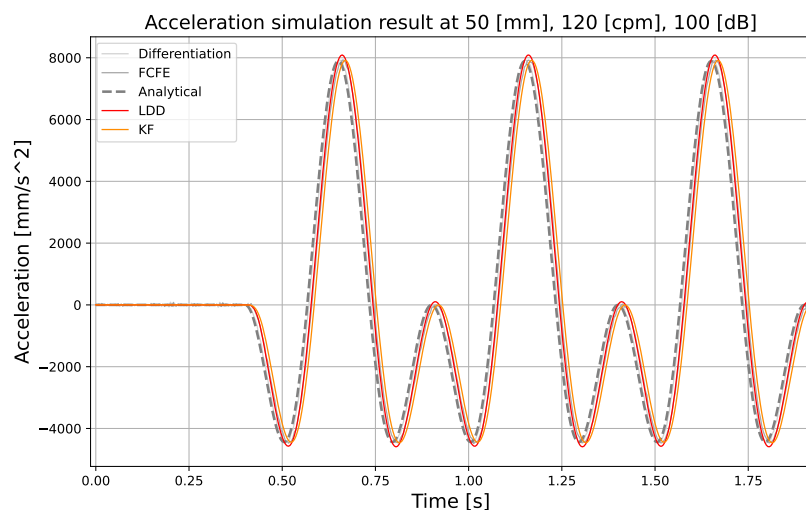
### 4.3 Simulation results

---

- Differentiation: No parameters available.
- LDD:  $\alpha_1 = 3, \alpha_2 = 4, \alpha_3 = 3, R = 75$
- FCFE: 4<sup>th</sup> order
- KF:  $\Delta y_{min} = 0.2$ , all other parameters specified in section 2.

### 4.3 Simulation results

Simulations have been run at the mentioned SNR's in a combination of two compression depths and three compression rates for a total of six tests per SNR. The simulation serves to indicate the expected comparative performance of the estimators, using the analytical benchmark.



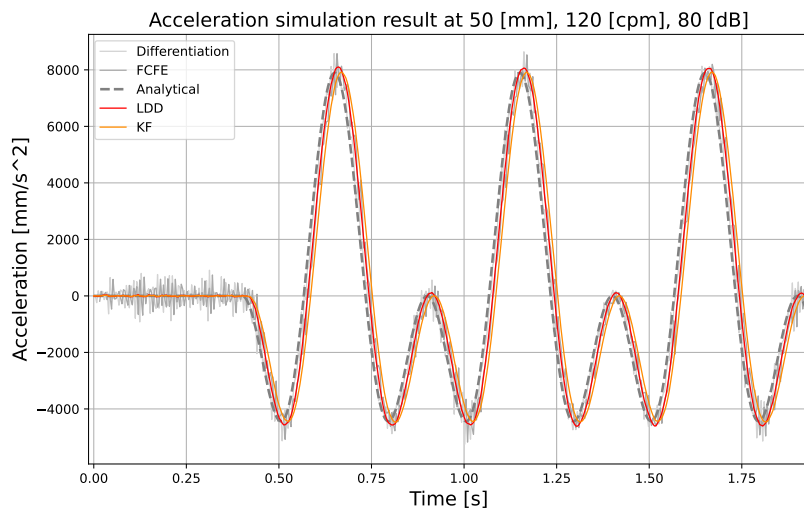
**Figure 4.2:** Result of simulation using 50 [mm], 120 [cpm] and SNR = 100 [dB].

Figures 4.2 - 4.4 shows simulations of compression testing at 50 [mm] and 120 [cpm]. From figure 4.2 it can be seen that all estimators follow the analytical acceleration quite closely. It will be seen however, that small shifts in phase can lead to significant deviation, for example with regards to the KF.



### 4.3 Simulation results

---



**Figure 4.3:** Result of simulation using 50 [mm], 120 [cpm] and SNR = 80 [dB].

Looking at figures 4.3 and 4.4 a deterioration in the Differentiation and FCFE estimates can be observed as the SNR decreases. These two estimators perform poorly especially in the low-speed regions (see figure 4.3). And at an SNR of only 60 these estimates are reduced to a noisy signal which only roughly contain the shape of the acceleration signal (see figure 4.4). Notice also how the LDD and KF retains their shape (see figure 4.4).

Figures 4.5 and 4.6 display a comparison of the performance of each estimator based on the change in SNR. Figure 4.6 highlights the robustness of the LDD and KF estimators, while figure 4.5 highlights the lack thereof for the Differentiation and FCFE estimators.

Note that figures 4.2 to 4.6 shows the first three compressions of each simulation, including a small idle period at the start. Showing only a few compressions is done to provide a better visual illustration and as stated each simulation consisted of a total of 30 compressions.

### 4.3 Simulation results

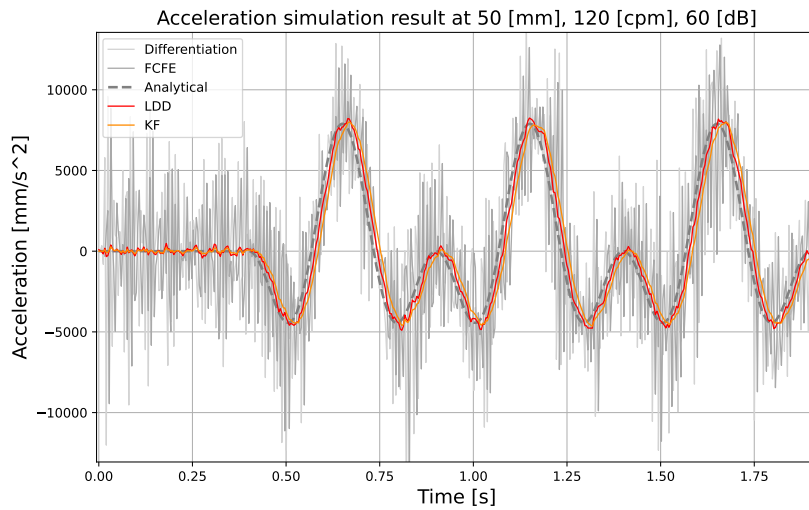


Figure 4.4: Result of simulation using 50 [mm], 120 [cpm] and SNR = 60 [dB].

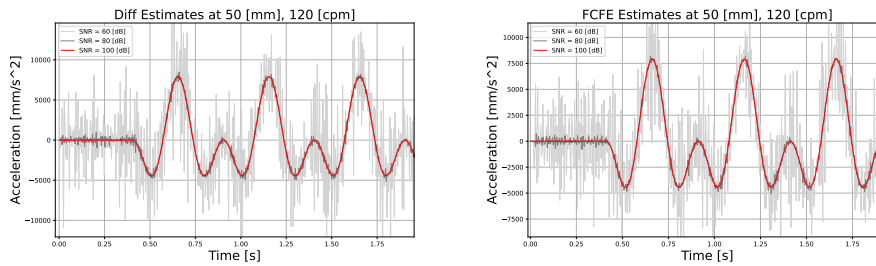


Figure 4.5: Differentiation and FCFE results compared with change in SNR.

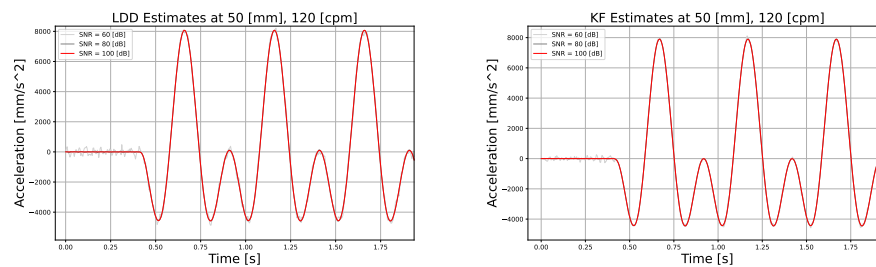


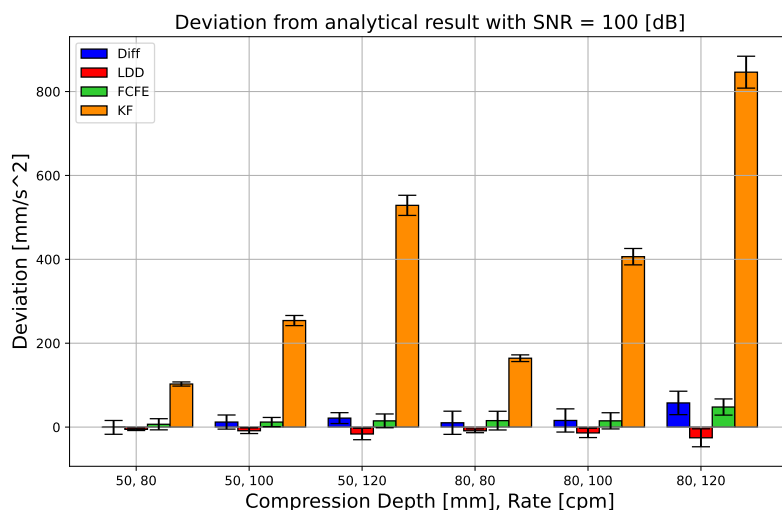
Figure 4.6: LDD and KF results compared with change in SNR.

## 4.3 Simulation results

### 4.3.1 Peak acceleration deviation

The purpose of this simulation is to establish how accurate the estimators are at the acceleration peaks. The reason for evaluating the peak acceleration is because of its relation to the peak force which will be evaluated in section 5.

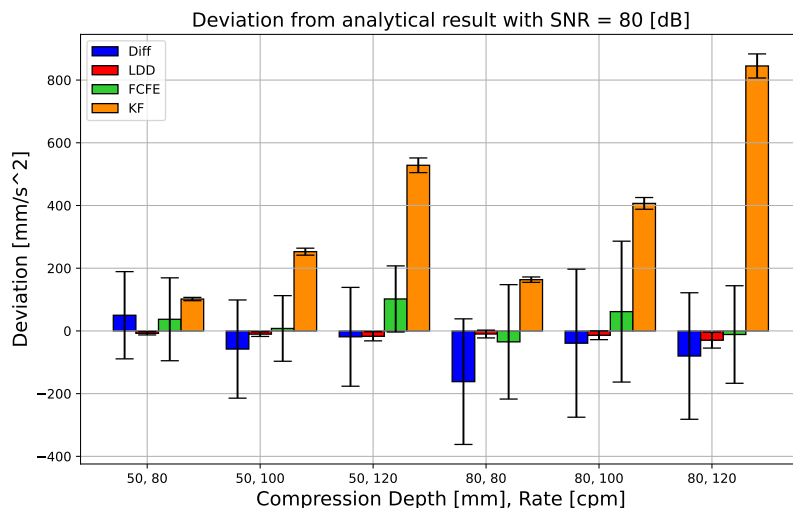
The resulting average deviations of the simulations for all noise levels are shown in the figures 4.7 - 4.9. Considering first the tests using a SNR equal to 100 [dB] in figure 4.7, it can be seen that the KF has the largest deviation, but it displays a similar uncertainty compared to the other estimators. All the other estimators display quite small deviations.



**Figure 4.7:** Deviation at peak acceleration for all simulations with a SNR = 100 [dB].

Considering the simulations with a SNR equal to 80 [dB] seen in figure 4.8, results are somewhat similar. It should be noted however that both the Differentiation and FCFE methods show a large degree of uncertainty in deviation, indicating that these estimators have a comparatively larger noise sensitivity. However, the KF still shows the largest deviation across estimators.

### 4.3 Simulation results



**Figure 4.8:** Deviation at peak acceleration for all simulations with a SNR = 80 [dB].

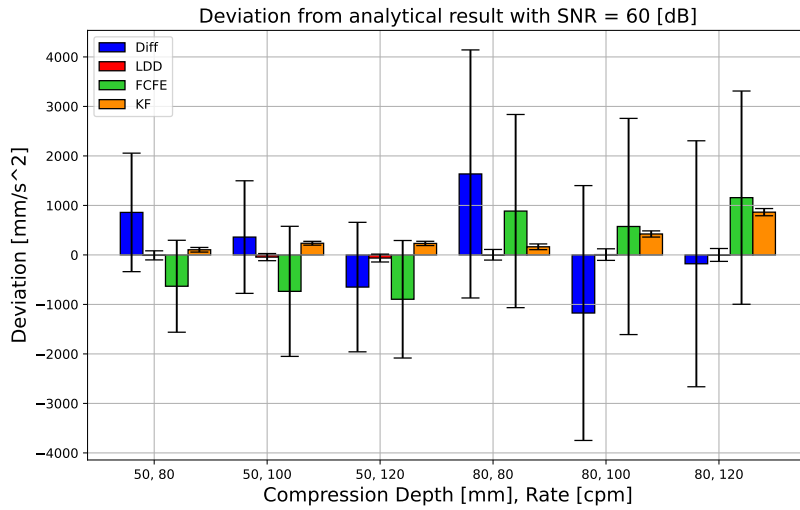
Moving over to the last case, using a SNR equal to 60 [dB] in figure 4.9, it is safe to say that the estimation results for the Differentiation and FCFE have been fully compromised due to the noise level. As mentioned it can be seen in figure 4.4 that the outputs have been reduced to highly fluctuating signals using these estimators. This tendency could also be observed to a degree in the case of SNR equal to 80 [dB]. The KF shows similar accuracy and uncertainty across all noise levels, thus proving its robustness.

Finally, one estimator remains to be discussed: the LDD estimator. Simulation results show that this estimator provides both a consistently small deviation and is also resistant to the noise levels considered in the simulation.

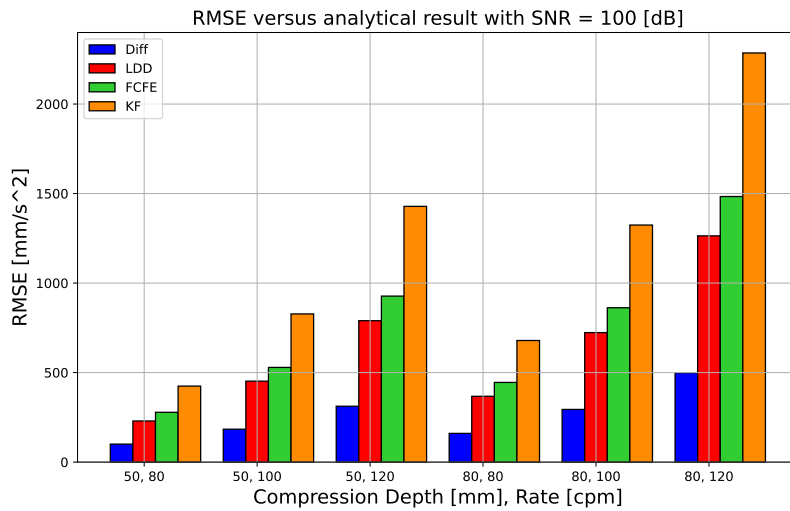
#### 4.3.2 RMSE performance

RMSE captures the overall accuracy of an estimator and provides insight into the accuracy and robustness of the methods. Figures 4.10 - 4.12 show the RMSE results for the same tests as shown in the previous section.

### 4.3 Simulation results



**Figure 4.9:** Deviation at peak acceleration for all simulations with a SNR = 60 [dB].

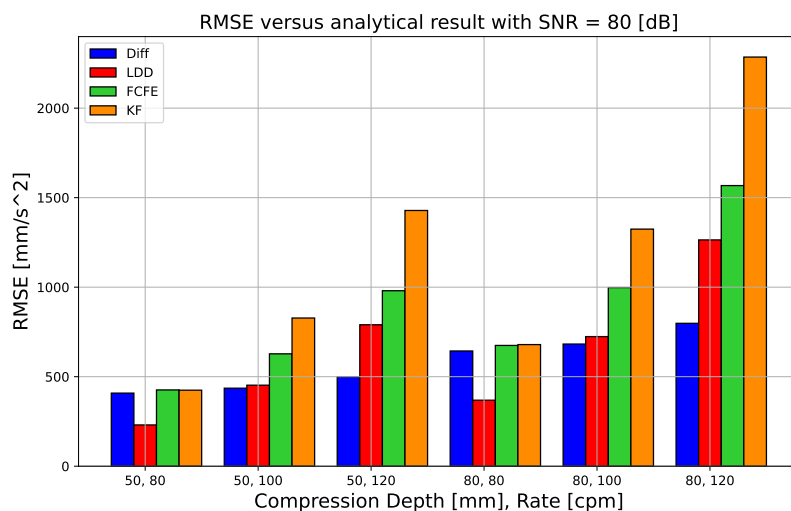


**Figure 4.10:** RMSE performance with a SNR = 100 [dB].

Looking at figure 4.10 it can be seen, similarly to the deviation results, that the KF has the largest RMSE, while the Differentiation estimator shows the

### 4.3 Simulation results

lowest RMSE. The development at a SNR equal to 80 [dB] in figure 4.11, also looks similar, with the performance of the Differentiation showing the most decline in accuracy, while the other methods remain quite similar.



**Figure 4.11:** RMSE performance with a SNR = 80 [dB].

Again, as shown in figure 4.12 the results for the Differentiation and FCFE estimators are severely compromised at a SNR equal to 60 [dB]. With regards to the LDD and KF, their results remain largely unaffected by the change in SNR, and again the LDD shows the best accuracy of the two.

Considering the performance of the estimators based on visual analysis and quantitative evaluation, the following insight is gained:

- Differentiation: Very effective and accurate at the highest SNR, however the results are quickly compromised as SNR is lowered.
- LDD: Consistently displays accurate and robust performance across SNR values.
- FCFE: Also displays good accuracy at the highest SNR value, but results are compromised as SNR decrease.
- KF: Shows consistent but less accurate results, displaying its robustness as SNR decreases, appearing unaffected by the changes in SNR.

### 4.3 Simulation results

---

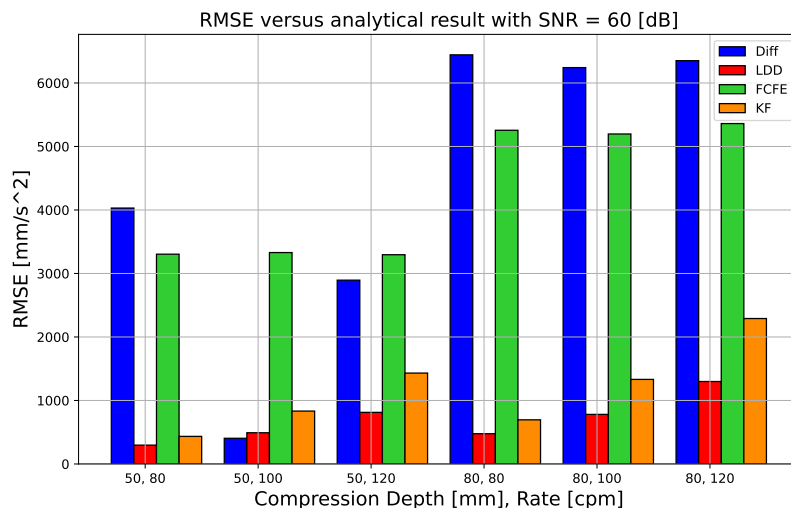


Figure 4.12: RMSE performance with a SNR = 60 [dB].

#### 4.3.3 Time performance

It is a goal for the estimators and the compensation algorithm that all the calculations can be handled in between sampling periods, thus making the implementation real time. This term should, however, be used carefully as the hardware used does not give any guarantee of true real time performance.

For this evaluation, the python module *timeit* was used. All estimators have separate methods for updating the estimates. Each of these methods was run 100 000 times per estimator, and the minimum calculation time found by using the python function *min()* on the result vector. The resulting value was used as the true value in accordance with the documentation for the *timeit* module. It states that higher values are typically not caused by variability in *python's* speed but other processes interfering with the timing measurement [39], i.e. because of the CPU schedule of the system.

The simulation was run on a computer using Windows 10.0.19045 with an Intel i5-8300H quad core CPU at 2.30 [GHz], with 8 [GB] RAM and 19.3 [GB] available memory.

### 4.3 Simulation results

---

Time performance				
-	Diff	LDD	FCFE	KF
$t[\mu s]$	0.199	0.899	36.700	91.799

**Table 4.1:** Calculation time per time-step update of each estimator with parameters used in simulation tests performed in the same *python* environment.

As table 4.1 shows, the time-step calculation for every estimation update method is well below the sampling period  $T_s$  of 4 [ms]. Note that the calculation time is drastically higher for the FCFE and KF methods, while the Differentiation method is the fastest as expected.

These results do not take other factors into account such as data transfer, variable updates and force compensation that occur outside of the estimation method. However this proves that the estimation methods themselves are not an obstacle for a real time implementation.



# Chapter 5

## Results

In this section, results from experiments on the robot are presented. The first part will focus on the performance of the acceleration estimators, followed by an evaluation of the compensation scheme without spring load, and finally an evaluation of performance in a typical test scenario. The results from this test will be considered as the practical accuracy improvement that the compensation scheme provides. The respective sections will include comments on the comparative performance of the estimators, however more comprehensive discussion of the results are kept to chapter 6.

The following experiments have been conducted:

- **Estimator-accelerometer comparison:** To evaluate the performance of the acceleration estimators separately from the compensation scheme.
- **Compression test w/o resistance:** This experiment is done in order to evaluate the compensation scheme to a reference without the dependency of a compression spring.
- **Static measurement test:** In order identify the static accuracy of the load cell and to determine the spring rate of the springs used for the following experiment.
- **Spring compression test:** This experiment is conducted to obtain

## 5.1 Test procedure

---

a measure of the performance of the compensation scheme as it will be used in a testing environment.

### 5.1 Test procedure

Expect for the static measurement done in section 5.4, all tests have been performed within the following framework: Each test consist of the robot performing a total of 60 compressions at the desired depth and rate. Plots showed in this chapter will be of the three first compressions in order to provide illustrations of the data. As the implemented control scheme uses a simple proportional error regulator implemented in RAPID, the robot performs around 15 to 20 compressions before it is considered to be performing compressions at a consistent and accurate depth according to its internal measurements. Thus, the population that serves as the basis for statistical results consist of the last thirty compressions of each experiment.

### 5.2 Estimate-accelerometer comparison

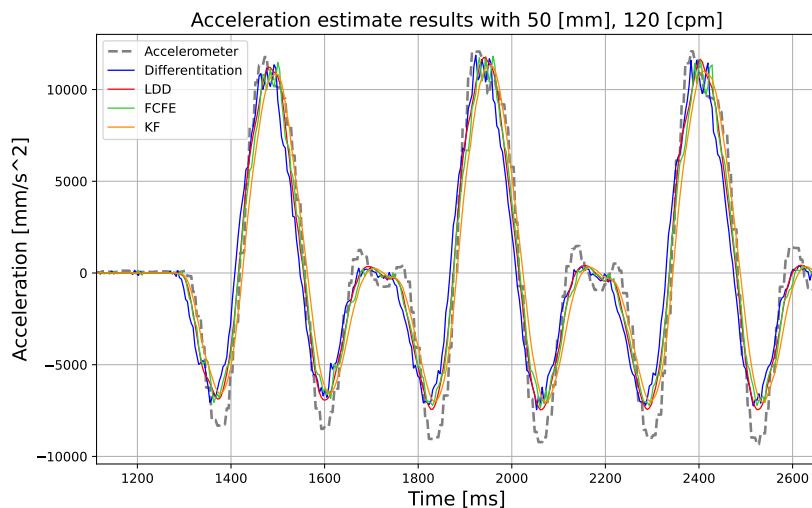
The first experiment is an evaluation of the acceleration estimate on experimental position data. This way the accuracy of the estimate can be evaluated without any disturbance from the load measurement equipment. For this evaluation the estimators are compared to the data output from the accelerometer presented in section 3.6.3. Note that the sample rate of the accelerometer is limited to 100 [Hz], which is not as high as the estimator frequency. However, it still provides an adequate reference from which to evaluate the performance.

The quantitative evaluation of performance as explained in section 2 is based on deviation measurements based on the range of compression depths and rates, as specified in section 4.2. The experiments were performed under the condition that the robot was performing compressions without spring resistance. The same instances of experimental data are further reviewed in the next section.

Figure 5.1 shows that all estimators provide a reasonable estimate of the

## 5.2 Estimate-accelerometer comparison

---

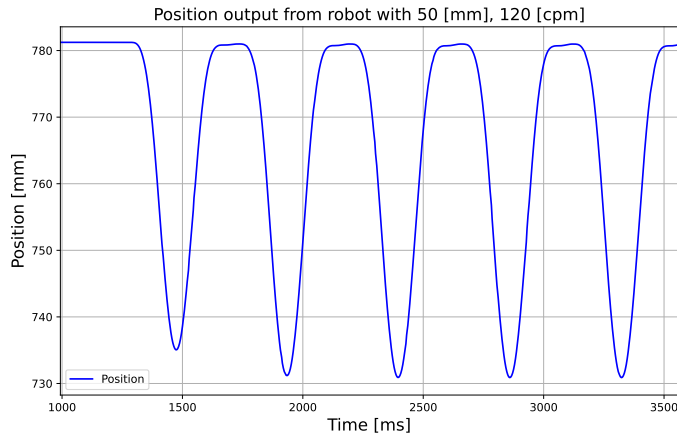


**Figure 5.1:** Acceleration estimates compared to accelerometer data for the first three compressions of 50 [mm] depth and 120 [cpm] compression rate.

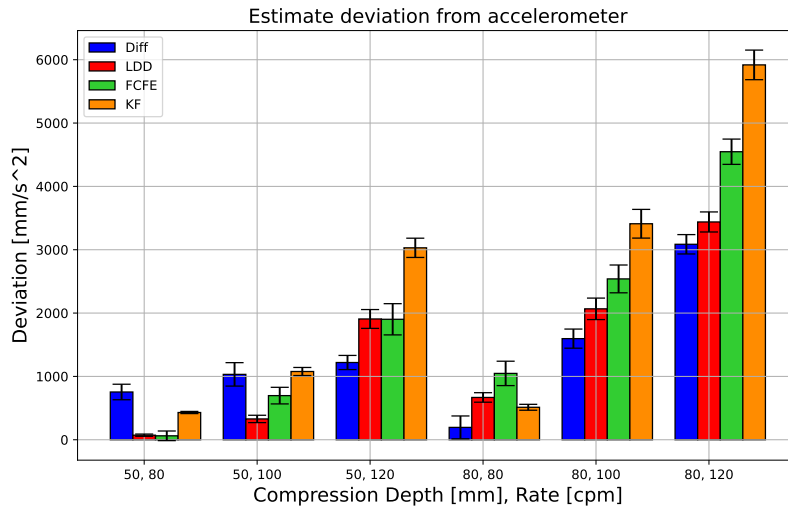
measured acceleration. Compared to the figures of the simulation using the same parameters (figures 4.2 to 4.4) it can be seen that the peak acceleration values are higher, however this is likely due to the slightly different position profile produced by the robot as seen in figure 5.2. This is further discussed in chapter 6. Also, based on the visual evaluation of the output the SNR of the signal output from the robot is in the area between 100 and 80 [dB]. It can also be seen that all estimators provide a large deviation at the negative acceleration peaks. This is also the case around the zero acceleration where the accelerometer shows slight oscillations. However, these peaks are not of large interest for obtaining data and thus not a large concern with regards to the performance of the compensation scheme.

Figure 5.3 shows the acceleration deviations using the accelerometer as reference. This plot shows some similarities to the data presented in section 4.3.1, however the deviation is generally of a larger magnitude, especially when considering the compression parameter conditions at 80[mm] and 120 [cpm]. It should be noted that this data considers the acceleration peak, which is assumed to be the same point as the compression force peak.

## 5.2 Estimate-accelerometer comparison



**Figure 5.2:** Position output of the five first compressions from the robot with 50 [mm], 120 [cpm]. Note that the first compression is slightly more shallow than the subsequent compression. The cause of this is the regulator mentioned in section 5.1.



**Figure 5.3:** Acceleration estimate peak deviation from accelerometer measurement. Caps indicate confidence interval (CI) of 95% from variation in data set.

## 5.3 Compression test without resistance

The aim of this experiment is to evaluate the performance of the compensation scheme when performing compressions while there is no spring resistance on the robot. If the compensation scheme had ideal performance, the resulting output should be a constant 0 [N]. This is not a realistic expectation, but it will clearly demonstrate the effectiveness of the compression algorithm. The relative errors compared to the robot performing compressions on a spring are expected to be similar. In fact, if there is a large deviation it might reveal that there are factors not accounted for.

The resulting deviations are presented as error in [N] from the zero line which is considered optimal compensation performance. The deviation tolerance of the compression machine mentioned in chapter 1 is included to visualize the comparative performance of the compensation scheme. The deviation tolerance is 0.4 [kg] according to the compression machine calibration certificate(see appendix B).

Figure 5.4 illustrates the effect of the compensation scheme for compressions without resistance at 50 [mm] and 120 [cpm]. It can be observed that the deviation from the optimal value (0 [N]) is significantly reduced by the compensation scheme regardless of estimation method.

Tests at 50 [mm] compression depth are shown in figure 5.5. It can be clearly seen that regardless of estimation method, the compensation scheme yields much improved results when compared to the uncompensated output. The LDD appears to have the best accuracy across all compression rates. It shows a small but consistent accuracy advantage over the other estimators.

The same can be seen for compression tests at 80 [mm] in figure 5.6. The LDD again consistently shows the lowest deviation across all compression rates. The FCFE and KF also demonstrate low deviation at times, but not they do not appear as consistent as the LDD. The Differentiation estimate shows increased deviation, but still significantly lower than the uncompensated output.

It can be seen that with the exception of two cases, the compensation scheme cannot produce an error below the specified tolerance of the compression

### 5.3 Compression test without resistance

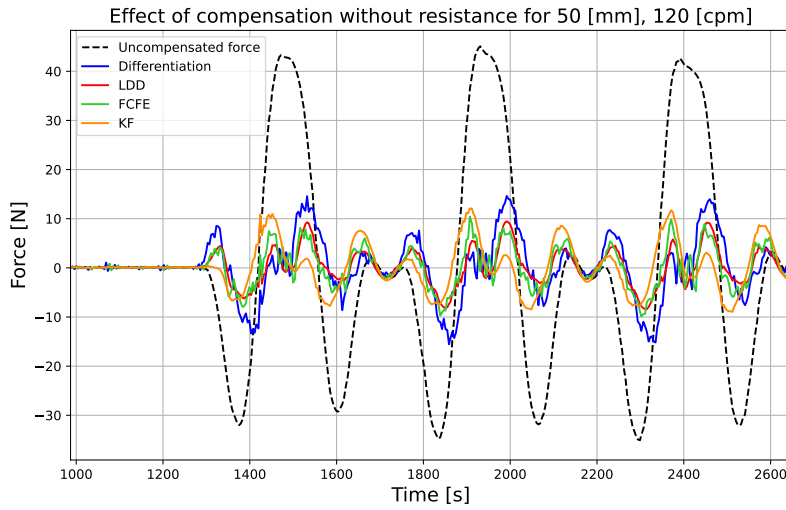


Figure 5.4: Effect of compensation without resistance for 50 [mm], 120 [cpm].

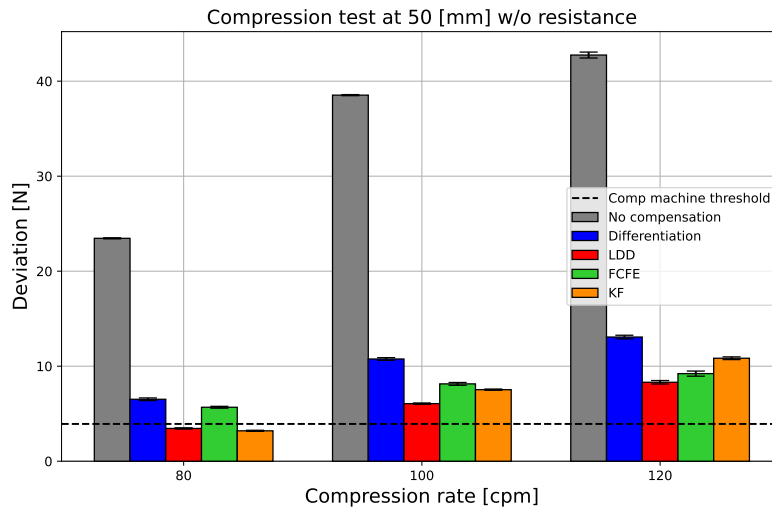


Figure 5.5: Deviation from zero at 50 [mm] without resistance.

machine.

## 5.4 Static measurement for determination of spring rate

---

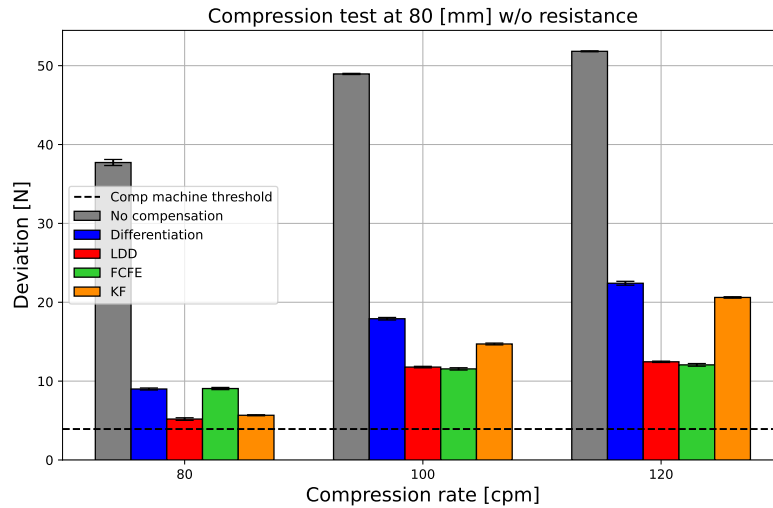


Figure 5.6: Deviation from zero at 80 [mm] without resistance.

## 5.4 Static measurement for determination of spring rate

Before the performance of the compensation scheme can be properly evaluated in a typical test scenario, a spring test under static conditions must be performed in order to determine the spring rate of the linear spring that will be used in the place of test objects. The spring rate will be used to calculate the reference load during the final experiment in the following section. In addition to the load cell the static test will use a calibrated scale which provides an independent frame of reference.

The secondary objective of the static measurement evaluation is to find the deviation of the output from the load cell. Static measurements have been performed at 50 [mm] compression depth. It is assumed that the springs are linear, i.e. that their spring rate is independent of displacement.

The tests have been performed according to the following procedure:

- Interferometer system is connected and calibrated.

## 5.4 Static measurement for determination of spring rate

---

- Compression spring and base is placed on the load surface of the scale.
- The zero point for compressions is set to a  $5[N]$  pre-load, ensuring a point of reference where the robot tool and spring are firmly in contact with each other.
- Interferometer and scale is zeroed, current position and load is read from the robot's teach pendant.
- Robot is manually jogged to the target depth, as accurately as possible.
- All relevant values are read at target depth.
- All steps are repeated a total of 5 times to establish a population of data.

The interferometer described in section 3.6.2 is used to establish a reference for position that is independent of the robot, thus evading the mechanical inaccuracies such as the compliance error of the robot. The interferometer is mounted on a boom which is part of the compression tool mounted to the robot, the lasers are connected to optical cables that provide output to a lab computer. One laser is connected to each side of the point of compression, and the average of these two outputs are read as the true value, as described in chapter 3. The setup for the test can be seen in figure 5.7.

Note also that before the static measurements were performed, the whole tool assembly was dismounted from the load cell's mechanical interface. The load cell was then reset in order to eliminate any bias or creep which might have accumulated over time.

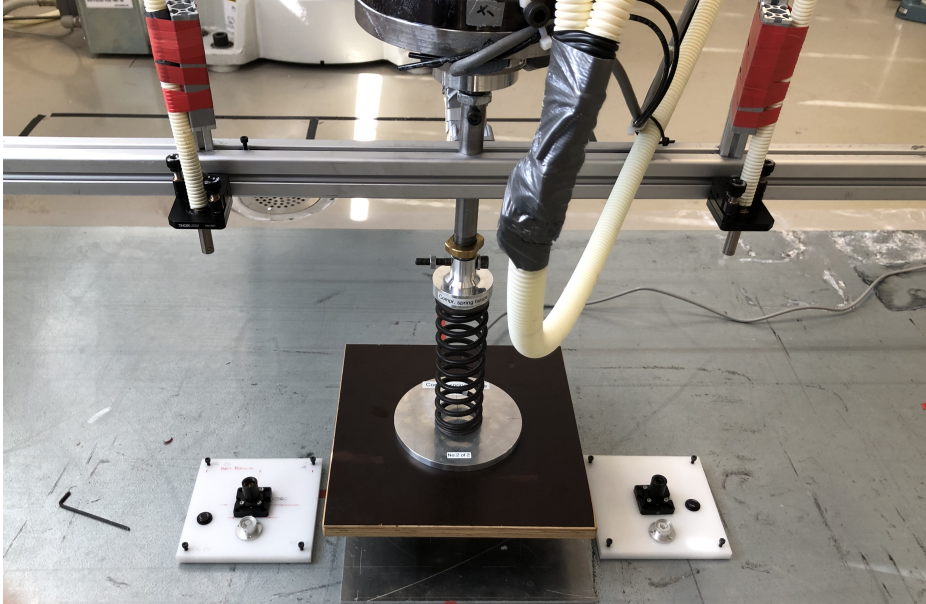
### 5.4.1 Spring TA22

The spring TA 22 is the stiffest spring used for testing. The reason for using a stiff spring, was in order to investigate performance at a higher load than what is typical during normal use.

As seen in tables 5.1 the measurements from the load cell had an average deviation from the scale of  $0.41[kg]$  for both depths. Although not optimal,



## 5.4 Static measurement for determination of spring rate



**Figure 5.7:** Test setup for static measurement with spring TA22. Interferometer sensor heads can be seen fastened on to the aluminium profile boom, with reflectors placed on the floor. Also the spring is set on top on the scale (brown plate) AH46.

Static measurement at 50 [mm] depth, TA22						
Run number	Robot depth [mm]	Interferometer depth [mm]	Deviation [mm]	Scale [kgf]	Load cell [kgf]	Deviation [kgf]
1	49.97	49.132	0.839	73.07	73.47	0.40
2	49.95	49.087	0.863	73.04	73.45	0.41
3	50.07	49.197	0.873	73.22	73.64	0.42
4	49.99	49.141	0.850	73.16	73.56	0.40
5	49.93	49.092	0.839	73.09	73.51	0.42
Mean	49.982	49.130	0.853	73.116	73.53	0.41
Std. dev	0.04833	0.040	0.014	0.06530	0.07	0.01

**Table 5.1:** Static measurement data at 50 [mm] depth using TA 22.

it confirms that the load cell is reasonably accurate and, at least statically, the values from the load cell are trustworthy.

## 5.4 Static measurement for determination of spring rate

It should also be noted that the static error of the depth measured by the robot deviates from the interferometer measurements. The deviation increases with the depth due to compliance error as expected when the robot is put under high loads [25]. This is further discussed in chapter 6.

### 5.4.2 Spring TA48

For compression testing, it was also desirable to use another spring with a lower spring rate, more closely resembling a typical test object. The table 5.2 shows the resulting spring rate for the spring TA 48, which is used to calculate the expected reference for the dynamic tests using this spring.

Static measurement at 50 [mm] depth TA48						
Run number	Robot depth [mm]	Interferometer depth [mm]	Deviation [mm]	Scale [kgf]	Load cell [kgf]	Deviation [kgf]
1	50.00	49.679	0.321	45.48	45.72	0.24
2	50.10	49.726	0.374	45.53	45.80	0.27
3	50.08	49.798	0.282	45.55	45.83	0.28
4	50.06	49.721	0.340	45.53	45.78	0.25
5	50.00	49.626	0.375	45.46	45.73	0.27
Mean	50.05	49.710	0.338	45.51	45.77	0.26
Std. dev	0.04	0.057	0.035	0.0341	0.04	0.02

**Table 5.2:** Static measurement data at 50 [mm] depth using TA48.

Spring rate, $k$		
Spring	Robot	Interferometer
TA22	14.351	14.599
TA48	8.920	8.981

**Table 5.3:** Spring rate,  $k$  for the springs used in compression testing calculated from static tests using scale, robot and interferometer measurements.

Looking at table 5.3 the value of the spring rate  $k$ , that most closely represents the true value, will be the average from the interferometer and scale measurements.

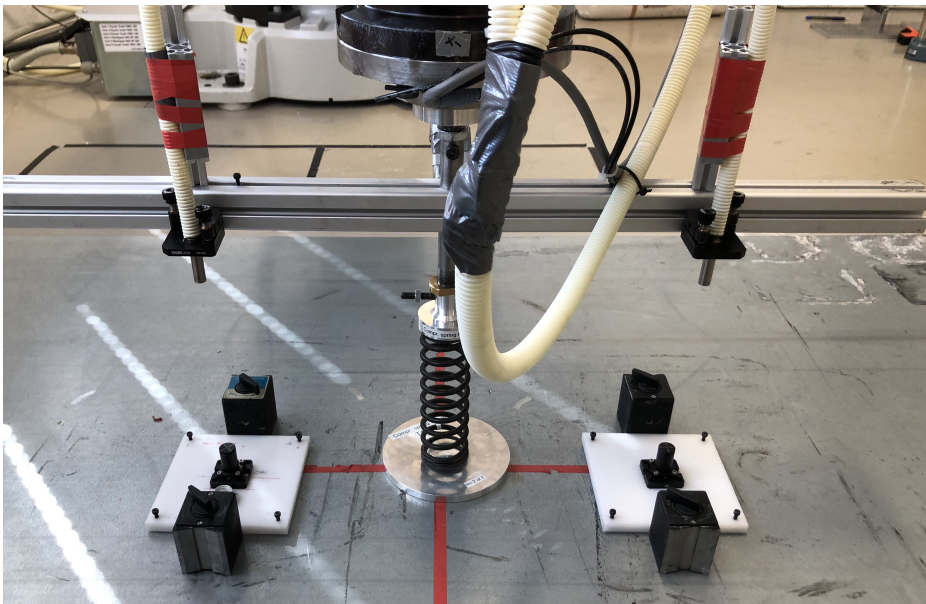
## 5.5 Spring compression test

---

It is important to keep in mind that since the compensation scheme is using the load cell which has a known static error and an unknown dynamic error, and the robot's internal position measurement, it is expected that this will contribute to errors in the results.

## 5.5 Spring compression test

In order to acquire a measure of how well the compensation scheme actually performs in a practical setting, dynamic tests using the two springs previously mentioned have been performed. The setup for these tests is shown in figure 5.8.

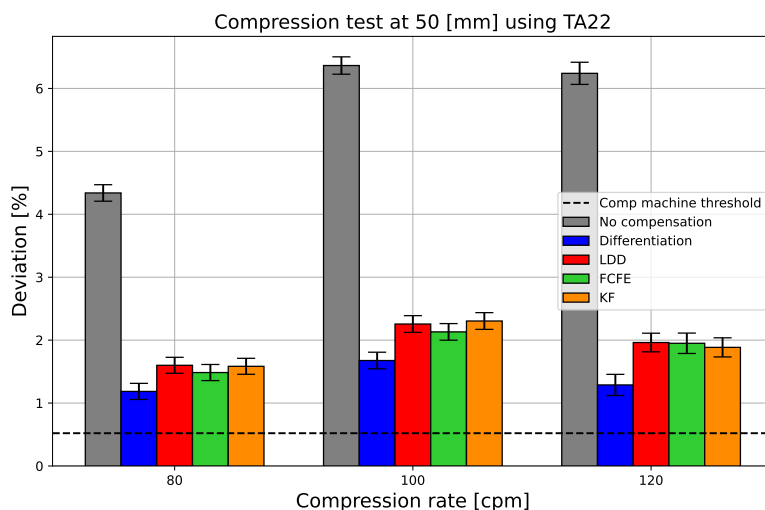


**Figure 5.8:** Test setup for compression tests with spring TA22. Interferometer is mounted like on static measurement. The scale AH46 is not used.

The tests are performed using the compression test application described in section 3.2, assigning a zero point at  $5[N]$  pre-load like in the static measurement test. The deviation tolerance of the compression machine is again included as a percentage of the reference value.

## 5.5 Spring compression test

---



**Figure 5.9:** Measurement deviation from static reference at 50 [mm] using TA22 spring.

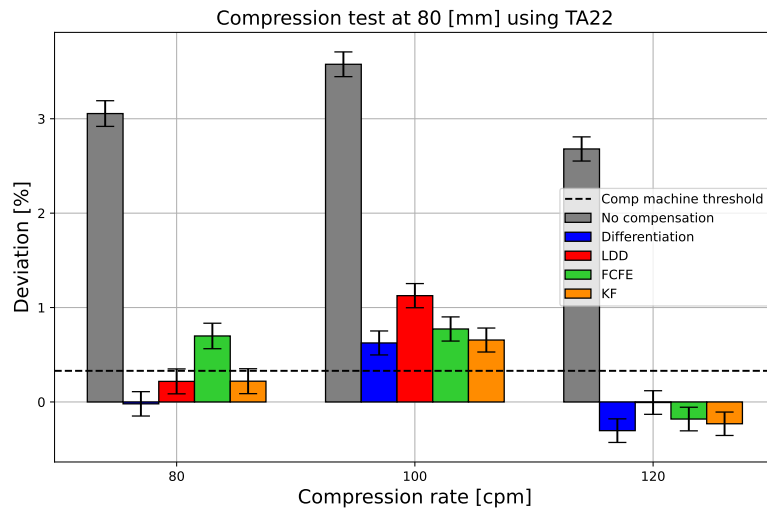
The deviation seen in figures 5.9 - 5.11 is calculated as described in section 2.4.1 according to the test procedure described in section 5.1. The average is compared to the result of the spring rate multiplied by the average compression depth as measured from the interferometer ( $F_{ref}$ ). The resulting deviation is converted to a percentage of the reference value. Since the reference value changes depending on compression depth and spring type, it is more convenient to present results as percentages. Uncertainty values are calculated according to the procedure described in section 2.4.2.

From the first set of tests shown in figure 5.9, it can be seen that the Differentiation method gives the smallest deviations from the reference. It can also be seen that all the estimators provide significantly improved accuracy when compared to the results without compensation at all, similar to the compression test without resistance.

The same trends are present in the subsequent results as well (figures 5.10 and 5.11). In both these cases it can be seen that the compensated results, regardless of estimators have even greater effect compared to the non-compensated cases. Again, the Differentiation estimator yields the smallest deviation across most trials. Of the other estimators, it is difficult to mark

## 5.5 Spring compression test

---



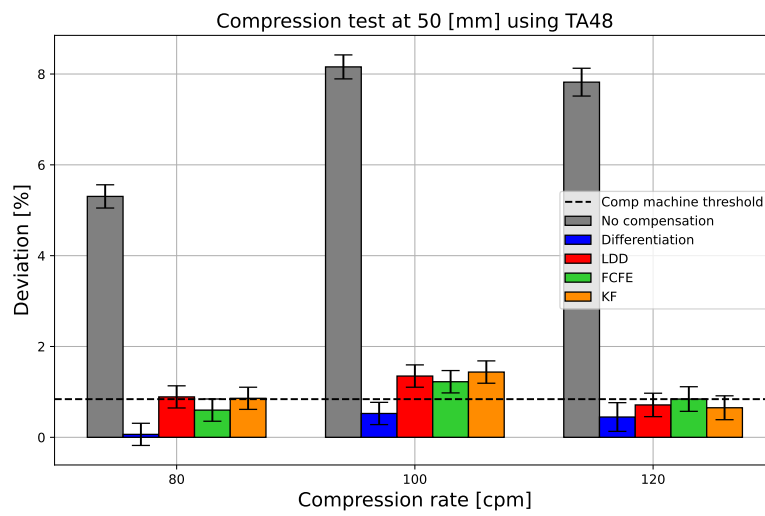
**Figure 5.10:** Measurement deviation from static reference at 80 [mm] using TA22 spring.

one as better than the others as they have very similar performance and no one is consistently outperforming the others.

Also note that the spring TA48 was only evaluated at 50 [mm] depth. A dynamic test was attempted at this depth, but the spring reached absolute compression causing overload in the load cell, thus ending the test run after 2 compressions.

## 5.5 Spring compression test

---



**Figure 5.11:** Measurement deviation from static reference at 50 [mm] using TA48 spring.

## Chapter 6

# Discussion and future work

Trough implementation, experimental testing and review of results, a number of topics worthy of discussion have arisen. Firstly, a summary of results is given, as well as findings from results that are worthy of discourse. Secondly, topics of implementation for the future and other solutions that have potential to be improved upon, are shed light on.

### 6.1 Discussion

#### 6.1.1 Summary of findings

Results indicate that the compensation scheme has a positive impact on the accuracy of the load data. All estimators provide an improvement, although the comparative performance is difficult to separate in a definitive manner. Uncompensated peak load error is in the area of 3 - 8 % of the reference value depending on the spring, compression rate and compression depth. The compensated errors are in the area of 0 to 2.5 % while uncertainties for compensated and uncompensated results are in the range of  $\pm 0.1 - 0.15$  %. Generally speaking all the estimators are able to significantly reduce the error, but crucially most fail to reduce the error to the threshold set by the compression machine, which is 0.4 [kg] (see appendix B).

## 6.1 Discussion

---

With regards to the objective of increasing the load data accuracy by compensating for load caused by tool mass acceleration, results conclusively indicate that the peak load deviation is effectively reduced using the compensation scheme for compressions with and without spring resistance.

Results also indicate that the objective of increasing the load data accuracy to within the tolerance specification of the compression machine, has not been accomplished. Although the failure of reaching this objective raises doubts over using an industrial robot for this application, there is also potential objectives for future work which might contribute to further reducing the load deviation to within the specified level.

When reviewing results, it has been attempted to gain insight into how compression parameters affect the compensation error, in an attempt to better understand the system under consideration. Neither compression rate nor depth conclusively indicate an increase in peak load deviation. There is potentially many factors behind this, and it deserves to be investigated further.

The relationship between compression parameters is complex. The load deviation error is suspected to be especially affected by the robot's internal position error which is contributed to by factors that are discussed in section 6.1.9. Most prominent is likely the compliance error caused by the spring force reacting on the robot.

Additionally, the robot is subject to additional errors caused by having to work against its own inertia, and any eventual backlash errors occurring at the point of directional change which also happens to be at the maximum point for acceleration and force.

This work has also led to a better understanding of the causes and effects of the robot's internal position error, although it is far from a full comprehension. The identification and compensation of the position error remains an objective which has not reached any conclusion. It is of high interest due to the fact that not only is it an aim to have the best position data possible, but it will lead to better acceleration estimates. This would in turn help to improve the accuracy of the compensation scheme.



## 6.1 Discussion

---

### 6.1.2 Differentiation Estimator

Although a very simple method, the differentiation estimator showed promising results in spring compression testing, consistently providing the smallest deviation of all the methods. In the unloaded tests it consistently showed the poorest accuracy of the estimators, however only by a small margin. The method displayed poor robustness to noise in simulation, to the degree that the output became fully compromised at SNR equal to 60 [dB]. The method's sensitivity to noise is well known and was expected. The robot's position output had a relatively low noise, thus the method still proved effective in the experimental tests. The estimator was also the most computationally efficient, as expected. However, the low noise robustness make it difficult to conclude that this is the best estimator for implementation in the compensation scheme.

### 6.1.3 LDD Estimator

The LDD estimator showed promising results during simulation, being unaffected by noise and consistently showing small deviations and comparatively low RMSE values across the whole range of tests. The step-wise computation time was also considerably lower than FCFE and KF methods. In unloaded compression testing the method showed consistently lowest deviation, with a few exceptions. In the spring compression testing it showed no particular advantage over the other estimators. All results taken into account, the method showed to be both accurate and robust, making it the preferred candidate.

### 6.1.4 FCFE Estimator

The FCFE estimator showed little advantage over other methods during simulation, while also being very sensitive to noise. The order of the method was chosen as the order which gave the smallest deviation across the whole range of compression rates and depths. However the parameters which yielded best accuracy made the estimator quite noise sensitive, although not to the same degree as for the Differentiation estimator. Generally in

## 6.1 Discussion

---

the experimental results presented in chapter 5 the estimator shows no significant performance advantage, however it performs on average very similar to the other estimators. Computation time was lower than for the KF but considerably higher than the other two methods.

### 6.1.5 KF Estimator

The KF estimator showed large deviations in the simulation, but also showed low uncertainty even with higher noise signals. Also in the experimental acceleration evaluation it consistently showed comparatively large deviations. In the unloaded compression testing it showed no performance advantage over other estimators. It showed slight but inconsistent performance advantage in the spring compression tests. Also, as expected this method is the most computationally expensive.

Results show that the use of the KF might not have served it justice. It is neither based on the physical model nor the target trajectory. This might explain why it is outperformed by the LDD in the simulation case and also to some degree in the experimental case. This is also a point of consideration for further improvement. The current implementation only bases its estimations on measurement and noise characteristics, and while those elements are important for a KF implementation, the lack of physical parameters other than the ones describing the motion is absent, and thus the performance might not reflect the utility of a typical KF.

The KF could be improved by being implemented as a model using the analytical trajectory of the position, velocity and acceleration as described in section 4.

Also, the assumption of time-independence of the process and measurement covariances would have made it possible to perform offline computation of steady state  $M_{ss}$  and  $K_{ss}$  using the algebraic Riccati equation as done in [7]. This would significantly improve the time performance of the estimation.

Although the current implementation serves as a valid candidate, especially considering the robustness and comparatively good accuracy of the method, it does not provide any large advantage over the LDD Estimator. Especially with regards to the peak acceleration deviation for both simulation and

## 6.1 Discussion

---

experimental data.

### 6.1.6 Previous data acquisition implementations

As mentioned in section 3, several iterations of development has been necessary to ensure that data could be captured with adequate frequency and consistency. An aspect of the implementation that was not straightforward. By consistency it is meant that sampling occurs at a rigid sampling rate.

This mainly involves the central parameters for the data acquisition done as part of this thesis: position and force data. Previous implementations has lacked both in update speed and consistency, meaning that the values, due to their implementations, have been updated at highly variable intervals. Fortunately this has been significantly improved with the final iterations of implementations.

The acquisition of force data was first implemented as a websocket thread using the subscription service provided by RWS. However this implementation had some major drawbacks. The implementation was able to capture data at a sampling interval in the area of 5 - 15 [ms]. There were two problems with this: The sampling interval was not constant, and towards the higher end of the interval it is also too slow to be used as part of the compensation scheme. Thus the more direct approach of accessing the data directly from the *NetBox* was implemented, as described in section 3.3.2.

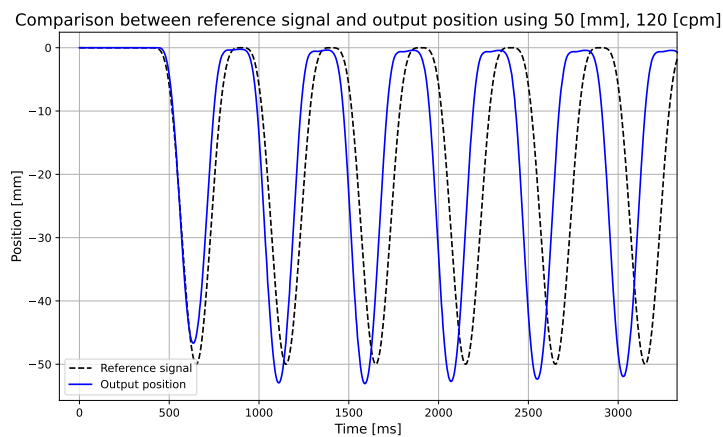
The previous implementations for position data acquisition was based on accessing the data through RAPID, however this also proved to be a slow and inconsistent approach. The inconsistency was manifested by gaps in the data when looking at the output with respect to time. This is very likely due to the poor multitasking performance of RAPID, which has to handle both the execution of the program commands and simultaneous calculation of robot trajectories. This was eventually solved by the implementation taking advantage of the EGM Protocol.

## 6.1 Discussion

---

### 6.1.7 Acceleration estimate comparison with simulation

The main cause of difference in results when comparing the simulation and experimental results is the difference in the generated signal for simulation and the robot position output. These have been plotted against each other in figure 6.1. It can be seen that although the signals have the same general shape, there are some differences that amplify when considering the respective acceleration outputs. Thus the same experiments cause differences up to  $3000 [mm/s^2]$  for peak acceleration estimates when comparing simulations and practical experiments.



**Figure 6.1:** Comparison between the generated signal used in simulation and the position output measured from the robot. Note that the reference signal is not directly passed into the RAPID implementation, rather it is the same function modified to work in RAPID.

### 6.1.8 Measurement deviation

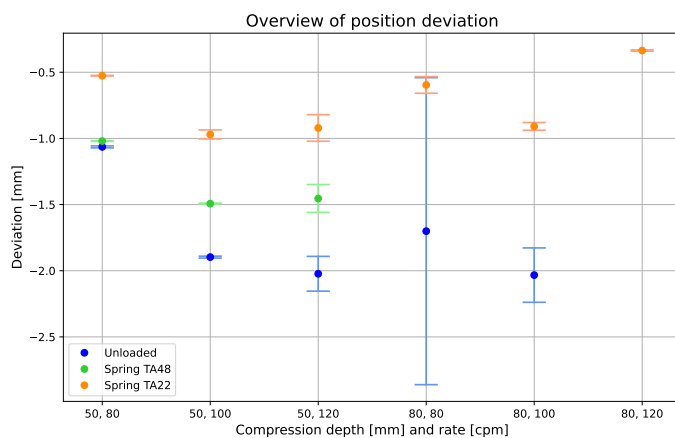
As mentioned, finding the relationships between compression parameters and resulting deviations from the results presented in chapter 5 can be difficult. For example, in section 5.5 the resulting deviations shows a slight tendency to rise with the compression rate. However, this is most prominent in the increase from 80 to 100 [cpm] (see figures 5.9 - 5.11). Note that in the case of the spring TA22 at 80 [mm] and 120 [cpm] there is actually a

## 6.1 Discussion

---

noticeable decrease in error for uncompensated and compensated deviations alike. This can be seen in figure 5.10.

To gain further insight into this issue, the figure 6.2 shows the deviation between the robot position measurement and the interferometer measurement. The stiffest spring (TA22) shows no clear increase in positional error neither with respect to compression rate nor depth. For the softest spring (TA48) the deviation only changes by increasing the compression rate from 80 to 100 [cpm], while the rate increase from 100 to 120 [cpm] does not appear to have an impact on deviation. Note that this test was only performed with compressions to 50 [mm].



**Figure 6.2:** Position deviations with standard deviation for experiments with and without resistance.

For the unloaded tests the increase in deviation from 80 to 100 [cpm] at 50 [mm] depth can also be observed. However for the remaining tests, the deviation value resides in roughly the same area. Note that the last compression deviation could not be calculated as one of the interferometer sensor heads were misaligned during the test. However since the test was conducted without resistance it did not cause any issues with the compensation result.

The trend that is clearly observed in this figure however, is that the position deviation increases when less compression resistance is applied. The physical cause of the effect can be interpreted like this:

## 6.1 Discussion

---

In the bottom half of a compression the robot is attempting to reverse its direction of movement. Thus, it is working against the combined force of its own inertia and gravity. For a robot arm with a mass of 900 [kg] (see appendix B) these forces/torques are not trivial by any means. The combined force of all joints and links trying to reverse its direction of motion causes strain in the joints of the robot, resulting in the deviation.

- **Stiffest spring (TA22):** The stiffer spring causes a stronger resistance to the load applied to it by the robot. When the robot is reversing its movement, the spring is essentially aiding it in this process, thus causing only a small deviation.
- **Softest spring (TA48):** The softest spring is causing some resistance to the load, however not to the same degree of the stiffest spring. It is aiding the robot in the reversal of its motion, but not as much.
- **No resistance:** When the robot is moving without resistance, it has no aid in reversing its motion.

Considering the static deviation results presented in section 5.4, the situation looks different. The deviation here occurs in the opposite direction (see tables 5.1 and 5.2). Since there is no motion to consider, the load of the spring and any geometrical errors are now the only factors contributing to deviation. In this case the softest spring causes the smallest deviation, as the spring force is naturally smaller, thus exerting a smaller force on the robot components, which experience less strain.

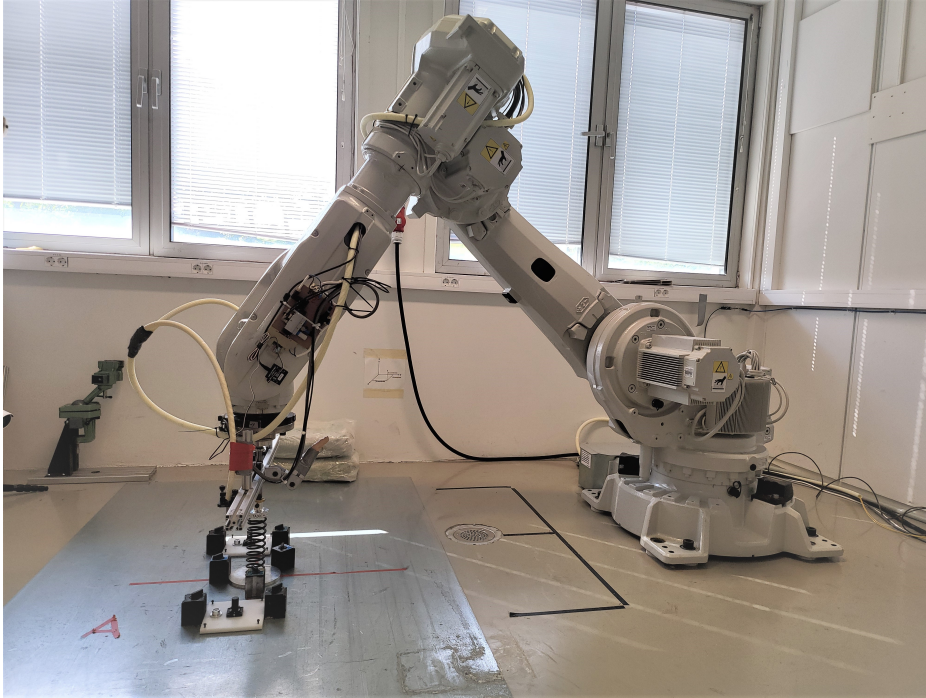
Although the deviations are not catastrophic, nor are they non-trivial as they cause the load cell to be exposed to different loads than the position combined with the spring rate might suggest. This way of operating the robot is likely not what it was intended for, as it might be unreasonable to expect this kind of accuracy from a 6-axis robot manipulator, compared to for example the purpose built compression machine, a far stiffer but also limited design. The stiffness of industrial robot's have been a topic of interest in research [25], [40].

Also, the pose of the robot must be noted. The tests have been performed in the middle of the test working area, where the robot has a stretched out pose, with an angle at joint 3 (see figure 6.3). This pose causes the robots

## 6.1 Discussion

---

joint number 2 to have to perform much of the work of moving the robot. Although this motor is large, working against the torque of the weight of the entire arm surely causes a significant load on this motor, which might also explain why the robot overshoots the reference in dynamic testing.



**Figure 6.3:** Robot in working pose.

### 6.1.9 Positional error sources

Considering the application of an industrial robot for the task that has been the focus of this thesis, four potential sources of positional error have been identified and will be further discussed:

- **Compliance error:** The application of the robot exposes components to a high load.
- **Dynamic error:** Caused by the elasticity of the robot joints and/or links in a dynamic environment.

## 6.1 Discussion

---

- **Geometric error:** Caused by the the robot's position and pose in the working area.
- **Backlash error:** Occurs when direction of movement is changed, as a result of mechanical tolerances in joint gears.

When the robot is in the maximum depth position, all the mentioned error sources are manifested. There is a high load on the robot due to the spring tension. The motion is considered high-speed with rapid changes in direction. In addition the geometric error of the robot in the designated working area must be considered.

Compliance error occurs in high load applications, and is a result of the elastic strain in the robot's mechanical components caused by the applied load. Compliance error has been researched in the context of robotic milling and machining applications. In [25] the authors examined compliance error in contact applications using contact forces of 150 - 650 [N] in two high load robotic manipulators. The evaluation was based on movement in circular trajectories at speeds of 10 - 50 [mm/s]. The movements were performed in different locations of the working area. The results for robot 1 (max. payload 300 [kg]) at 400 [N] and 650 [N] contact force showed that the error was typically in the area of 0.6 - 0.7 [mm] and 1.0 - 1.1 [mm] respectively. Error was significantly higher for robot 2 (max. payload 150 [kg]) at 1.90 and 3.42 [mm] for 400 and 650 [N] respectively. Although the error is not equal it is of a similar order compared to the static test results in this thesis. Also note that this study was done with the force acting in multiple directions of the robot coordinate system.

In [40] the authors implemented a real-time laser tracking system to compensate for compliance error for the application of robotic drilling. Testing consisted of drilling circular holes. The results showed that without compensation feedback the hole position errors were around 0.83 - 0.45 [mm], which is also comparable to the static results in this thesis.

Several other contributions have been made with focus on position error in the field of robotic machining and milling such as [41] and [42]. As mentioned, during compression testing the spring load actually contributes to reducing the position error. It must still be taken into account in the eventual implementation of a position compensation scheme.



## 6.1 Discussion

---

The positional error caused by the dynamics in a robot adds another layer of complexity to the issue. In [43] the authors perform an inverse dynamic analysis and evaluation of position error in a heavy duty industrial robot. The work aims to achieve trajectory optimization and position error compensation in heavy load and high-speed applications.

In [44] work was done on a real time position error compensation scheme based on an external laser tracker. Differently to the interferometer used as part of this thesis, the authors employed a tracker that follows the robot TCP and measures its position. The proposed dynamic compensation scheme reduced the position error to 0.02 [mm].

[25] also describes the effect of the geometrical error by experiments done in different circular trajectories at multiple points in the manipulator working space. However as described above the error was similar in the working areas that were tested.

In order to improve the positional error, one alternative is to simply relocate the testing area closer to the robot's base, although there is no guarantee that this would improve the error. Another alternative is to perform a mapping or calibration of the geometrical error and correct it based on the robots position. This is naturally more extensive work depending on the desired size of the calibrated working area.

It must be stated that the study of the sources of the position error has not been a primary focus of this work. This section aims to highlight some of the probable causes of the error that was found as a result of the experiments conducted in chapter 5. It should be recognized that this field of study is large and complex, and that finding the optimal solution for the application under consideration is not an easy task.

It should be taken into account that other error sources that are not specified might also have affected the results. The determination of all error sources and the degree to which they affect the results is a complex process. As mentioned, the elimination of position error also has relevance for the load compensation performance as the position errors are magnified by differentiation.

## 6.2 Further work

---

### 6.1.10 Impact of pre-load on deviation

Applying pre-load is a common practice during testing to ensure that the object and compression tool is in firm contact at all times, in order to avoid unintentional movement of the contact point or test object. However, the force caused by the pre-load might have a negative impact on the deviation in two ways:

- As the static experiments have been conducted using a zero-point with a pre-load, the spring rate for each spring is affected by this.
- Also in the spring compression test, the tool mass is calculated at the point of contact, so the pre-load also have an impact on the compensation scheme directly.

One method that could improve the contribution of error from this is to estimate the mass of the tool, before it is in contact with the test object. This is simply a manner of implementation choice. This can also be considered a flaw in the static testing procedure and might have contributed to the static measurement error.

## 6.2 Further work

### 6.2.1 Compliance error compensation scheme using Interferometer

The interferometer presents opportunities to improve the position error. One alternative is to perform an offline compensation in which the error is measured by dynamic tests in compression depth intervals of 10 [mm]. The errors can then be compensated for by adding a constant compensation factor for each compression depth zone. This method is quite simple and could be effective, although it has some drawbacks: For example this compensation scheme does not account for variance in spring stiffness. This means that for test objects where spring rate deviates substantially from the spring used for compensation, it will become inaccurate. One immediate solution

## 6.2 Further work

---

to this is to also account for spring rate in the compensation scheme. However this makes the compensation process more labour intensive, depending on how many springs are used.

Another alternative is to fully integrate the interferometer into the test setup. This would provide extremely accurate position data, and also grant the estimator with the most accurate data, thus potentially improving the load compensation scheme further. However, this also has some drawbacks which must be explained. Although the interferometer is a very accurate instrument, it requires a tedious routine for setup and calibration before conducting each experiment, such as sensor head and reflector alignment and calibration using the recording software. In addition, the sensor heads are very delicate and should be handled with care. Also, the reflectors are sensitive to dust and should also be handled in a manner to avoid dust collection as best as possible. Unfortunately, with the test center and robot being open to any Laerdal Medical employees, the correct handling of this equipment by an average user cannot be guaranteed. The handling of the instrument would require additional training, thus elevating the threshold for use. Thus, this implementation could prove counterproductive since the aim of the compression testing application is to lower the user threshold.

### Automation of calibration procedure

If the first alternative to the compliance error compensation scheme using the interferometer was to be implemented, it would be desirable to have an automated calibration procedure that was integrated into the compression test application. The calibration procedure would ideally be behind a password protection, and when the calibration was done for all depths, a report was produced including the measured errors and the new compensation parameters.

#### 6.2.2 Compliance error compensation scheme using load cell

A third alternative solution to achieve increased accuracy of position data, is to implement a compensation system using the load cell as described in [25]. This would eliminate the need for any additional equipment other than the

## 6.2 Further work

---

load cell which is already integrated in the testing system. This also would not add any complexity for the end-user such, nor would it require any regular calibration as it is an online compensation system.

### 6.2.3 Improvement of acceleration estimation

In order to improve upon the acceleration estimators, the aforementioned compliance error compensation scheme would very likely improve this performance. Additionally, depending on the selected estimator there are additional methods for improving the estimates.

Although the aim of this work has been to design an estimator based on position measurement only, results indicate that it might be necessary to include additional equipment to achieve the specified load data accuracy. One alternative is then to base the estimator on a combination of the position data and an accelerometer such as the one used for the comparison as part of this work.

With regards to the KF estimator, there is as mentioned other methods of implementation that might also yield better estimates than the implementation used as part of this thesis, such as a physical model or a trajectory based implementation.

Other types of prediction estimators could also be considered using the planned trajectory according to the compression function described in chapter 4.

### 6.2.4 Dynamic load cell performance/time response

An evaluation of the dynamic time response has not been performed as part of this work, however it would be valuable to gain insight in the transient response of the load cell which is not well documented, and not documented at all in the documentation for the load cell.

An initial test showed that when performing a rapid change in z-direction against a spring, the load cell provided an immediate response but then

## 6.2 Further work

---

settled at 3 [N] below this value.

Since the test center at Laerdal already has equipment such as the interferometer it is possible to conduct experiments with high accuracy to evaluate the dynamic characteristics of the load cell.

## Chapter 7

# Conclusion

The main objective of this research has been to increase the accuracy of load data of CPR-manikin compression tests by using an acceleration estimator based load compensation scheme. An established aim was to achieve the level of accuracy of the compression machine of 0.4 [kg].

The results indicate that the compensation scheme improves the load data accuracy by 1.5 - 6 % when accounting for measurement uncertainties depending on test case and choice of acceleration estimator.

Based on the combined performance in all test cases and simulations the LDD acceleration estimator has to be considered the best method when taking into account the accuracy, robustness and low complexity of the method.

Results also indicate that reaching the level of accuracy possible with the compression machine has not been achieved on the robot. In order to achieve a comparable performance, several issues should be further addressed:

- To match the level of dynamic accuracy, acceleration estimation based on position measurements alone might not be sufficient. The estimator might need to be based on a system using both accelerometer and position measurements in unison. Alternatively, other forms of sensor fusion could be employed.

## Conclusion

---

- A compensation scheme for the dynamic compliance error should be implemented. It should ideally be based on using a physical model of the robot combined with load cell measurements. Optionally, a simpler solution might be performing a periodical position error calibration scheme using the interferometer.
- Further improvement of position only based estimators, such as the KF.

Also, as the work on this thesis has uncovered, it is of high relevance to conduct work evaluating the relation of position error to the compression test parameters, such as compression depth, rate and stiffness of the spring/test object.

With these remarks in mind, it is still possible to conclude with a success of the as the objectives were partially accomplished. The current compensation scheme yields significantly better accuracy than not using a compensation scheme at all.

This work can be considered a contribution to the field of dynamic load compensation methods and acceleration estimators. For the field of dynamic load compensation it can be considered a novel application of a simple compensation scheme for problems that do not require compensation in a multi-axis configuration. For acceleration estimators it can be considered a study of the performance of the estimators considered in this work. Additionally this work can be considered a contribution to Laerdal Medical's product quality assurance and ultimately their work towards helping save lives.

# Bibliography

- [1] Jan-Thorsen et al. Gräsner. European resuscitation council guidelines 2021: Executive summary. *Resuscitation*, 161:1–60, 2021.
- [2] M. Uchiyama and K. Kitagaki. Dynamic force sensing for high-speed robot manipulation using kalman filtering techniques. In *Proceedings of the 28th IEEE Conference on Decision and Control*,, pages 2147–2152 vol.3, 1989.
- [3] Saverio Farsoni, Chiara Talignani Landi, Federica Ferraguti, Cristian Secchi, and Marcello Bonfe. Compensation of load dynamics for admittance controlled interactive industrial robots using a quaternion-based kalman filter. *IEEE robotics and automation letters*, 2(2):672–679, 2017.
- [4] Daniel Kubus, Torsten Kroger, and Friedrich M. Wahl. Improving force control performance by computational elimination of non-contact forces/torques. In *2008 IEEE International Conference on Robotics and Automation*, pages 2617–2622, 2008.
- [5] Juliang Xiao, Saixiong Dou, Wei Zhao, and Haitao Liu. Sensorless human-robot collaborative assembly considering load and friction compensation. *IEEE robotics and automation letters*, 6(3):5945–5952, 2021.
- [6] Shih-Hsuan Chiu, Cheng-Chin Chen, Kun-Ting Chen, Xin-Jie Huang, and Sheng-Hong Pong. Joint position-based impedance control with load compensation for robot arm. *Journal of the Chinese Institute of Engineers*, 39(3):337–344, 2016.



## BIBLIOGRAPHY

---

- [7] João Falcão Carneiro and Fernando Gomes De Almeida. On the influence of velocity and acceleration estimators on a servopneumatic system behaviour. *IEEE Access*, 4:6541–6553, 2016.
- [8] Johnny Rodriguez-Maldonado. Estimation of angular velocity and acceleration with kalman filter, based on position measurement only. *Measurement : journal of the International Measurement Confederation*, 145:130–136, 2019.
- [9] P.R. Belanger. Estimation of angular velocity and acceleration from shaft encoder measurements. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 585–592 vol.1, 1992.
- [10] Fargham Sandhu, Hazlina Selamat, S. E. Alavi, and Vahid Behtaji Siahkal Mahalleh. Fpga-based implementation of kalman filter for real-time estimation of tire velocity and acceleration. *IEEE sensors journal*, 17(17):5749–5758, 2017.
- [11] Seyed Ali Baradaran Birjandi, Johannes Kuhn, and Sami Haddadin. Joint velocity and acceleration estimation in serial chain rigid body and flexible joint manipulators. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7503–7509. IEEE, 2019.
- [12] B Nalepa and A Gwiazda. Kalman filter estimation of angular acceleration. *IOP conference series. Materials Science and Engineering*, 916(1):12072, 2020.
- [13] Wen-Hong Zhu and T Lamarche. Velocity estimation by using position and acceleration sensors. *IEEE transactions on industrial electronics (1982)*, 54(5):2706–2715, 2007.
- [14] Jiexin Zhang, Pingyun Nie, Yuhang Chen, and Bo Zhang. A joint acceleration estimation method based on a high-order disturbance observer. *IEEE robotics and automation letters*, 7(4):12615–12622, 2022.
- [15] R.J.E. Merry, M.J.G. van de Molengraft, and M. Steinbuch. Velocity and acceleration estimation for optical incremental encoders. *Mechatronics (Oxford)*, 20(1):20–26, 2010.
- [16] Yuxin Su, Chunhong Zheng, and P.C. Muller. A simple linear velocity estimator for high-precision motion control. In *2006 IEEE Conference on Emerging Technologies and Factory Automation*, pages 23–29. IEEE, 2006.

## BIBLIOGRAPHY

---

- [17] Lisha Chen, M. Laffranchi, N. G. Tsagarakis, and D. G. Caldwell. A novel curve fitting based discrete velocity estimator for high performance motion control. In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1060–1065. IEEE, 2012.
- [18] Y.X. Su, C.H. Zheng, Dong Sun, and B.Y. Duan. A simple nonlinear velocity estimator for high-performance motion control. *IEEE transactions on industrial electronics (1982)*, 52(4):1161–1169, 2005.
- [19] R.H. Brown, S.C. Schneider, and M.G. Mulligan. Analysis of algorithms for velocity estimation from discrete position versus time data. 39(1):11–19, 1992.
- [20] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.
- [21] F. Auger, M. Hilaret, J. M. Guerrero, E. Monmasson, T. Orłowska-Kowalska, and S. Katsura. Industrial applications of the kalman filter: A review. *IEEE transactions on industrial electronics (1982)*, 60(12):5458–5471, 2013.
- [22] JCGM. *Evaluation of measurement data — Guide to the expression of uncertainty in measurement*, first edition, September 2008.
- [23] ABB, 3HAC050798-001. *Application manual - Controller Software IRC5*, a9 edition, 03 2021.
- [24] Mohamed Slamani, Albert Nubiola, and Ilian A. Bonev. Modeling and assessment of the backlash error of an industrial robot. *Robotica*, 30(7):1167–1175, 2012.
- [25] Monica Katherine Gonzalez, Nikolas Alexander Theissen, Asier Barrios, and Andreas Archenti. Online compliance error compensation system for industrial manipulators in contact applications. *Robotics and computer-integrated manufacturing*, 76:102305, 2022.
- [26] Xiaojia Shi, Fumin Zhang, Xinghua Qu, and Bailing Liu. An online real-time path compensation system for industrial robots based on laser tracker. 13(5):172988141666336, 2016.
- [27] Python executive summary. <https://www.python.org/doc/essays/blurb/>.
- [28] Qt documentation. <https://doc.qt.io/>.

## BIBLIOGRAPHY

---

- [29] A tour of the C# language. <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
- [30] ABB, 3HAC073319-001. *Application manual - Externally Guided Motion*, a5 edition, 09 2022.
- [31] Google protocol buffers overview. <https://protobuf.dev/overview/>.
- [32] ABB, 3HAC029364-001. *Operating manual - Introduction to RAPID*, a2 edition, 09 2013.
- [33] ABB, 3HAC050946-001. *Technical reference manual - RAPID kernel*, a468 edition, 08 2022.
- [34] Robotstudio suite. <https://new.abb.com/products/robotics/robotstudio>.
- [35] Robot web services. <https://developercenter.robotstudio.com/api/RWS>.
- [36] ATI Industrial Automation, 9620-05-Net FT. *Net F/T - Installation and Operation Manual*, 11 edition, 12 2013.
- [37] Attocube Systems AG. *Real-Time Displacement Sensor for Machine Integration*, 01 2018.
- [38] Phidgets mot 1100\_0 accelerometer. <https://www.phidgets.com>.
- [39] Timeit module. <https://docs.python.org/3/library/timeit.html#module-timeit>.
- [40] Zheng Wang, Runan Zhang, and Patrick Keogh. Real-time laser tracker compensation of robotic drilling and machining. *Journal of Manufacturing and Materials Processing*, 4(3):79, 2020.
- [41] Weidong Zhu, Guanhua Li, Huiyue Dong, and Yinglin Ke. Positioning error compensation on two-dimensional manifold for robotic machining. *Robotics and computer-integrated manufacturing*, 59:394–405, 2019.
- [42] Gang Xiong, Zhou-Long Li, Ye Ding, and LiMin Zhu. A closed-loop error compensation method for robotic flank milling. *Robotics and computer-integrated manufacturing*, 63:101928, 2020.
- [43] Kun Yang, Wenyu Yang, and Chunming Wang. Inverse dynamic analysis and position error evaluation of the heavy-duty industrial robot with elastic joints: an efficient approach based on lie group. *Nonlinear dynamics*, 93(2):487–504, 2018.

## BIBLIOGRAPHY

---

- [44] Seemal Asif and Philip Webb. Realtime calibration of an industrial robot. *Applied System Innovation*, 5(5):96, 2022.
- [45] User Datagram Protocol. RFC 768, August 1980.
- [46] Ross K Kurose J. *Computer Networking: A Top-Down Approach*. Pearson Education Limited, 7 edition, 2021.
- [47] Transmission Control Protocol. RFC 793, September 1981.
- [48] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Rfc 2616, hypertext transfer protocol – http/1.1, 1999.

# Vedlegg A

## Protocol overview

### A.1 User Datagram Protocol

UDP was defined to make available a datagram mode of packet-switched computer communication in the environment of an interconnected set of computer networks. This protocol assumes that the Internet Protocol (IP) is used as the underlying protocol. [45]

UDP is a transport layer protocol within the Internet protocol stack. UDP provides a procedure for applications to send messages to other applications with a minimum of protocol mechanism. The protocol is transaction oriented (as opposed to connection oriented), and delivery and duplicate protection are not guaranteed [45].

Essentially UDP provides the minimum overhead needed to pass an application message to the network layer. This includes some light error checking and multiplexing/demultiplexing. UDP takes messages from an application process, adds source and destination port number and two other fields to the header, and passes the resulting segment to the network layer. UDP uses no handshaking before any transaction between source and destination is initiated, in other word the successful transmission of an application messages is not something UDP is concerned with, leaving this to the task of ensuring reliability to the application developer. The lack of handshaking,

## A.2 Transmission Control Protocol

---

labels the protocol as connectionless [46].

One big advantage with UDP as opposed to TCP is that there is a better application level control of what data is sent, and when. This is useful for transmitting real-time data as it needs to be done as quickly as possible and at the time when desired. Additionally, if a message is lost, it is not of concern to re-send this message as the state of the application will have changed, thus the lost data has no value. TCP has both congestion control and re-sending functionality while UDP does not [46].

## A.2 Transmission Control Protocol

TCP is intended for use as a highly reliable host-to-host protocol between hosts in packet switched computer communication networks, and in interconnected systems of such networks.

TCP fits into the transport layer of the layered protocol architecture just above the network layer in which the IP is located. This provides a way for the TCP to send and receive variable-length segments of information enclosed in internet datagram "envelopes" [47].

TCP is said to be connection-oriented because application processes transmitting data between themselves must first perform a handshake. A handshake is some set of preliminary segments that establish the parameters of the ensuing data transfer.

A TCP connection provides full-duplex service, which means that if a connection and handshake has been established, application-layer data from process A can flow simultaneously to and from process B. The connection is also always point-to-point, i.e., always between a single sender and a single receiver [46].

Before application level data is sent further down through the transport layer and the network layer, TCP directs the data into a send buffer. An interesting note from the TCP specification [47] is that the specific time when the data is sent is not very rigid referring to the statement that TCP should "send data in segments at its own convenience.". This posed some

### A.3 Hypertext Transfer Protocol

---

issues for a real-time system and the I/O-signal (accessed from the robot-controller) *NetBoxFz* which is the force in Z-direction from the load cell, must be transmitted through TCP. This led to the force data stream neither being as fast as or completely in sync with the position stream. As mentioned in section 3.3.2 an implementation capturing data directly from the Net F/T system solved this issue.

### A.3 Hypertext Transfer Protocol

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990. HTTP allows an open-ended set of methods and headers that indicate the purpose of a request. It builds on the discipline of reference provided by the Uniform Resource Identifier (URI), as a location (URL) or name (URN), for indicating the resource to which a method is to be applied [48].

HTTP is as mentioned central in the communication between the controller and computer applications due to its integration in RWS, and is thus used as the application layer transport protocol for the compression test application, and among others in the load compensation application.

Vedlegg B

Hardware documentation



# Calibration certificate

Document No: Att 1 to 00083659

<b>Calibrated by:</b>		Eivind Tønnessen Senior Product developer Shared Services/TestCenter Laerdal Medical AS
-----------------------	--	--

Instrument tested		Reference instrument	
ID No:	AH53	ID No:	AH46
Name	Compression machine 1418	Name	Scale
Model:	SP4MC3MR	Model:	SP4MC3MR
Type	HBM Load cell	Type:	HBM Load cell
Sn:	01477483	Sn:	01457237
Range	0 - 140 kg	Range	0 - 150 kg
Location	TestCenter/Room "Dovre"		

<b>Environment:</b>	25 ± 3°C	50 ± 30 % RH
---------------------	----------	--------------

## Measurement method:

The calibration was performed according to Laerdal procedure PSS - PR01 - 0979 Rev. I. Calibration procedure for load cell and movement sensor.

Measurement uncertainty was calculated according to IKM doc. B06.07.04 Kalibrering av load cell i kompresjonssimulator.

Ref. Att 2 to 00083659 for raw data and calculations

All the measurements represent the average of three measurements.

At 30 kg there were performed five measurements. This were used to calculate repeatability.

## Calibration results

Reference value [kg]	Measured value [kg]	Deviation [kg]	Calibration accuracy [kg]	Uncertainty [kg]
0,00	0,01	0,01	0,20	0,035
15,15	15,15	0,00	0,20	0,041
30,34	30,33	0,00	0,20	0,047
45,07	45,05	-0,02	0,20	0,053
59,78	59,74	-0,04	0,20	0,059
75,24	75,20	-0,04	0,20	0,065
90,15	90,10	-0,05	0,20	0,071
105,28	105,22	-0,06	0,20	0,077
120,02	119,94	-0,08	0,20	0,083
0,00	0,01	0,01	0,02	0,035

## Conclusion

PASS

**Date of calibration:** 06.12.2022  
**Next Calibration:** December 2023

# Investigation report for NC-00006127

## Background

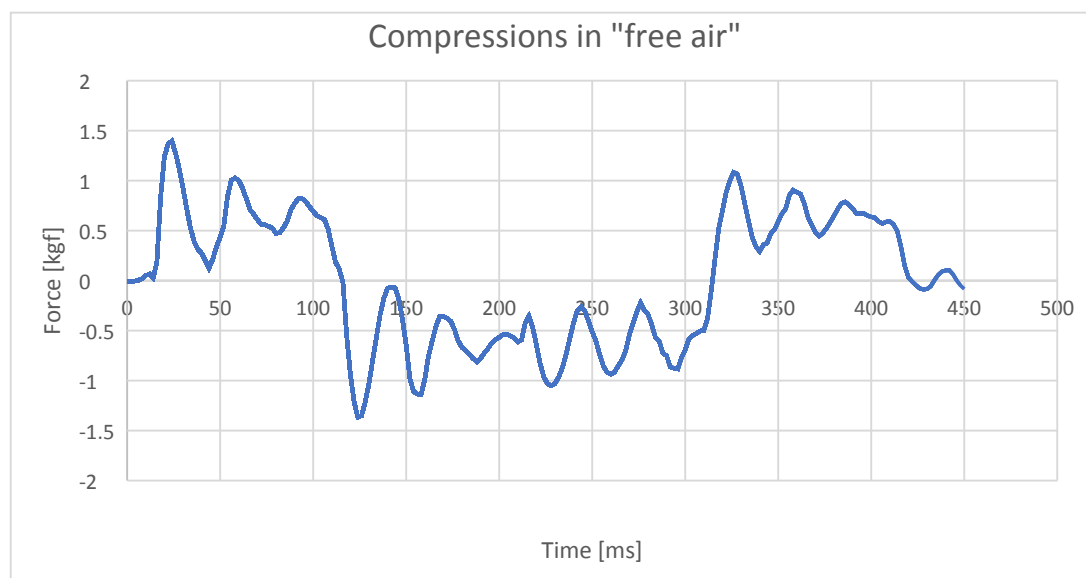
There have been some concerns that the forces measured by the compression machine (equipment no. 1418 with load cell AH53) may be less accurate than specified. The machine has been calibrated yearly since it was built, and has always been found to conform to the following tolerances:

Range [kgf]	Accuracy [gf]
0 - 70	±200
70 - 140	±400

However, the calibration is performed using a set of **static** loads. This is a situation very different from the typical operation of the machine, where compression forces vary between zero and maximum load in just a few tenths of a second. There are at least two important factors that pose a challenge when trying to measure correct forces in circumstances like this:

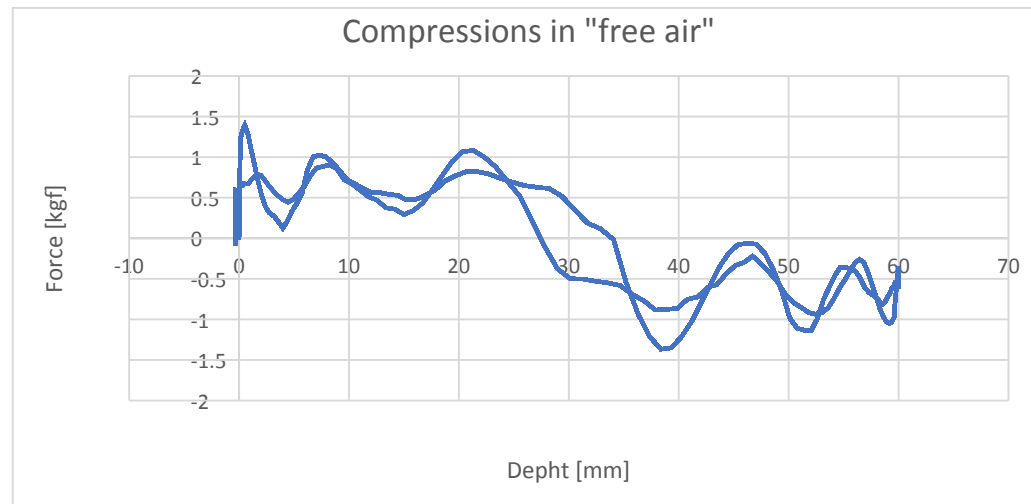
- The load cell is not in direct contact with the test object. When the mass of the compression tool and its fixtures are accelerated and decelerated during a compression cycle, it causes a difference in the force acting on the test object and the force acting on the load cell.
- There are several steps involved in the process of converting the physical forces applied to the load cell into values that can be presented to the user. The load cell must first generate its output signal, then that signal must be amplified (and maybe also noise filtered), before being passed to an A/D converter. The digital raw values may in turn be processed and refined even further by the computer software (e.g. averaging and filtering) before any force readings become available to the user. One or several steps in this process may have a damping effect on the force signal, preventing rapidly changing forces to be measured correctly.

A key finding when initiating the nonconformance was that the machine reports force values quite different from the expected "0.0 kgf" when compressions are performed in "free air":



In this simple test, the machine was set to a compression depth of 60 mm, and a rate of 120 per minute, using the default duty cycle of 40/0/40/20. The measured forces are most likely a result of the acceleration and deceleration of the compression tool and its fixtures, as described in section a) above.

Here are the same force values, this time plotted as a function of compression depth:



The plot shows that the load cell experiences a push when the compression cycle starts and ends, and a pull in the bottom part of the compression. These results caused concern, since the deviations from the expected "0.0 kgf" were much larger than the specified tolerances.

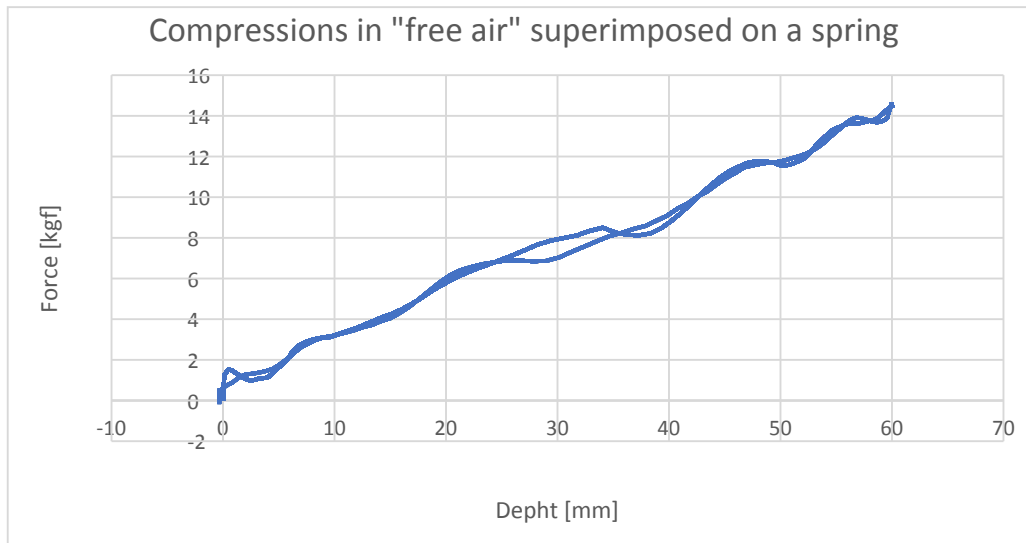
The challenges described in section b) are harder to address and quantify directly, in the absence of a high-speed reference load cell.

## Further investigations

In typical use of the compression machine, for example when testing manikin lifetime or accuracy, the key parameters for the user (with regards to force) are the maximum and minimum values for each compression. So, an important question is; how will the force deviations described above affect these values?

The plots show that the force deviations at maximum depth and zero depth are both less than 0.5 kgf. The deviations are much more severe in other parts of the cycle. But, the effect of these force deviations in the upper half of a compression cycle (when compression forces generated by the test object itself are low) will be to raise the force values, and the effect in the bottom half will be to lower the force values. So, in both cases the measured forces are shifted away from the extreme minimum and maximum values, as opposed to being shifted towards the extremes, and potentially generating values that could compete with the real minimum and maximum values.

To illustrate this, the force deviation data from above have been mathematically superimposed onto a linear spring with stiffness 0.25 kgf/mm, representing a very soft manikin with a compression force of 15 kgf at 60 mm depth:



This graph shows that when the compression machine is working against a counterforce (e.g. a manikin chest), the forces generated by the accelerated and decelerated tool/fixture influence the force curve in areas that typically are not important.

In an effort to address both the challenges described in a) and b) above, an extended version of the dynamic force tests in the validation protocol for the equipment has been performed (see validation report 00012375 rev. B). These test results support a **dynamic** force accuracy of  $\pm 0.4$  kgf, provided that the static forces are spot on perfect. To account for some variations in the yearly calibration (which will continue to use static forces), an extra  $\pm 0.2$  kgf will be added to the **total dynamic force tolerance**. The status form for the compression machine will be updated to reflect this conclusion.

## Retrospective analysis

The results described above are not in accordance with the current tolerances for the compression machine. However, the breach is relatively minor (less than 3x current tolerances). It is therefore judged **not necessary** to perform a review of all applicable test reports that have been written since the machine was built.

The new force tolerances will be available to all users through the updated status form.

**Document number:** 00012375  
**Document revision:** B  
**Document title:** Validation report for Compression test machine (no1418)

## Table of content

<b>1. Purpose and scope</b>	<b>2</b>
<b>2. Definitions</b>	<b>2</b>
<b>3. Authority and responsibility</b>	<b>2</b>
<b>4. References</b>	<b>2</b>
<b>5. Revision history</b>	<b>2</b>
<b>6. Description of the validation</b>	<b>2</b>
6.1. General	2
6.2. Measuring equipment used in validation	2
6.3. Results	3
6.4. Discussion of the results	3
<b>7. Conclusion</b>	<b>3</b>

## 1. PURPOSE AND SCOPE

The purpose with the validation is to verify that the Compression machine has satisfactory functionality and accuracy and fulfills requirements specified in ref. [1]. It is also verified that the safety for the operator is satisfactory.

This report describes the results from the validation for Compression machine no. 1418. It is performed according to ref. [3].

## 2. DEFINITIONS

R&D: Research & Development  
T&V: Test & Verification

## 3. AUTHORITY AND RESPONSIBILITY

R&D T&V department is responsible for carrying out the validation.  
Kåre Håvarstein and Eivind Vereide have performed the validation testing.  
The originator of this report is Eivind Vereide (R&D T&V department).

## 4. REFERENCES

[1 ]	PRO-RP01-1096 rev. E	Functional Requirements - Lab Compression Machine.doc
[2 ]	PRO-SP01-2466 rev. E	Documentation for Compression test machine (no 1418)
[3 ]	PRO-PR01-1020 rev. C	Validation Protocol for Compression test machine system (no1418)
[4 ]	PSS-PR01-0979 rev. F	Calibration procedure for load cell and movement sensor in compression machine (no 1418)

## 5. REVISION HISTORY

Rev A -> B:  
Added measurement of the dynamic forces, ref attachment 4.

## 6. DESCRIPTION OF THE VALIDATION

### 6.1. General

The software in the actual equipment "1418\_1\_CompMachine.exe" was in revision 2.0.2.1 during the validation. Software version used in this validation is described in ref. [2].

The validation is performed according to protocol ref. [3].

The validation was performed in the period January 25 – April 3 2012 in room temperature at LMAS Stavanger 2. floor by Kåre Håvarstein and Eivind Vereide.

The check of dynamic forces was performed May 19, 2022 in room temperature at Test center Stavanger 2. floor by Eivind Vereide and Sigmund Høyen.

### 6.2. Measuring equipment used in validation

- Stop watch.
- Slide calliper AA20

### **6.3. Results**

#### 6.3.1 Check list for validation

The list in attachment 1 was checked out. All the items were OK.

#### 6.3.2 Observations

No observations noted.

### **6.4. Discussion of the results**

Ref Dynamic forces measurement in attachment 4:

In the range of 0-70 kg the maximum deviation for dynamic forces is: 0,27 kg. This is not according to [1] chap 4.6 which state:

- The force accuracy shall be  $\pm 0,2$  kg for loads in range 0-70 kg.

This issue is addressed in NC-00006127.

Else: All other results were satisfactory and within the requirements.

## **7. CONCLUSION**

The validation task was performed according to protocol ref. [3].

With except of the dynamic forces, the results of the validation show that Compression machine has satisfactory functionality and accuracy and fulfills requirements specified in ref. [1]. It is also verified that the safety for the operator is satisfactory.

# CERTIFICATE OF CALIBRATION

No.: 23/12650



*Sign Lalden 20/3-23*



Page: 1 of 3  
Ref. to records:  
23/12650

<b>Date and place of calibration:</b> 09.03.2023 Laerdal, Stavanger	<b>Date of issue:</b> 14.03.2023
<b>Calibration performed by:</b> Laima Naudziunaite <i>[Signature]</i>	<b>Authorized signatory:</b> Sigurd Berge <i>[Signature]</i>

**Customer:**

Laerdal Medical as  
Tanke Svilandsgt. 30  
4001 STAVANGER  
Norway

**Contact:**

**Calibration item:**

Manufacturer: HBM  
Model: SP4MC3MR  
Options: 150  
Description: Load cell  
Serial no.: 01457233  
Tag/ID no.: AH46  
IKM ID: 112416  
Additional info: AH46  
Location: R&D Test & Verification  
Agreed next cal. 09.03.2024

**Conclusion of the results:**

Statement of conformity to accuracy specification: Pass.

**Remarks:**

Given accuracy specification is from customer.

**Condition:**

The item has no visible damage.

**Project:** 153203

**Customer PO:** on site

Accredited for compliance with NS-EN ISO/IEC17025 - Calibration. It states that the laboratory provide metrological traceability to the International System of Units (SI). The measurements results are traceable through calibrations provided by accredited laboratories or national metrological institutes under CIPM MRA. The results relate only to the item calibrated at the time and under the conditions that the calibration was performed. This report shall not be reproduced except in full.

ILAC-MRA (International Laboratory Accreditation Cooperation Mutual Recognition Arrangements) confirms mutual recognition of calibration certificates. Other signatories to ILAC-MRA are more than 60 countries world wide. Examples are National Association of Testing Authorities, Australia (NATA), Singapore Accreditation Council (SAC), United Kingdom Accreditation Service (UKAS), and many more. See full membership at [www.ilac.org](http://www.ilac.org).



# CERTIFICATE OF CALIBRATION



No.: 23/12650

Page: 2 of 3

**Cal. Procedure:** BL06.07.02 - Calibration of weighing instruments

**Method:**

The weighing instrument is calibrated by standard weights with air density 1,2 kg/m<sup>3</sup> and mass density 8000 kg/m<sup>3</sup>. The calibration is based on EURAMET cg-18 (11/2015) Guidelines on the Calibration of Non-Automatic Weighing Instruments. The results relate only to the item calibrated at the time and under the conditions that the calibration was performed.

**Environmental conditions:**

Air Temperature: 20 ± 5 °C  
 Humidity: 50 ± 30 %RH

**Traceability:**

The calibration is traceable to national or international standard, see below for details.

**Uncertainty:**

The reported expanded uncertainty of measurement is stated as the standard uncertainty of measurement multiplied by the coverage factor  $k = 2$ , which for a normal distribution corresponds to a coverage probability of approximately 95%. The standard uncertainty of measurement has been determined in accordance with EA-Publication EA-4/02. Measurements which are not covered by the scope of accreditation are marked with nc.

**Decision rule and statement of conformity:**

When a statement of conformity to a specification or standard is provided, the default decision rule is according to ILAC-G8:09/2019 section 6, choice b (4.2.3 Non-binary Statement with Guard band  $w=U$ ). Specific risks are: Pass/fail: < 2.5%, Conditional pass/fail: < 50%.

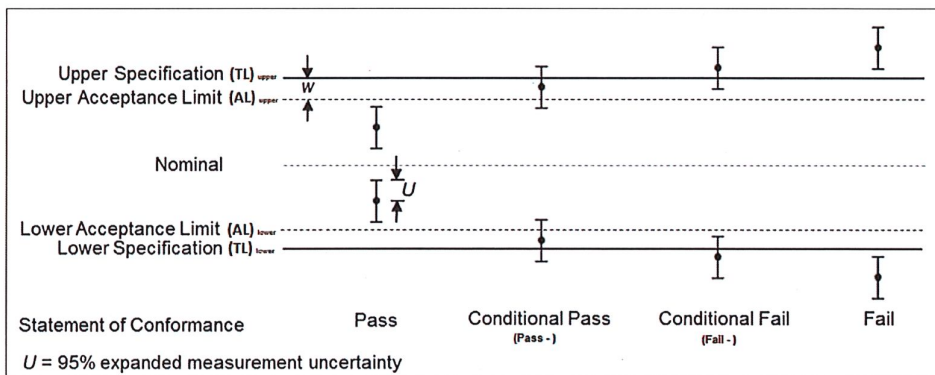


Figure 5 Graphical representation of a non-Binary statement with a guard band (shown for  $w = U$ )

**Measurement standards and equipment:**

Description	Manufacturer	Model	Tag/ID no.	Certificate no.	Traceability
Temperature Humidity Data Lo	Testo	184 H1	TE-184H1-01	22/30658	IKM/Fluke/NIST
Mass set	Kern & Sohn GmbH	M1	MA-1-215-M1	23/12681	IKM/RISE

# CERTIFICATE OF CALIBRATION



No.: 23/12650

Page: 3 of 3

**Location:** Floor, R&D, Test Lab  
**Range(s)/Resolution(s):** 150/0,01 kg  
**Specification:** 0,1/70kg; 0,2/150kg  
**Specification is set by:** Customer

**Calibration results:**

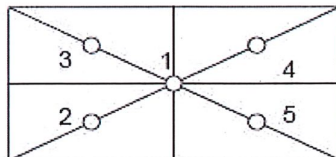
	Test load [kg]	Indication [kg]	Error [kg]	Uncertainty (k=2) [±kg]	
Cal. (as found)	20	20,0000	0,0000	0,0090	
	150	149,960	-0,040	0,020	
Adjustments:	No adjustment				
Measurement*	10	10,0000	0,0000	0,0088	
	20	20,0000	0,0000	0,0090	
	40	39,997	-0,003	0,012	
	80	79,990	-0,010	0,012	nc
	150	149,957	-0,043	0,024	nc

\* The measurement indication and error is based on three repetitions.

**Eccentricity test:**

Method: The load receptor is divided (visually) into 4 equal sections. The test load is placed in the center and the balance is tared. The test load is then placed in center of each section, ending with center.

Test load:	40 kg	Indication
		[kg]
Center (1) (Tare)		0,00
Front left (2)		-0,01
Back left (3)		-0,01
Back right (4)		-0,01
Front right (5)		-0,01
Center (1)		0,00



Front



Front



attocube systems AG

Tel. +49 (0) 89 4207 97 0

Fax +49 (0) 89 4207 97 20 190

E-Mail: info@attocube.com

# IDS Test Report

IDS Werkszeugnis

## General information / Allgemeine Angaben

Object <i>Gegenstand</i>	Interferometric Displacement Sensor (IDS)
Manufacturer <i>Hersteller</i>	attocube systems AG
Serial number <i>Seriennummer</i>	IDS3010-A01-2220
Number of pages <i>Anzahl der Seiten</i>	4
Date <i>Datum</i>	05.05.2021 09:37:37
Reference interferometer <i>Referenz Interferometer</i>	PTB* calibrated IDS3010 (IDS3010-A01-0009) <i>PTB calibration mark 54012 PTB 15 [1]</i>

\*: PTB: Physikalisch Technische Bundesanstalt, the National Metrology Institute of Germany

## Authorization / Ermächtigung

Signature <i>Unterschrift</i>	Signatory <i>Unterzeichner</i>	Position <i>Position</i>	Date <i>Datum</i>
	Dr. Martin Zech	CTO	15.12.2019

## Typical system accuracy / Zu erwartende Messgenauigkeit des Systems

When combining an IDS with the ECU (Environmental Compensation Unit), the displacement measurement at ambient is guaranteed to have an displacement measurement accuracy of below  $\pm 1$  ppm at distances up to 5 m. This test report guarantees that the IDS performs according to the given specifications.

*Die Weglängenmessung des IDS ist bei mittels ECU (Environmental Compensation Unit) kompensierten Umgebungsbedingungen und Distanzen bis zu 5 m garantiert genauer als  $\pm 1$  ppm. Bei Distanzen bis zu 4,2 m ist eine Genauigkeit von kleiner  $\pm 0.1$  ppm zu erwarten. Dieses Werkszeugnis garantiert, dass das getestete IDS die Spezifikation einhält.*

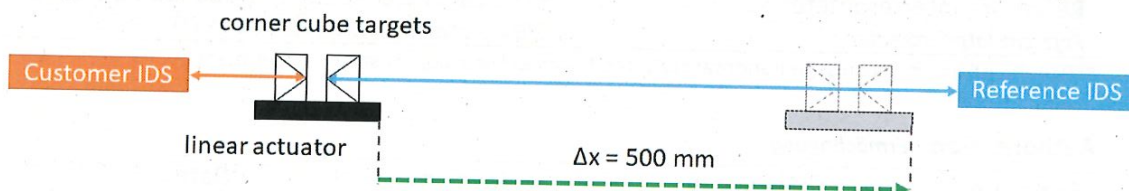


### 1. Description (NIST traceability) / Beschreibung (NIST Rückführbarkeit)

The measurement accuracy of the IDS mainly depends on the precise knowledge of the absolute value of the laser wavelength [1]. The IDS assumes the laser wavelength to be a defined and static value. A deviation from the internally stored to the true laser wavelength value would directly result in a systematic measurement error that sums up with additional displacement. To reference and stabilize the laser wavelength a gas cell is integrated in the IDS. This gas cell is filled with acetylene gas featuring stable absorption peaks at certain wavelengths. A control loop is implemented to adjust and stabilize the laser wavelength to the absorption peak at 1530.3711 nm. The position of the absorption peak is known with an expanded uncertainty (k=2) of  $\pm 0.3$  pm. This is certified by NIST, The National Institute for Standards and Technology of the United States [2].

### 2. Testing method (PTB traceability) / Testmethode (PTB Rückführbarkeit)

To ensure that the laser wavelength is 1530.3711(3) nm, an interferometric displacement measurement over a larger distance of 500 mm is performed. If, by mistake, the IDS is locked on one of the neighbouring gas cell absorption peaks, its absolute wavelength differs with a value of at least 0.5 nm. In the IDS displacement measurement, this wavelength error of 0.5 nm would be added with every periode of the interference pattern. The IDS interference pattern is periodic over half of the wavelength. Therefore the resulting systematic measurement error would be larger than :  $\frac{0.5 \text{ nm} \cdot 0.5 \cdot 10^{-9} \text{ m}}{\frac{1530}{2} \cdot 10^{-9} \text{ m}} = \sim 327 \text{ } \mu\text{m}$



In the measurement setup, the 500 mm displacement is performed with a linear actuator. Corner cube targets are mounted on the actuator to function as a measurement target. The IDS is combined with a collimated sensor head (M12/C7.6) to measure the displacement of the corner cube targets. The measurements with the test and the reference IDS are taken from opposite sides which follows that the target displacement is measured with an opposite sign. The opposite sign is corrected within the evaluation. The reference IDS is the device that was used for the IDS PTB calibration that successfully proved the IDS measurement accuracy [3]. For evaluation the difference between the measured displacement values is calculated. The customer IDS passes the test when the measured difference of test and reference IDS is below  $1 \text{ } \mu\text{m}$  which is clearly below the systematic error of a wrong laser wavelength. We follow that the correct gas cell absorption peak is locked and that the laser wavelength is 1530.3711(3) nm which enables the IDS measurement accuracy. In the testing setup the remaining difference is mainly caused by imperfection in the collinear alignment of both measurement beams. No active refractive index compensation is used due to the relative comparison of the devices.




### 3. Test results / Testergebnis

Test result Testergebnis	PASSED
-----------------------------	--------

The tested IDS fulfills the criteria (see testing method) that allows conclusion for the correctness of the wavelength and therefore the measurement accuracy. It will perform according to the specification that is given under "Typical system accuracy".

*Der Test erfüllt das Kriterium (siehe Testmethode) aus welchem der Rückschluss auf eine korrekte Wellenlänge und damit korrekte Systemfunktion des IDS gemacht wird. Das IDS erfüllt Spezifikationen aus dem Abschnitt „Zu erwartende Messgenauigkeit des Systems“.*

### References / Verweise

[1] attocube TechNote #19: "Metrology aspects of the IDS3010"	
[2] IDS laser wavelength NIST calibration certificate	
[3] IDS displacement measurement accuracy PTB calibration certificate	



## Test notes

1. **Purpose:** The aim of this test is to ensure that the tested IDS is within the specification that attocube guarantees (see typical system accuracy).
2. **Measurement procedure:** The test and the reference IDS measurement outputs are compared in a displacement measurements over a larger displacement that allows conclusions for a correct system performance (see testing method).
3. **Measurement setup:** Possibly collinear setup of the two measurement beams on a target that is moved by a linear actuator (see testing method).
4. **Pass criteria:** The tested IDS passes the test when the estimated deviation to the reference IDS is in a region that ensures the correctness of the laser wavelength which concludes the measurement performance (see testing method).
5. **Traceability:** The reference IDS is calibrated by the PTB, the National Metrology Institute of Germany). The wavelength reference gas cell is calibrated by NIST, The National Institute for Standards and Technology of the United States. PTB and NIST are members of several international metrology organisations such as the International Laboratory Accreditation Cooperation (ILAC). It is a signatory to the Multilateral Agreements (MLA) of these organisations. The mutual recognition of accreditations among the Member Countries ensures the acceptance of conformity assessments in Europe and all over the world.
6. **Quality accreditation:** attocube is certified and regularly audited for the following standards: ISO 9001 – quality management system, ISO 14001 – environmental management system, ISO 50001 – energy management system



## Detailed Certificate of Calibration

Equipment Calibrated		Customer Information
Description:	Six-Axis Force/Torque Sensor	
Manufacturer:	ATI Industrial Automation	
Serial Number:	FT25156	
Model:	Delta	
Calibration:	SI-660-60	
Electronics:	Net F/T	
Gain Multiplier:	100%	

Equipment Condition and Notes: Factory new.

Calibration Results: Passed

Offset: Normal

Gain: Normal

Calibrated Ranges ( $\pm$ ):

Fx	Fy	Fz	Tx	Ty	Tz
660 N	660 N	1980 N	60 N-m	60 N-m	60 N-m

Measurement Uncertainty (95% confidence level, percent of full-scale load):

Fx	Fy	Fz	Tx	Ty	Tz
1.25%	1.25%	1.50%	1.00%	1.25%	1.75%

The above Measurement Uncertainty values are the maximum amount of error for each axis expressed as a percentage of its full-scale load.

Calibration Temperature: 22.2 $\pm$ 1.1 $^{\circ}$  C (72 $\pm$ 2 $^{\circ}$  F)

Temperature Compensation: hardware

Calibration Method: WI-FTP-105, Net FT Calibration Instructions

Date of Calibration: 16 Jul 2018

Certificate Date: 16 Jul 2018

Calibrated by: Robert Perrone, Calibration Technician



This calibration is traceable to the National Institute of Standards and Technology (NIST).

ATI Industrial Automation (ATI) certifies that the above product was calibrated in accordance with applicable ATI procedures. These procedures are compliant with the ISO 9001 standard to ensure that the above product is within ATI specifications. To meet this level of accuracy any loads must be correctly aligned to the transducer origin and the transducer must be mounted to a sufficiently strong surface.

To ensure the transducer measurement uncertainties listed on page 1 are met, more-conservative limits are used as calibration targets during the calibration process. If any of the calibration targets are exceeded the calibration will not pass. It is possible for a transducer to exceed these calibration targets while meeting the page 1 measurement uncertainties. The following calibration targets were used for this transducer:  $F_x=1.00\%$ ,  $F_y=1.00\%$ ,  $F_z=0.75\%$ ,  $T_x=0.75\%$ ,  $T_y=1.00\%$ , and  $T_z=1.00\%$ .

Note: If this is a recalibration of a legacy transducer that does not have precision locating features (such as dowel holes), there could be additional error in  $T_x$  and  $T_y$  due to inexact mounting location. Precision locating features are highly recommended for best accuracy and can be added by ATI.

This certificate shall not be reproduced except in full without written approval from ATI. This certificate only applies to the items listed and does not include unlisted ancillary items such as data acquisition equipment.

For questions or comments, please contact your ATI representative.





**Calibration Accuracy Section**  
**Sensor System FT25156, Delta/SI-660-60**  
**Force units: N; Torque units: N-m**

<b>Calibrated Ranges (±)</b>					
<b>Fx</b>	<b>Fy</b>	<b>Fz</b>	<b>Tx</b>	<b>Ty</b>	<b>Tz</b>
660	660	1980	60	60	60

<b>Applied Loads</b>						
	<b>Fx</b>	<b>Fy</b>	<b>Fz</b>	<b>Tx</b>	<b>Ty</b>	<b>Tz</b>
1	0.000	422.581	0.000	-48.357	0.000	0.000
2	-422.581	0.000	0.000	0.000	-48.357	0.000
3	0.000	-422.581	0.000	48.357	0.000	0.000
4	422.581	0.000	0.000	0.000	48.357	0.000
5	0.000	556.028	0.000	-21.262	0.000	0.000
6	-556.028	0.000	0.000	0.000	-21.262	0.000
7	0.000	-556.028	0.000	21.262	0.000	0.000
8	556.028	0.000	0.000	0.000	21.262	0.000
9	0.000	311.376	0.000	-3.954	0.000	-35.604
10	0.000	311.376	0.000	-3.954	0.000	35.615
11	-311.376	0.000	0.000	0.000	-3.954	-35.602
12	-311.376	0.000	0.000	0.000	-3.954	35.612
13	0.000	-311.376	0.000	3.954	0.000	-35.615
14	0.000	-311.376	0.000	3.954	0.000	35.604
15	311.376	0.000	0.000	0.000	3.954	-35.612
16	311.376	0.000	0.000	0.000	3.954	35.602
17	0.000	0.000	556.028	-42.390	0.000	0.000
18	0.000	0.000	556.028	0.000	-42.417	0.000
19	0.000	0.000	556.028	42.412	0.000	0.000
20	0.000	0.000	556.028	0.000	42.399	0.000
21	0.000	0.000	1112.055	0.000	0.000	0.000
22	0.000	0.000	-1112.055	0.000	0.000	0.000
23	0.000	0.000	-556.028	42.390	0.000	0.000
24	0.000	0.000	-556.028	0.000	42.417	0.000
25	0.000	0.000	-556.028	-42.412	0.000	0.000
26	0.000	0.000	-556.028	0.000	-42.399	0.000

*Refer to page 6 for important information on regarding this report.*



Full-Scale Error						
	Fx	Fy	Fz	Tx	Ty	Tz
1	-0.32%	0.11%	0.13%	0.08%	0.09%	0.09%
2	0.12%	0.03%	0.23%	-0.21%	0.10%	-0.08%
3	-0.13%	-0.05%	0.11%	-0.08%	-0.02%	0.07%
4	0.25%	-0.06%	0.14%	0.21%	-0.07%	-0.01%
5	-0.09%	0.15%	0.10%	-0.09%	-0.03%	0.06%
6	-0.06%	-0.14%	0.12%	0.11%	-0.04%	0.12%
7	-0.28%	-0.04%	0.11%	0.04%	-0.01%	0.07%
8	0.17%	-0.14%	0.18%	-0.03%	0.03%	0.07%
9	-0.08%	-0.19%	0.03%	0.02%	0.03%	-0.03%
10	-0.20%	-0.09%	0.01%	-0.03%	0.00%	-0.04%
11	0.06%	-0.03%	0.03%	-0.02%	-0.03%	-0.05%
12	0.12%	-0.07%	0.04%	0.19%	-0.04%	-0.01%
13	-0.17%	0.03%	0.01%	-0.04%	0.04%	-0.04%
14	-0.02%	0.24%	0.04%	0.02%	0.11%	-0.03%
15	-0.17%	0.14%	0.05%	0.03%	0.03%	0.00%
16	-0.25%	-0.14%	0.04%	-0.19%	0.00%	-0.06%
17	0.13%	-0.09%	-0.06%	0.08%	0.28%	0.16%
18	0.28%	0.09%	-0.07%	0.00%	-0.03%	0.08%
19	-0.13%	0.13%	-0.05%	-0.02%	0.24%	0.08%
20	0.15%	0.07%	0.02%	-0.18%	0.03%	-0.11%
21	-0.18%	0.13%	0.14%	0.03%	-0.25%	-0.09%
22	0.08%	0.05%	-0.10%	-0.10%	-0.03%	-0.09%
23	0.01%	0.09%	0.08%	0.10%	0.05%	0.09%
24	-0.02%	0.13%	0.07%	-0.03%	0.11%	0.13%
25	-0.13%	0.10%	0.07%	-0.12%	-0.10%	-0.01%
26	0.05%	0.02%	0.09%	0.18%	0.02%	0.00%

Refer to page 6 for important information on regarding this report.

Offset Report						
	Fx	Fy	Fz	Tx	Ty	Tz
F/T Offset	-0.0909	0.0241	-0.3425	0.0044	0.0087	-0.0035
	<b>SG0</b>	<b>SG1</b>	<b>SG2</b>	<b>SG3</b>	<b>SG4</b>	<b>SG5</b>
SG Offset	0.2600	2.1200	6.6050	2.6900	3.1750	1.2850
±SG Limit	1599.9638	1599.9638	1599.9638	1599.9638	1599.9638	1599.9638

Offsets are measured in a unique configuration not available to the user.

Refer to page 6 for important information on regarding this report.



<b>Gain-Check Report</b>						
	<b>SG0</b>	<b>SG1</b>	<b>SG2</b>	<b>SG3</b>	<b>SG4</b>	<b>SG5</b>
<i>Lower Limit</i>	<i>0.7000</i>	<i>0.7000</i>	<i>0.7000</i>	<i>0.7000</i>	<i>0.7000</i>	<i>0.7000</i>
<i>Lower Output</i>	<i>0.8526</i>	<i>0.8318</i>	<i>0.8418</i>	<i>0.8985</i>	<i>0.8398</i>	<i>0.8018</i>
<i>Upper Output</i>	<i>0.8539</i>	<i>0.8388</i>	<i>0.8460</i>	<i>0.9048</i>	<i>0.8410</i>	<i>0.8086</i>
<i>Upper Limit</i>	<i>1.0000</i>	<i>1.0000</i>	<i>1.0000</i>	<i>1.0000</i>	<i>1.0000</i>	<i>1.0000</i>

*Gain readings are measured in a unique loading configuration.*



As part of our commitment to quality, each ATI force/torque transducer undergoes rigorous accuracy testing. This process, which involves applying and verifying a rich set of loading cases designed to cover the transducer's entire six-axis calibrated range, is designed to ensure that your transducer meets the measurement uncertainties listed in this Certificate of Calibration.

Our transducers often exceed our quality standards for accuracy. Often, transducers perform exceptionally well in certain loading situations. This report summarizes the performance of your ATI F/T transducer in our factory tests. It can be thought of as a 'best-case scenario' snapshot of your transducer's performance under laboratory conditions, in a variety of loading situations. You can expect the accuracy of your transducer measurements to fall somewhere between its performance during testing and the measurement uncertainties listed on its calibration certificate.

The *Calibration Accuracy Section* contains several tables of data. The *Calibrated Ranges ( $\pm$ )* table lists the transducer's rated range for each axis. The *Applied Loads* table lists the loads applied during calibration and testing. The *Full-Scale Error* table shows the sensor system's measurement error as a percentage of full scale for each axis in each loading case. The *Offset Report* table, if included, shows transducer readings during offset adjustment and associated control limits. And the *Gain-Check Report* table, if included, shows verification of the transducer's sensitivity and associated control limits. If included, the *Before and After Report* table shows a loading case relating the transducer's performance as received to its performance after recalibration.

*For best accuracy, be sure to use your transducer's precision location features, and mount your transducer to a stiff surface.* If an ongoing guarantee of sensor accuracy is important to you, we recommend that your sensor be tested annually. Contact your ATI Industrial Automation distributor to schedule recalibrations.

# IRB 6620

## Industrial Robot



The IRB 6620 is a flexible and agile robot with a large working envelope. The robot combines four robots in one: floor-standing, tilted or inverted mounted and shelf capability. This compact robot opens up opportunities for new flexible and improved line concepts. Functions that all help saving floor space, creating higher robot density and shorter lines.

### **The agile large robot**

This relatively small and compact robot is easy to install thanks to its low weight. In spite of its small size neither the handling capacity nor the reachability are affected. The robot's compactness and shelf capability also make it ideally suited for applications where robots are placed on different levels. The shelf mounted robot may in this case be tilted to increase space between the two levels.

With the IRB 6620 you get a flexible and cost efficient solution that generates high uptime and lower production costs.

### **Machine tending**

IRB 6620 is suited for machine tending applications such as Die Casting and Injection Molding, that requires minimised extraction time. The main benefit is the huge working range downwards making the robot ideal to mount on top of a machine working downward to save costly floor space. Another benefit in different material handling also saving floor space applications is the IRB 6620 ability to work inverted mounted. In all these configuration the installation will benefit from the very low weight of the robot.

### **Main Applications**

- Machine Tending
- Material Handling
- Spot Welding

## Specification

Robot version	Reach (m)	Payload (kg)*
IRB 6620	2.2	150
Number of axes	6	
Protection	IP 54, IP 67 with Foundry Plus 2 option	
Mounting	Floor, titled, inverted	
Controller	IRC5 Single Cabinet	

\* Extra loads can be 50 kg on to the upper and 100 kg mounted on to the robot base.

## Performance (according to ISO 9283)

	Position repeatability	Path repeatability
IRB 6620	0.03 mm	0.62 mm

Actual values are equal or below the given value.

## Technical information

### Electrical Connections

Supply voltage 200 - 600 V, 50/60 Hz

### Physical

Robot base 1007 x 760 mm

Robot weight 900 kg

### Environment

Ambient temperature for mechanical unit

During operation +5° C (41° F) to + 45° C (113° F)

During transportation and storage -25° C (-13° F) to +55° C (131° F)

During short periods (max. 24 h) up to +70° C (158° F)

Relative humidity Max. 95%

Safety Double circuits with supervisions, emergency stops and safety functions. 3-position enable device

Emission EMC/EMI shielded

Option Foundry Plus

- UL approved
- Rigid design with collision resistance
- Prolonged wrist with smooth surfaces for dress pack life time
- Large centre hole for welding cables in the foot
- Easy change of dress pack
- Weld spatter protection on gear sealing
- 50 kg load capacity on rear upper arm for dress pack

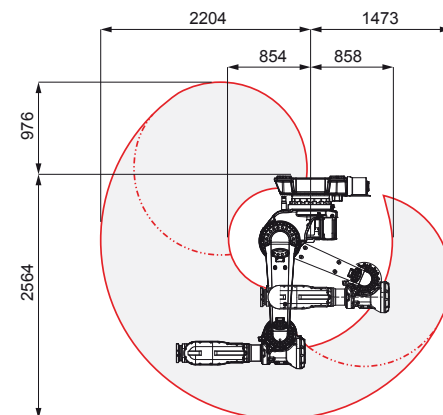
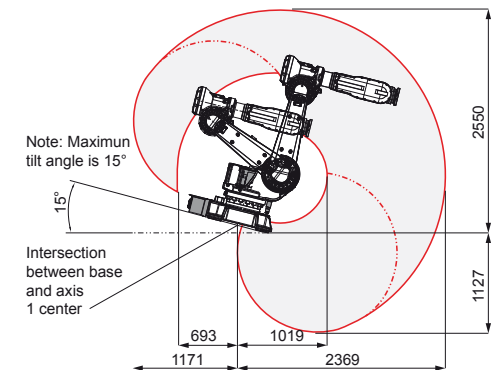
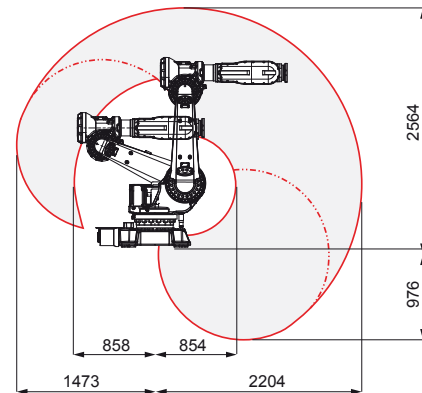
Data and dimensions may be changed without notice.

## Movement

Axis movement	Working range	Axis max. speed
Axis 1 Rotation	+170° to -170°	100°/s
Axis 2 Arm	+140° to -65°	90°/s
Axis 3 Arm	+70° to -180°	90°/s
Axis 4 Wrist	+300° to -300°	150°/s
Axis 5 Bend	+130° to -130°*	120°/s
Axis 6 * Turn	Default: +300° to -300° Max. rev: ± 96	190°/s

A supervision function prevents overheating in applications with intensive and frequent movements. \*Limitations with DressPack.

## Working range



Vedlegg C

Poster presentation

# Implementation of acceleration estimator to increase load data accuracy for a robot based, testing system for CPR-manikins

Author: Anders Espedal

## Introduction

CPR-training manikins are essential tools in education and training of both medical and non-medical personnel in order to perform correct Cardiopulmonary Resuscitation (CPR). To ensure the quality and longevity of these products, they are put through extensive testing regiments. Laerdal employs the assistance of several tools and machines to perform this testing. These machines must be able to emulate human-like compression movements in order to secure that the products are tested in a manner that is as similar as possible to what they would experience during thousands of compressions exerted by trained and untrained personnel upon them.

## Problem statement

One such machine is an ABB IRB 6620 Industrial Robot. The robot has been used for a wide variety of testing but is much used for compression testing of CPR-manikins. Data collection during testing is an important resource for product developers to evaluate product performance and longevity, and thus accurate data is of high importance. The robot has a load cell mounted onto the end-effector for measurement of compression force. The force data gathered has been suspected to be affected by the mass of the compression tool mounted below the load cell. The main aim of this thesis is to develop a compensation scheme which can eliminate the disturbance caused by the tool mass. Additionally, if the accuracy improvement is good enough it is also desirable to use the robot as a calibration instrument for product specific force sensors.







**Laerdal**  
helping save lives

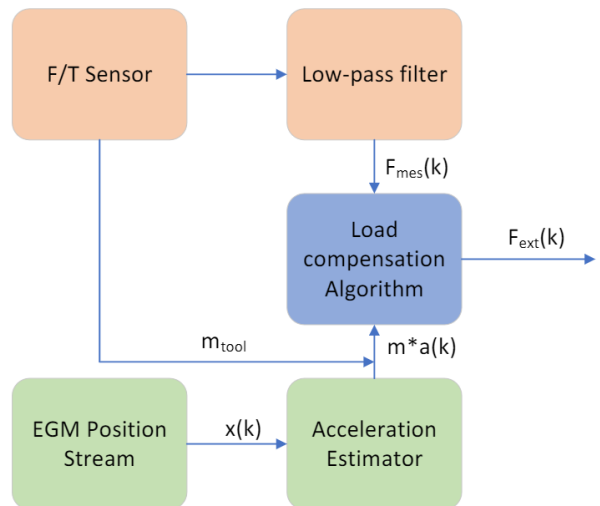


## Method

The compensation scheme is static measurement of the tool mass, and acceleration estimation based on position measurement to estimate the force disturbance caused by the tool mass during testing. Four acceleration estimators are implemented into the compensation scheme and compared:

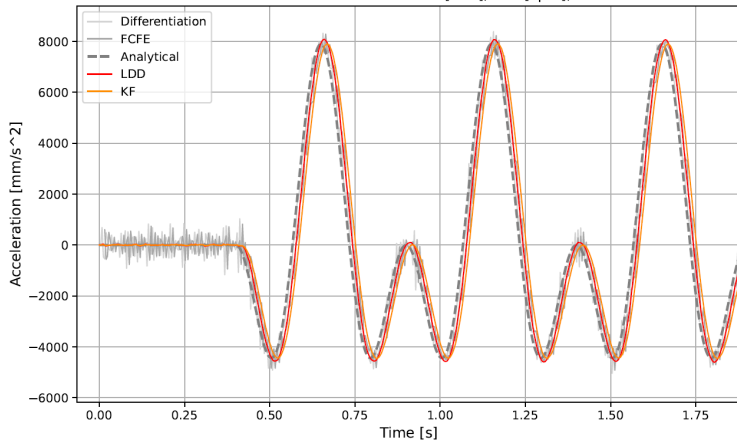
- **Second order backwards finite difference**
- **Linear double differentiator**
- **Fast Curve Fitting Estimator**
- **Kalman Filter**

Additionally due to the high load environment the robot is used in results in a significant position error due to compliance. To measure this effect and to be able to calculate the reference load during testing, an interferometer measuring device is used.



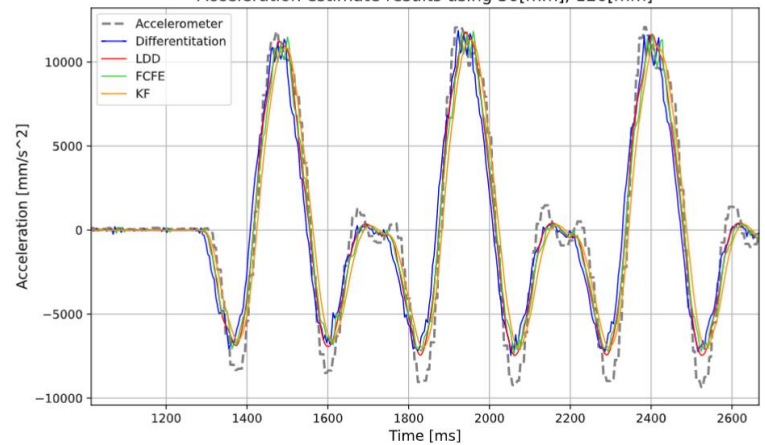
## Results

Acceleration simulation result at 50 [mm], 120 [cpm], SNR: 80



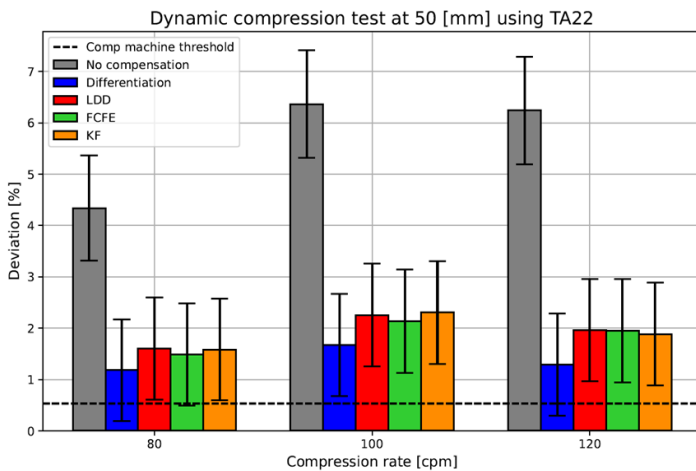
**Estimation output based on position generated by the theoretical expression for the motion profile of a human chest compression. The estimators are compared to the analytical double derivative of the position function.**

Acceleration estimate results using 50[mm], 120[mm]

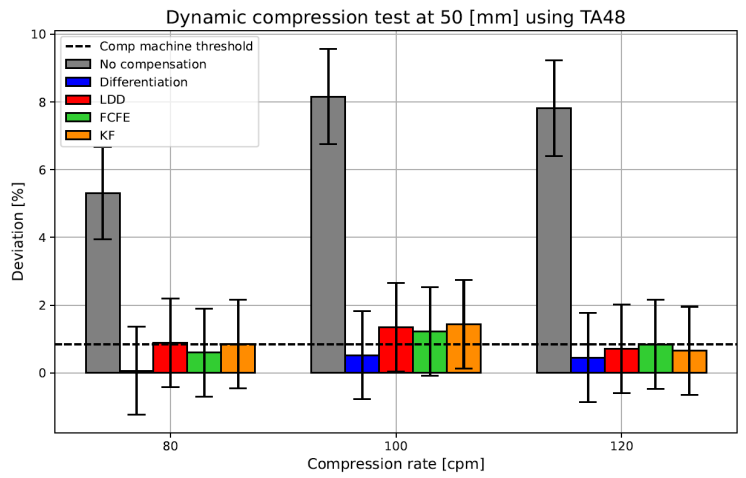


**Estimation output based on position data from the robot. The estimates are compared to data from an accelerometer mounted on the robot.**

Tests involve simulation and testing of the acceleration estimator alone, and as part of the compensation scheme. For the compensation results, the output is compared to a static reference value based on the spring rate of the compression spring used in the test and the depth measured by the interferometer. The results presented are the average peak compression force deviations from the reference. All estimators show significant improvement in peak compression force deviation. The results are also compared to the accuracy of another machine that is currently used as a calibration instrument for product sensors shown as a dashed line on the deviation plots. The stiffer spring shows a larger error due to the increase in compliance error. This error negatively affects the accuracy of the robot's position measurement, which in turn affects the acceleration estimate.



**Results showing deviation from reference for each estimator, compared to the output without compensation. The spring TA22 is the stiffest spring used in testing and is stiffer than a typical test object (manikin).**



**Results showing deviation from reference using a different spring (TA48). This spring has a spring rate that is more typical for actual test objects.**

## Conclusion

Although the results show significant improvement in accuracy, they are not conclusive as to whether the robot can be used as a calibration instrument. The robot as a mechanical system displays lack of stiffness in order to achieve the required accuracy for this application. Further work in improvement of the compensation scheme and on the dynamic compensation of position compliance error is required to make this a feasible solution.