ISSN 0280-5316
ISRN LUTFD2/TFRT--5889--SE

Prevention of Servo-Induced Vibrations in Robotics

Peter Larsson

Department of Automatic Control
Lund University
August 2011

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
August 2011

Document Number
ISRN LUTFD2/TFRT--5889--SE

Author(s)
Peter Larsson

Supervisor
Anders Robertsson Automatic Control Lund, Sweden
Peter Valdt Bernecker & Rainer Malmö, Sweden
Tore Hägglund Automatic Control Lund, Sweden (Examiner)

Sponsoring organization

Title and subtitle
Prevention of Servo-Induced Vibrations in Robotics. (Eliminering av vibrationer i industrirobotar)

Abstract
This thesis describes methods for treating servo-induced vibrations in robotics. The focus lies on portal robots (also called linear robots) that partially consist of hardware manufactured by Bernecker & Rainer Industrie-Elektronik Ges.m.b.H. It is of interest in many industrial applications that there is some kind of vibration treatment in the control strategy. The main reason for this is that it can be time consuming having to wait for transient vibrations to fade. Another reason is that mechanical wear increases if there are vibrations present in the application. Emphasis lies on how to prevent servo-induced vibrations, however, active vibration damping through simple model based feedback is also briefly investigated. A comparison is also conducted between the derived preventive solution and an already existing one in the servo drive systems at hand.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
100

Recipient's notes

Security classification

Till pappa

Acknowledgements

I am grateful to many people who have been helpful throughout the work with this thesis. Really hoping not to leave anyone of them unmentioned, I would like to start by thanking Peter Valdt for dedicating a lot of his time to my work. He has been both good company and has helped me with many technical issues concerning both hardware and software. We have also had many fruitful discussions about possible solutions to problems that were faced. I would also like to thank Patric Thysell for lots of good advice and help concerning implementation and technical possibilities. At the department of Automatic Control, I would especially like to thank my examiner Tore Hägglund and my advisor Anders Robertsson who have answered questions and given me guidance towards the right direction. I would also like to thank them for the time that they have devoted to several meetings. Moreover, I would like to thank Carl Windfeldt for helping me with software and Fredrik Holmberg for help concerning experimental hardware. Thanks goes to everyone at the B&R office in Malmö for the nice company.

The support of my family should not be left unmentioned. My father always inspired me and gave me my interest for subjects that are part of this work.

And Sara, thank you for keeping my spirit up when things were tough.

Contents

1. Introduction

This thesis concerns the subject of servo induced vibrations in external processes. The work has been conducted with both B&R software and hardware. External processes have been a portal robot built by a business partner to B&R and an experimental setup built in the B&R office in Malmö.

1.1 Background

Bernecker & Rainer is a multinational company delivering hardware and software solutions for industrial automation. Among other hardware products they produce are Programmable Logic Controllers, I/O modules, industrial PCs, servo motors, servo drive systems etc.

In this project however, servo motors, servo drive systems and PLCs have constituted the main hardware components for research (i.e., besides the external processes). All control algorithms evaluated in this project were created in B&R Automation Studio.

The states of a servo motor are well known through current and encoder measurements. Driving an external process with a servo motor, it is often the case that the external process states can be considered more or less proportional to those of the servo motor (e.g. acceleration, speed and position). The assumption that is made in those cases is that the external process consists of rigid parts. In reality the parts are more or less rigid and phenomenas like vibration can arise. A general strategy to cope with vibrations does already exist in the servo drive systems used for this project. The strategy is based on making reference trajectories softer by means of reshaping acceleration set points. This strategy is also used to prevent unnecessary material wear.

The objective for this thesis has been to find a method that performs better than the already existing one. At first the demand was to create an active vibration damping strategy. This was however found to be in conflict with more important criteria; no additional sensors should be involved in the solution and the solution should be applicable for a range of different processes. Thus we were led to a scenario where focus was turned to a method of how to prevent vibrations instead of treating vibrations already induced. This was found to adapt better to the more important criteria. Experiments and evaluations were performed on two processes for mainly one reason, to indicate that the solution was applicable on different processes.

1.2 Overview

This thesis is separated into three parts. The first part contains an initial study of the processes and the software tools. As construction of the experimental setup and familiarization with the process hardware was given a lot of attention at the initial stage, Chapter 2 is dedicated to hardware descriptions. Chapter 3 gives a brief description of Automation Studio so that the reader gets an understanding of how implementation of e.g. control strategies was conducted. Chapter 4 is dedicated to basic motor control and to the controller implemented in the servo drive systems at hand. The second part contains studies of the vibration phenomenon in the processes and descriptions of possible solutions. Studies of the vibration phenomenon are conducted in Chapter 5 and possible control solutions

are presented in Chapter 6. The last part deals with implementation issues in Chapter 7 and a presentation of final results is given in Chapter 8. Conclusions based on the work and the results are given in Chapter 9 as well as suggestions of future work.

If the reader is familiar with industrial servo control systems and PLCs, he or she can refrain from reading Section 2.3 to Section 4.2 and still be able to grasp the ideas of this thesis.

2. Process hardware

In this chapter hardware used for experiments and control implementation is presented. Two processes have been used, a process that was built solely for this particular project and an industrial portal robot.

Figure 2.1. *Shows the portal robot to the left and the motion control cabinet to the right.*

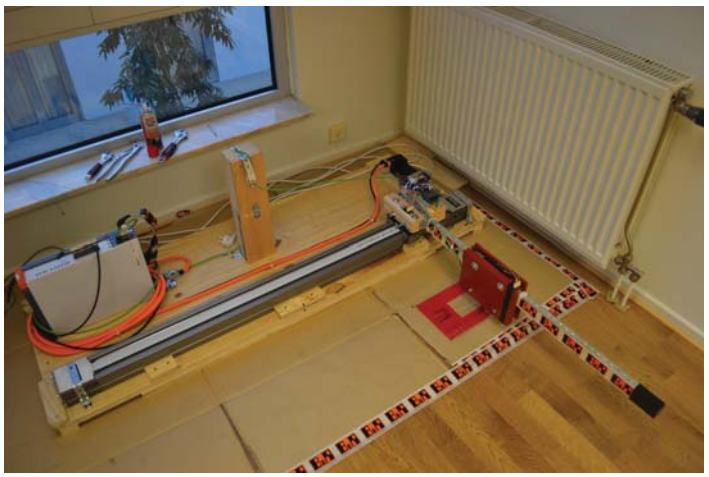

Figure 2.2. *The 1 DOF experimental setup.*

The experimental process was built at the B&R office in Malmö and the intention with its structure was to resemble two beams that constitute arms of an industrial portal robot (the term portal robot is sometimes referred to the term linear robot). Portal robots are built by many different manufacturers, different types and sizes are found for different applications. The main mechanics of portal robots consist of three beams or arms, two constitute what resembles a Cartesian coordinate system and a third arm is positioned orthogonally with respect to the two first mentioned arms. In other words three planes are spanned which allows three degrees of freedom. Definitions of coordinate systems may vary between different applications. The experimental setup allows one degree of freedom since translation is only possible along one arm. This proves to be enough since movements along each arm on a linear robot are controlled by one servo

respectively, i.e., the control methods for one arm can hopefully be generalized for all three and thus only one arm needs to be considered for experiments. Figures 2.1 and 2.2 show the portal robot and the experimental setup, both located at the B&R office in Malmö. What the portal robot and the experimental process have in common (in terms of hardware components) is that they both consist of parts manufactured by B&R; servo, servo drive system (ACOPOS), PLC (programmable logic controller), I/O modules, power supplies etc. During the project some of the hardware (e.g. PLC types) have been exchanged with other types within the B&R product family; this will be stated if it is of particular interest.

2.1 The portal robot

The mechanics of the portal robot used for experiments in this project was built by a business partner to B&R. It is a medium size, three axis linear robot with three servo motors mounted to belts via gearboxes. Each servo is coupled to a servo drive system; B&R ACOPOS 1016:s in the current configuration (see Figure 2.5). The servo drives are connected through a Powerlink interface to a PLC, both a B&R Power Panel 420 and a B&R X20 CP 1485 have been used. The PLC is connected to a PC with B&R Automation Studio via Ethernet. I/O modules are used for accelerometers that are mounted on critical spots of the robot. The I/O modules are connected to the PLC and a power supply besides the accelerometers according to Figure 2.3.

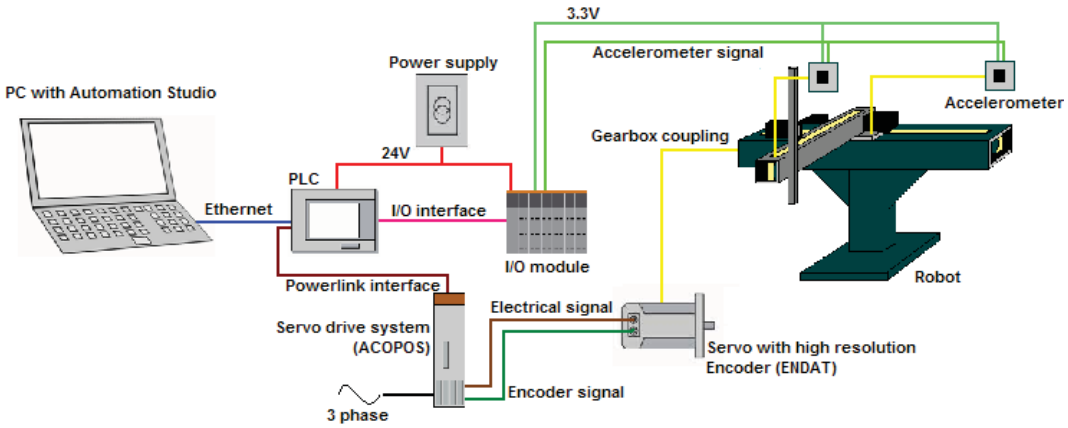

Figure 2.3. *An overview of the involved hardware in the portal robot. The servos are coupled via gearboxes to the mechanics. On the signal end is the power signal and the encoder signal to the servo drive system (ACOPOS). The three servo drives can be powered with three phase 400VAC, single phase 230VAC or 24VDC signals. The PLC is connected to the servo drives via Powerlink and to a PC via Ethernet, moreover the PLC is connected to its 24VDC power supply and to I/O modules used for accelerometers.*

The robot arms will be referred to as the X-axis, Y-axis and Z-axis according to Figure 2.4. In industry there is always a tool mounted on one side of the Y-axis. This tool may for instance be an expensive grip tool that carries a load from a processing machine to a conveyor. No such tool has been available for this project, thus this has been replaced by a 5*kg* weight.

The overall dynamics of portal robots can be considered as quite complex, there are many sources that can implicate undesired dynamics such as oscillations. Generally it can be concluded that as little as possible of those dynamics are desired at the mounting of the tool.

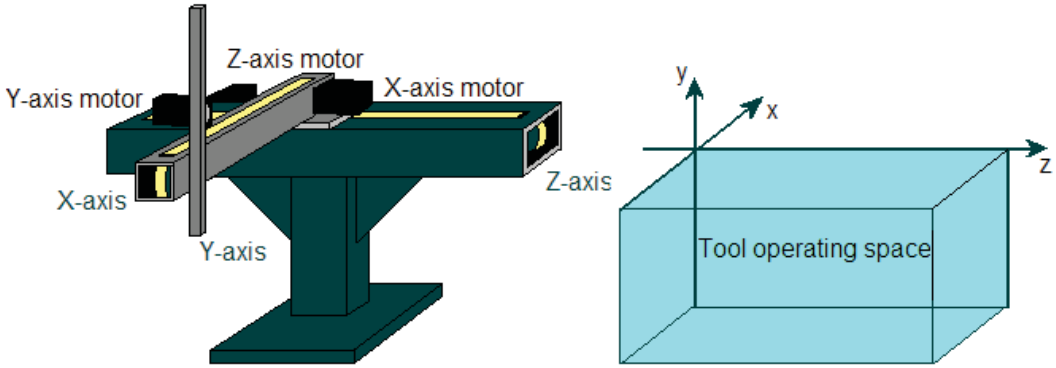

Figure 2.4. *A schematic illustration of the portal robot and the operating space for a tool mounted on the negative end of the Y-axis.*

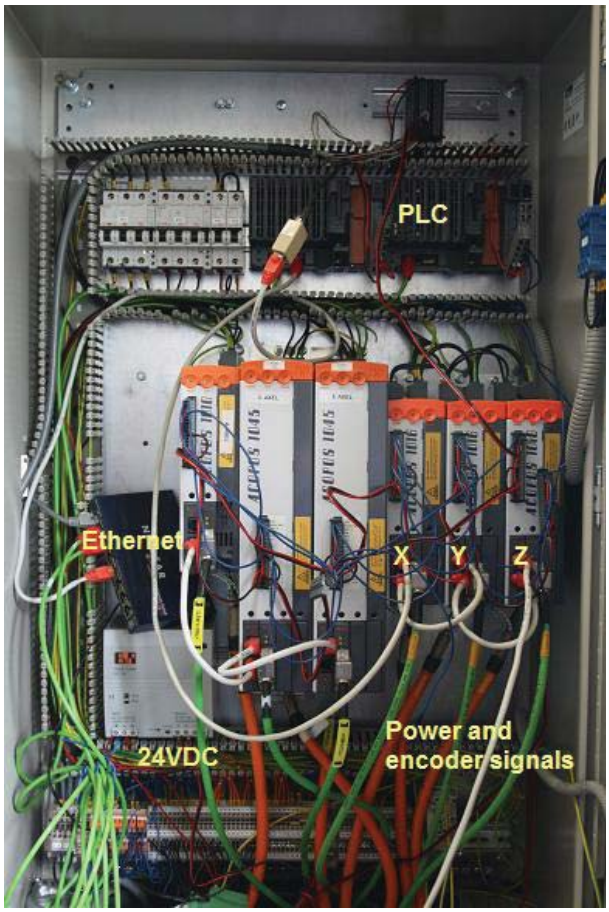

Figure 2.5. *The control cabinet of the portal robot. Some of the main components are marked, the three servo drives to the left are inactivated.*

Undesired oscillations can originate from e.g. belts, worn gears and, primarily,

bending of beams that constitute the arms. Belts and pulleys are replaced by racks and pinions or lead screws for some portal robots on the market. The rack has to be fine-toothed to get a nice motion but problems with noise and wear are quite common for such designs. In Figure 2.3 two accelerometers are illustrated, these are not intended to be used in an industrial design. The intention with the accelerometers is only to obtain and to verify experimental measurements. In an industrial design no additional sensors are desired, this means that the available signals for a possible control strategy are e.g. servo encoder and stator current signals.

2.1.1 Tool point location in space

As already indicated, in most industrial applications the movements of the tool is of big interest. The tool is often more brittle than the rest of the structure and also quite expensive in many cases. It is thus important that the positioning of the tool is accurate. For a portal robot a cuboid constitutes the operating space of the tool (see Figure 2.4). The length of the arms determine the length, width and depth of the cuboid.

On each arm a servo mounted via a single gear to a pulley and a belt roughly determines movements of the tool in one dimension of space. The tool position $\boldsymbol{P}$ can ideally be expressed as Equation 2.1

$$\boldsymbol{P} = (x, y, z) \qquad (2.1)$$

where x, y and z are coordinates as in Figure 2.4. Due to the nature of the mechanical structure only negative x, y and positive z values can be taken by $\boldsymbol{P}$. Throughout this project one revolution of the motor axis is divided into 36000*units*, this is however an adjustable setting. Table 2.1 contains gearbox and pulley data for each axis.

	Gearing g (motor:pulley)	Pulley diameter d (m)
X-axis	$10:1$	$89{,}13 \cdot 10^{-3}$
Y-axis	$7:1$	$89{,}13 \cdot 10^{-3}$
Z-axis	$10:1$	$101{,}86 \cdot 10^{-3}$

Table 2.1. *Gearbox and pulley data.*

A conversion constant k between *units* and m for each axis is calculated according to Equations 2.2-2.4 with the contents in Table 2.1.

$$k_x = \frac{2\pi d_x}{36000\, g_x} = \frac{2 \cdot 89{,}13 \cdot 10^{-3}\pi}{36000 \cdot 10}\, m/units = 1{,}56 \cdot 10^{-6}\, m/units \qquad (2.2)$$

$$k_y = \frac{2\pi d_y}{36000\, g_y} = \frac{2 \cdot 89{,}13 \cdot 10^{-3}\pi}{36000 \cdot 7}\, m/units = 2{,}22 \cdot 10^{-6}\, m/units \qquad (2.3)$$

$$k_z = \frac{2\pi d_z}{36000\, g_z} = \frac{2 \cdot 101{,}86 \cdot 10^{-3}\pi}{36000 \cdot 10}\, m/units = 1{,}78 \cdot 10^{-6}\, m/units \qquad (2.4)$$

where g is the gearing and d is the pulley diameter. Equation 2.1 can then be expressed as Equation 2.5

$$\boldsymbol{P} = (k_x s_x, k_y s_y, k_z s_z) \qquad (2.5)$$

where s indicates that a coordinate is given as the motor axis angle. In Equation 2.5 it is assumed that the position of the tool mounted on the negative end of the Y-axis can be expressed by the servo angles. Equation 2.5 applies for ideally rigid structures only, or when there are no movements of the arms. On a real process there is flexing and bending of arms, elongation and stretching of belts, cross couplings etc. that give rise to a distortion $\boldsymbol{P}_{dist} = (x_{dist}, y_{dist}, z_{dist})$ on the position of the tool. Depending on the control parameters in the servo drive the mechanical distortion phenomenas are more or less observed in the motor axis angle. A more realistic description of the tool position is expressed by Equation 2.6.

$$\boldsymbol{P}_{tool} = \boldsymbol{P} + \boldsymbol{P}_{dist} = (k_x s_x + x_{dist}, k_y s_y + y_{dist}, k_z s_z + z_{dist}) \qquad (2.6).$$

Roughly, it can be said that the main objective here is to minimize the distortions so that the description of the tool position gets as close as possible to Equation 2.5 using a stiffly tuned controller for the servo.

2.2 Hardware for experiments

The experimental setup in Figure 2.2 has been dimensioned and assembled at the B&R office in Malmö merely for this project. It is built to resemble a 2-DOF linear robot, however only one axis is coupled to a servo. A Parker LCB060 linear unit constitutes the dynamical arm. The operating range of LCB060 is $1m$. As for the portal robot described in Section 2.1, the linear unit has a belt pulley design. The tracks for the translation unit are not lubricated at delivery, i.e., friction is quite high. High friction means high damping at $x = 0$, this is not desired since propagation of arm oscillations to the belt and in turn to the servo is required for experiments. To reduce problems with high friction the tracks have been sprayed with silicone. Silicone might not be the optimal lubricant but using ordinary machine grease would harm the plastic contact surfaces on the translation unit.

The servo is attached to the LCB060 via a single gear gearbox and a clamp coupling. A second arm, the X-axis (see Figure 2.7), is mounted on the translation unit. The X-axis is made of a wooden plank and a metal strip is attached along two sides to increase the stiffness. A weight is mounted so that its position along

the X-axis is adjustable. The experimental setup allows one degree of freedom with the servo but since the weight position along the plank can be varied manually, two degrees of freedom are present. However for experiments this is irrelevant, only one degree of freedom can be considered. It is the oscillations from the X-axis that are induced by movements from the Z-axis that are of interest.

Accelerometers are attached on the translation unit and on top of the weight. The servo is connected to a servo drive system (a B&R ACOPOS 1045, see Figure 2.8) via Powerlink. The servo drive is in turn connected to a PLC, a B&R X20 CP 1485, which is connected to I/O modules for the accelerometers and via Ethernet to a PC with B&R Automation Studio. The setup for the experimental hardware is identical to the setup described in Section 2.1. Figure 2.6 shows the connections between the main components.

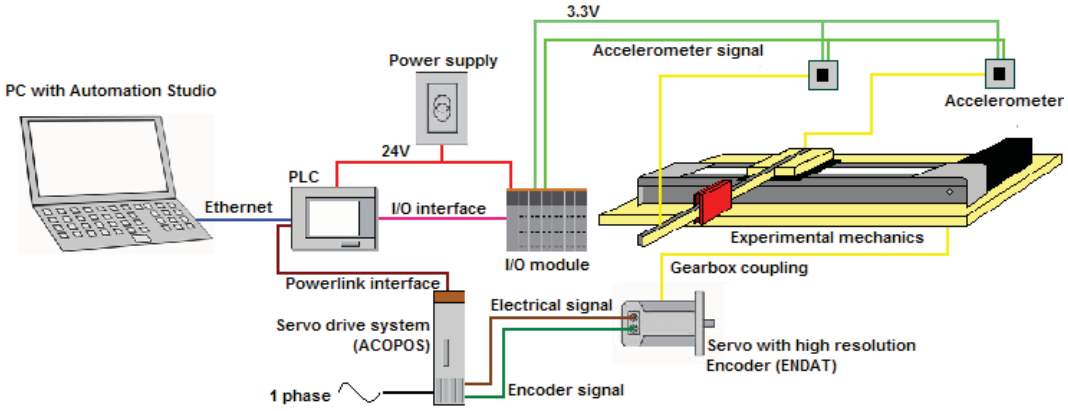

Figure 2.6. *An overview of the involved hardware for the experimental setup. The servo is coupled via a gearbox to the mechanics. On the signal end is the power signal and the encoder signal to the servo drive system (ACOPOS). The servo drive is powered with a single phase 230VAC signal in this configuration. The PLC is connected to the servo drive via Powerlink and to a PC via Ethernet, moreover the PLC is connected to its 24VDC power supply and to I/O modules used for accelerometers.*

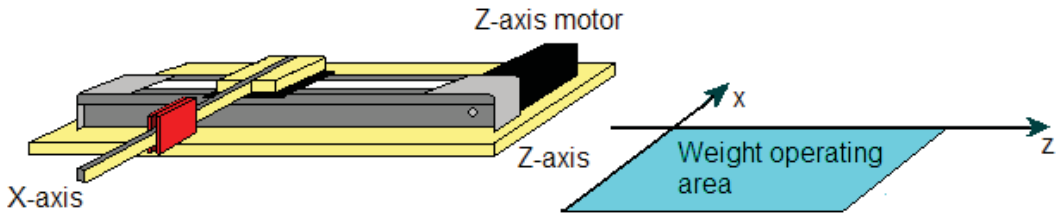

Figure 2.7. *A schematic illustration of the experimental setup and the operating area of the weight.*

Figure 2.7 illustrates the experimental setup and the Cartesian coordinate system that is spanned for the weight. In Figure 2.8 the “control unit” is shown, some of the main components can also refer to Figure 2.6. Figure 2.9 shows an accelerometer mounted on the arm weight.

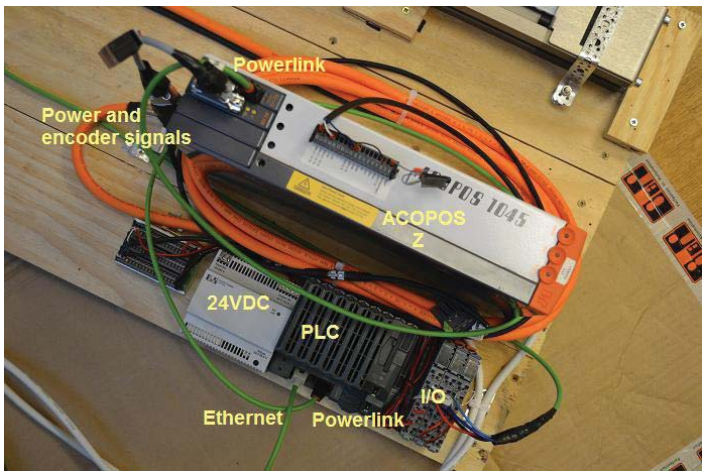

Figure 2.8. *The "control unit" of the experimental setup. The main components are marked.*

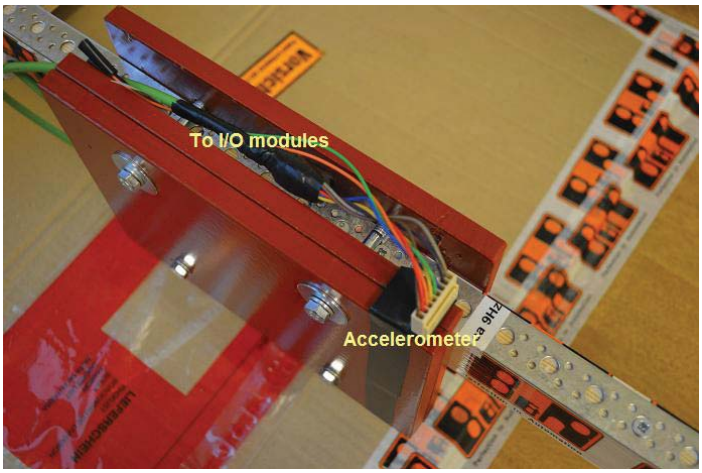

Figure 2.9. *Shows the weight (red) and an accelerometer attached to it.*

2.2.1 Weight location in the plane

The intention with the experimental setup is as already stated to resemble a 2-DOF linear robot where the X-axis is passive and the Z-axis is active. The weight can be said to emulate a Y-axis. However this is not entirely true, the Y-axis on a real portal robot induces oscillations at its natural frequency that propagate to the origin. Y-axis oscillations are perhaps the most critical ones for a portal robot. However, if vibrations on a critical spot can be transferred back to the servo from where they were induced it does not matter were the critical spot is located. Therefore the design of the experimental setup is fully sufficient. As it is possible to change the weight position along the X-axis it is also possible to change the characteristics of the oscillations.

Following the same discussion as in Subsection 2.1.1 the weight position can be described by Equation 2.7 where the cuboid is reduced to a plane. The X-axis value is represented by a constant *a*.

$$\boldsymbol{P}_{weight} = (a + x_{dist}, c_z s_z + z_{dist}) \qquad (2.7)$$

where c_z is the factor between *units* for the motor axis angle and *m* on the Z-axis. The distortions indicated by the subscript *dist* arise mainly from bending of the arm but perhaps also from elongation and stretching of the belt. It is only of interest to consider z_{dist} for the experimental setup. This is because x_{dist} is coupled to z_{dist}, i.e., if z_{dist} is minimized x_{dist} will also be minimized. The discussion above is valid because only one servo is active. When considering a portal robot where two more servos can be active at the same time there are additional terms that may have to be accounted for. The factor c_z is calculated with the contents in Table 2.2 according to Equation 2.8.

	Gearing *g* (motor:pulley)	Pulley diameter *d* (*m*)
Z-axis	5:1	$54{,}11 \cdot 10^{-3}$

Table 2.2. *Gearbox and pulley data.*

$$c_z = \frac{2\pi d_z}{36000\, g_z} = \frac{2 \cdot 54{,}11 \cdot 10^{-3}\pi}{36000 \cdot 5}\, m/units = 1{,}89 \cdot 10^{-6}\, m/units \qquad (2.8)$$

2.3 Servo motors

The servo motors that have been involved in this project are all three phase synchronous motors equipped with high resolution encoders. Table 2.3 shows the motor type mounted on each axis for the portal robot and the experimental setup.

	X-axis	Y-axis	Z-axis
Portal robot	B&R 8MSA3M.E3	B&R 8MSA3M.E3	B&R 8LSA35.E3
Experimental setup	-	-	B&R 8LVA22.B1

Table 2.3. *The three phase synchronous motors mounted on each axis on the portal robot and the experimental setup.*

The characteristics of the motors differ between the types; e.g. torque/speed ratios, maximum rated currents etc. Synchronous three phase motors belong to the category of rotating filed motors [1]. As all electric motors consist of a stator and a rotor and work according to principles of the Lorentz force, the stator filed is varied in a rotating field motor and thereby its name. The function of a rotating field motor according to B&R Training Manual 400 [1] can roughly be described as in Figure 2.10. In Figure 2.10 the stator windings are fed with one phase each, the sinusoids are shifted electrically by 120^o. The rotor can either be made up of a permanent magnet or an electromagnet that aligns itself according to the magnetic field induced by the stator. Since the phases are shifted the rotor will be passed between the stator windings [1].

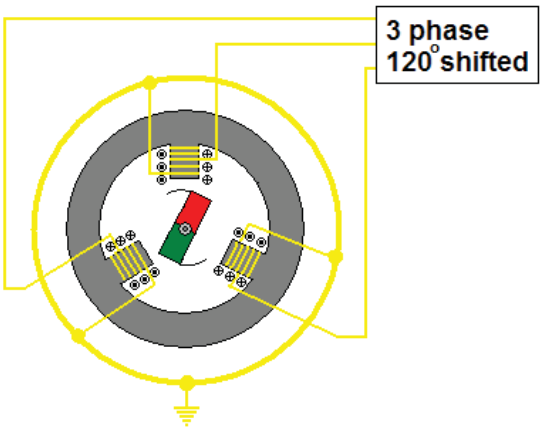

Figure 2.10. *The principle of a 3-phase rotating field motor.*

Induction motors belong to the category of rotating field motors as well as synchronous motors, the difference is how the magnetic field in the rotor occurs [1]. The rotor is always synchronous to the rotating stator field in a synchronous motor and thereby its name.

2.3.1 Motor axis states

In order to include an electric drive in a servo system at least position or speed measurements are needed. Different methods exist; e.g. resolvers and encoders. On the servos in this project high resolution encoders (ENDAT) are available for position measurements. The encoder within the motor housing is actually two encoders in one; both an absolute value encoder and an incremental encoder. This means that the position of the mechanics and the axis movements can be measured better than with solely an absolute value encoder. A simple absolute value encoder can be seen as a plate that rotates with the axis. One axis revolution is divided into a certain amount of binary numbers (e.g. “black-black-white-black-white-black-white-white” corresponds to 00101011). Each position corresponds to a binary number which means that the higher the bit number the higher the resolution. The advantage with the absolute value encoder is that there is a constant offset between the axis position and the robot arm position. A simple incremental encoder (not ENDAT), like the absolute value encoder, consists of a plate (actually the same plate) that rotates with the motor axis. The difference is that the bit number is lower in the incremental encoder than in the absolute value encoder. An incremental encoder increments a counter and thus more states are allowed within one revolution (see Figure 2.11). A state can e.g. be described by “black-white” that corresponds to 01. In summary an absolute value encoder gives the offset between the motor axis and the mechanics and an incremental encoder increments that position to a new value. In this way there is no need for a memory to store the last position when the power is turned off. It should mentioned that there is obviously no “speedometer” inside the servos, instead the position signal is differentiated to get an estimate of the axis speed. The incremental encoder (ENDAT) in the servos allows an even higher resolution than the one described by Figure 2.11. Instead of measuring two phase shifted square waves, the difference between two phase shifted sinusoid signals is also measured. This means that between each increment (or decrement) even more position values can be

obtained. The limit of the resolution lies mainly in the A/D conversion of the signal.

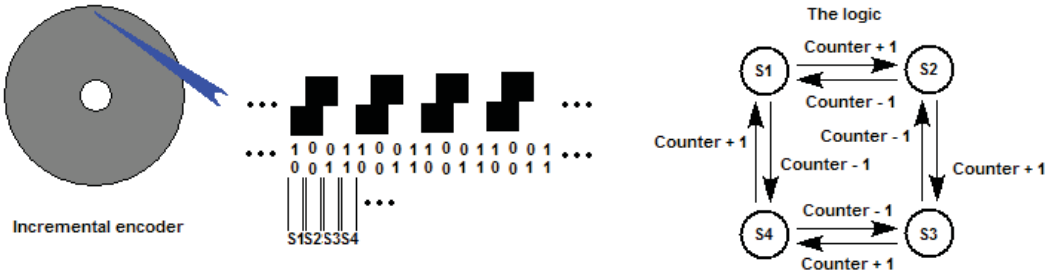

Figure 2.11. *Describes the function of a simple incremental encoder (not ENDAT). In the figure four individual states are possible since two rows are phase shifted* 90^o. *With a state shift a counter is either incremented or decremented, this allows a very high resolution.*

The servo motor is connected to the servo drive ACOPOS via two cables, green and orange in Figures 2.5 and 2.8. The green cable is for the encoder signal and the orange cable is for the motor power signal, i.e., from an automatic control perspective the orange cable is for the control signal and the green cable is for the feedback signal.

2.4 Servo drive system ACOPOS

The servo drive system B&R ACOPOS is a complex design. It consists of circuit boards with standard electrical components and advanced electronics. The servo drive system task is to control the states of the servo motor connected to it. There are many different types with different characteristics and features available in the product category. Table 2.4 shows what types that have been used in this project.

	X-axis	Y-axis	Z-axis
Portal robot	ACOPOS 1016	ACOPOS 1016	ACOPOS 1016
Experimental setup	-	-	ACOPOS 1045

Table 2.4. *The ACOPOS servo drive systems that have been used for each axis on the portal robot and the experimental setup.*

As already stated the ACOPOS is a complex structure, to describe the hardware in too much detail would go beyond the scope of this thesis, therefore a somewhat simplistic approach will be taken in this section.

2.4.1 ACOPOS Power signals

The ACOPOS can e.g. be supplied with single or three phase signals and *24VDC*. Depending on the power supply the system will be able to power the motors less or more. A rectifier circuit is present right after the power inputs, the circuit converts the sinusoids to a positive signal. After the rectifier is an electrolyte capacitor tank which from a physical perspective smooths out the rectified signal.

From a mathematical perspective it multiplies the voltage rms (root mean square) with $\sqrt{2}$. For a $230 VAC$ signal on the input pins of the ACOPOS, the DC voltage between the pins of the electrolyte is given according to Equation 2.9.

$$U_{capacitor} = \sqrt{2} \cdot 230\, VDC = 325\, VDC \qquad (2.9)$$

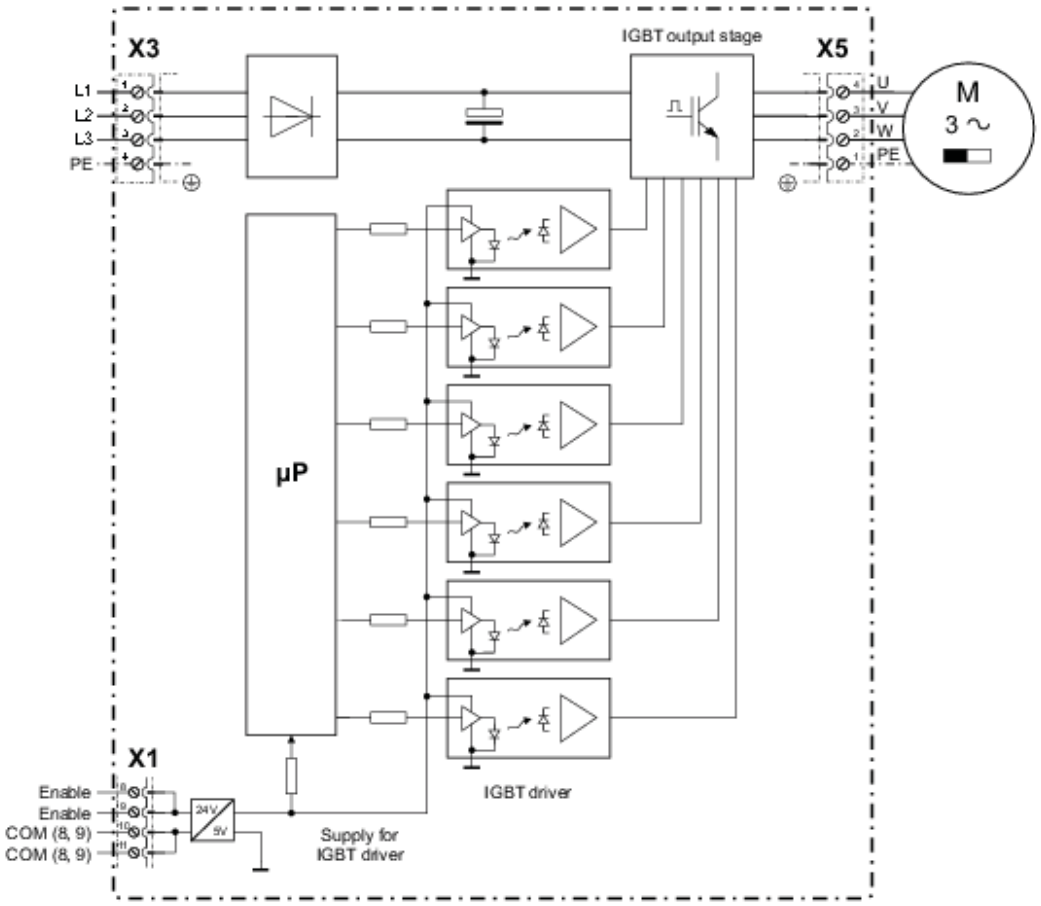

Figure 2.12. *A block diagram of the power signal handling in an ACOPOS. X3 is connected to the power supply and X5 to a servo motor. X1 is connected to a 24VDC supply. This voltage is DC-DC converted to 5V for driving ("enabling") the IGBT driver and logics.*

The voltage in Equation 2.9 lies over the IGBT output stage, this consists of insulated gate bipolar transistors that allow current to the servo motor. The current is varied with a PWM (Pulse Width Modulated) signal on the gates of the IGBTs. The drive current is switched between the stator coils in the motor so that it operates in a three phase manner. The PWM signal (which is the same as the motor control signal) is generated by a microprocessor. Since the motor drive is a high volltage application there is no galvanic connection between the IGBT output stage and the microprocessor. Instead optocouplers are used in the IGBT driver block between the microprocessor and the IGBT output stage. Roughly speaking optocouplers consist of two sides; one side with light emitting diodes and one side with light sensitive diodes. A light sensitive diode allows a current

through it as it is struck by a photon, these photons are generated in the light emitting diode. I.e., the light emitting side is on the same side as the microprocessor and the light sensitive side is on the same side as the IGBT output stage. Figure 2.12 shows a schematic block diagram of the power signal handling in a servo drive system as it is shown in the B&R Automation Studio “help section”.

2.13. S*ome of the connectors of a B&R ACOPOS 1045.*

Figure 2.13 shows the connectors referred to in Figure 2.12 on a B&R ACOPOS 1045. The connector X2 is not shown in Figure 2.12. X2 is connected in parallel with the electrolyte capacitor tank which opens up for the possibility of running the servo drive on a DC voltage (e.g. 24*VDC*) and/or connecting several servo drives in series.

2.4.2 ACOPOS Encoder monitoring

As explained in Subsection 2.3.1 the motor axis angle is observed by a high resolution ENDAT encoder for the purpose of axis state control. The control is achieved by the ACOPOS and thus the encoder and the ACOPOS are connected through a separate cable. Encoder monitoring is realized by a plug-in module to a slot in the ACOPOS, B&R AC120 modules have been used in this project. Figure 2.14 shows an AC120 in an ACOPOS 1045.

Figure 2.14. *A B&R AC120 encoder module in an ACOPOS 1045 servo drive system.*

The functioning mode of the AC120 can be set to ENDAT mode or incremental

mode according to the B&R Automation Studio “help section”. In incremental mode the module is intended to be used with simple incremental encoders with sinusoidal output signals. In ENDAT mode, which is the setting in this project, encoders with absolute value detection and motor parameter memories are used. In this way no new motor settings have to be performed when a motor is replaced by another. Instead the parameters are read from the encoder memory to the AC120 and in turn to the ACOPOS. It is only at startup that an absolute value is read from the encoder in ENDAT mode, at runtime new positions are obtained through increments or decrements of the absolute value. This is because of the reasons explained in Subsection 2.3.1, i.e., higher resolutions are allowed with incremental encoders.

The pin assignments of the 15-pin DSUB socket (see Figure 2.14) differ when the module is set in incremental mode or ENDAT mode. Among the signals are e.g. encoder power supply, data signals, reference pulses, clock signals etc. according to the B&R Automation Studio “help section”.

2.4.3 ACOPOS Powerlink interface

As already stated, the ACOPOS communicates with the PLC via a Powerlink interface. According to the B&R Automation Studio “help section”, Powerlink is an open real-time Ethernet protocol for networking automation components. The communication with the servo drive system is realized with a Powerlink plug-in module, in this project a B&R AC112 Powerlink v1 module. Figure 2.15 shows this module in an ACOPOS 1045.

The connection standard is cat 5e with shield, RXD, inverted RXD, TXD and inverted TXD on the pins. RXD corresponds to the “receive data” pin and TXD to the “transmit data” pin. Each AC112 is given a unique node number so that an individual ACOPOS can be identified in a Powerlink network. The node number is given as a hexadecimal number by two screw switches seen in Figure 2.15. The cycle time for the Powerlink communication is adjustable, however it has been set to $0{,}8\,ms$ throughout this project. The Powerlink protocol is deterministic. This means that a new value is expected every single cycle, i.e., the communication is synchronized with the execution of the program.

Figure 2.15. *A B&R AC112 Powerlink module in an ACOPOS 1045 servo drive system.*

2.5 Programmable logic controller

A PLC of today resembles a computer intended for automation applications. The early PLCs where created to replace relay circuits for binary process control in industry. Relay control systems had several disadvantages; it was expensive to replace whole circuit boards when new functionality was to be implemented, relays are electromechanical components with limited lifetimes, trouble shooting was very difficult as a control system could involve thousands of individual relays etc. [2]. According to G. Olsson and C. Rosen [2] the first PLC was introduced at General Motors in 1968 and the bar was set quite high. The PLC should among other things replace the relay based system, it should withstand the harsh factory environment, be easily programmed and it should be easy to perform programming changes etc. All this should be considered from the perspective that the first microprocessor was invented in 1971 [2]. The early PLCs resembled the relay systems with mainly binary logic, i.e., true or false (1 or 0) logics. During the following decades the development of PLCs has been very rapid in all areas that it concerns. They were made able to send and receive voltages which opened up for integration of PLCs closer to the analog process and they were made able to communicate with other PLCs and computers which opened up for the possibility of creating large networks [2]. What the old PLC and the PLC of today have in common is that they are intended to be programmed (e.g. by a computer) and then run independently of the programmer hardware [2]. However, PLCs have real-time like communication with computers in most complex applications nowadays. E.g., a PLC receives set points and sends error massages from and to computers.

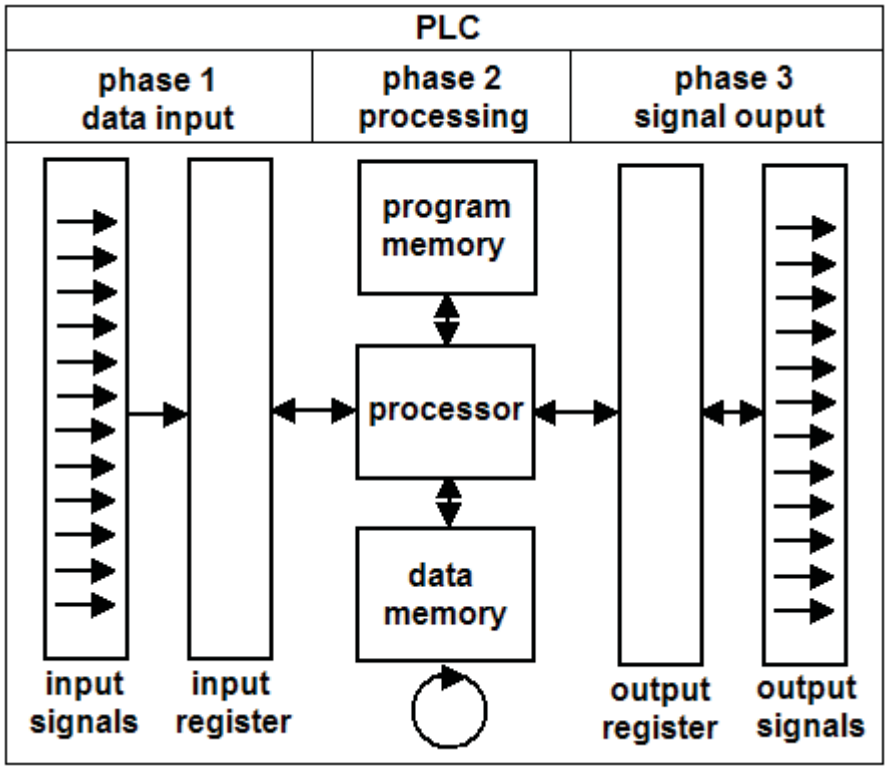

Figure 2.16. *The basic functionality and structure of a PLC. Each cycle (of an "infinite" loop) goes through three phases; data input, processing and signal output.*

Even if the PLC of today has a lot in common with its predecessors it resembles more and more an automation computer with lots of different functionalities. Different applications demand different performances or options. E.g., some applications demand a panel integrated PLC for HMIs and some applications demand a modular design to make integration with for instance I/O modules smooth and handy.

The basic functionality and structure of a PLC can roughly be described by the block digram in Figure 2.16 according to G. Olsson and C. Rosen [2]. As can be seen in Figure 2.16 the general structure of a PLC opens up for control possibilities of almost all industrial processes, the limits are set by the hardware and what is administrated in the program and the data memory.

There are many different types of PLCs in the B&R product range, in this project two kinds have been used; a B&R Power Panel 420 and a B&R X20 CP 1485. The Power Panel 420 has an integrated panel and the X20 CP 1485 has a more modular design, Figure 2.17 shows both.

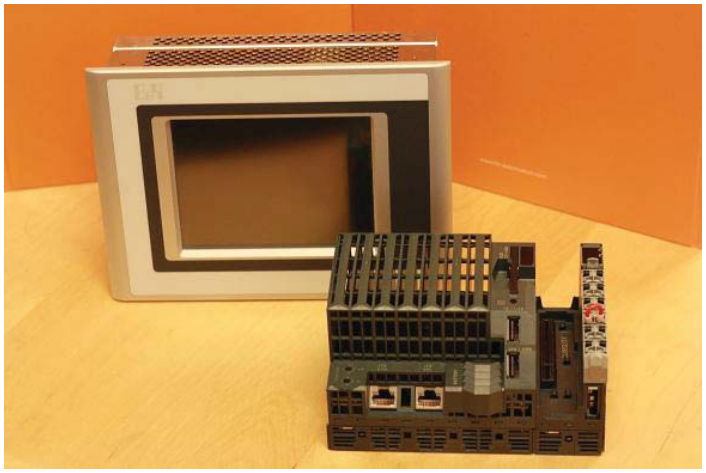

Figure 2.17. *A B&R Power Panel 420 to the left and a B&R X20 CP 1485 to the right.*

The Power Panel 420 is based on a Geode LX800, 500*MHz* clock frequency and 128*kB* L2 cache processor. It has 512*kB* of SRAM and 128*MB* of DDR SDRAM. Interfacing can be done through Compact Flash, Ethernet, USB and RS232 through a 9-pin DSUB socket. There is also an aPCI slot available on this version [3]. The X20 CP 1485 is based on an Intel Celeron 400, 2x16*kB* L1 cache, 256*kB* L2 cache processor. It also features an additional I/O processor. The PLC has 1*MB* of SRAM and 32*MB* of SDRAM. Interfacing can be done via RS232 through a 12-pin terminal block, Ethernet, Powerlink v1/v2, USB or X2X Link [3]. Like the Power Panel 420 it has a slot for Compact Flash cards [3]. The fastest task class cycle time is 0,4*ms* for both described PLCs. Throughout this project it has been set to 0,8*ms*. As already mentioned, interfacing with the PC is achieved through Ethernet and interfacing with the servo drive system is achieved through Powerlink.

2.6 I/O modules

I/O modules have been used in this project for accelerometers. Analog output DC

voltage for accelerometer power is generated by a B&R X20 AO 4622 module. It is capable of delivering two stable DC voltages in the range of $\pm 10\,V$ or two stable currents ranging from 0 to $20mA$ at resolution of 12 bits. The conversion time for all outputs is $0{,}3ms$ [4]. A B&R X20 AI 2632 module handles accelerometer data acquisition, it measures a voltage in the range of $\pm 10\,V$ or a current ranging from 0 to $20mA$ at a resolution of 16 bits. The conversion time for the AI module inputs is $0{,}05ms$ [4]. Since the accelerometers are not intended for control the conversion time is not an issue. Figure 2.18 illustrates how the I/O modules can be connected to a X20 CP 1485 PLC.

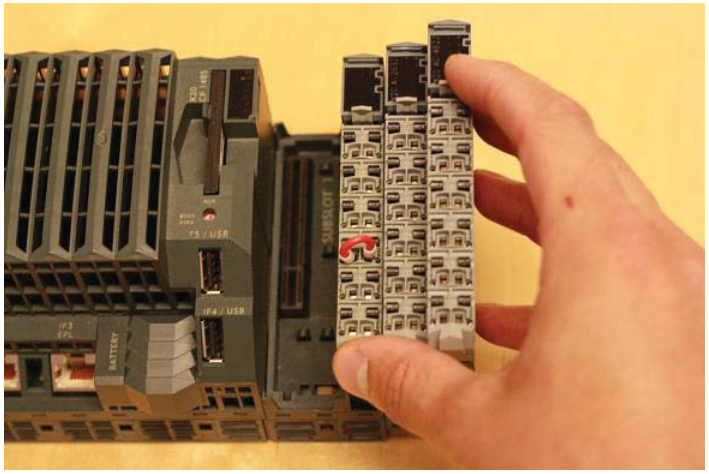

Figure 2.18. *The I/O modules used in this project can be mounted on a X20 PLC in a "plug and play" manner.*

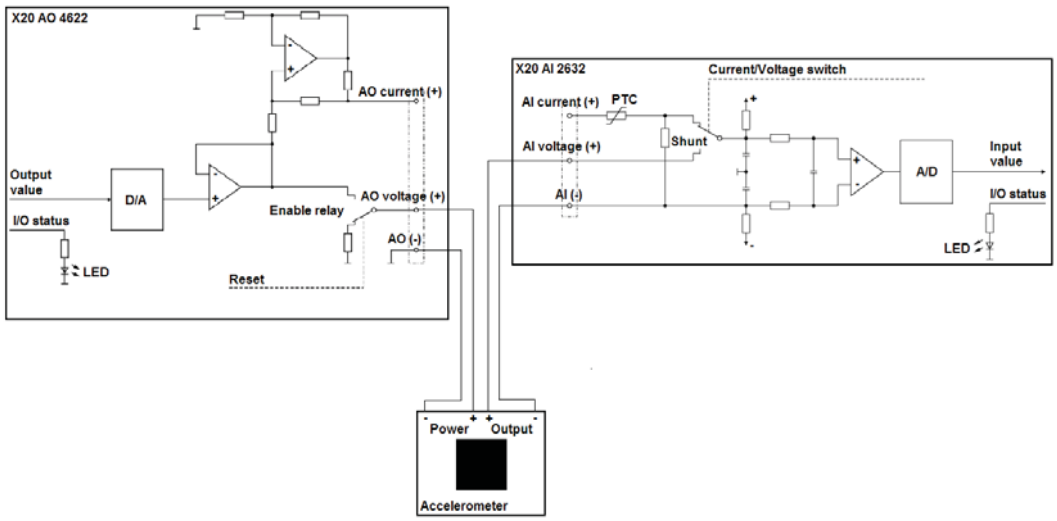

Figure 2.19. O*utput circuit diagram of the AO 4622 and input circuit diagram of the AI 2632. The enable relay is up and the Current/Voltage switch is down when measurements are acquired. An accelerometer is connected according to the figure.*

The AI 2632 features an adjustable input filter for each of the two individual input channels. Up to four first order low pass filters can be cascaded for each channel,

the cut-off frequency can be set from 1*Hz* and higher. The default value is 500*Hz*. In this project, however, a first order low pass filter with cut-off frequency at 50*Hz* has been used. The frequency is chosen so that the common network noise is damped. This is because the environment with lots of AC carrying cables around the accelerometers constitutes a clear source for such noise. Circuit diagrams for one of the inputs and one of the outputs of the I/O modules are shown in Figure 2.19 as they are shown in the B&R Automation Studio "help section". The connection with one accelerometer is also illustrated in Figure 2.19.

2.7 Accelerometers

The accelerometers used are MMA7260Q types mounted on circuit boards with hardware filters. They are capable of measuring accelerations in three dimensions, i.e., *x*, *y* and *z*. According to the data sheet the sensitivity can be set in discrete values of 1,5*g*, 2*g*, 4*g* and 6*g*. In this project the sensitivity was set to 6*g* since the noise to signal ratio is comparably low at this value. The bandwidth in the *x* and *y* directions is 350*Hz* and in the Z direction 150*Hz*. The accelerometers are powered by 3,3*V* and the proportional output is given in the range of 0 to 3,3*V* with 1,65*V* at neutral. With the I/O modules described in Section 2.6 the power and output signals are given as 16-bit integers, i.e., the power signal is given by Equation 2.10

$$U_{power\ digital} = \frac{3{,}3}{10}(2^{15}-1) = 10813 \qquad (2.10)$$

and the output $U_{acc\ ouput}$ is ranging between 0 and $U_{power\ digital}$ with the neutral value given by Equation 2.11

$$U_{neutral\ digital} = \frac{1{,}65}{10}(2^{15}-1) = 5407 \qquad (2.11).$$

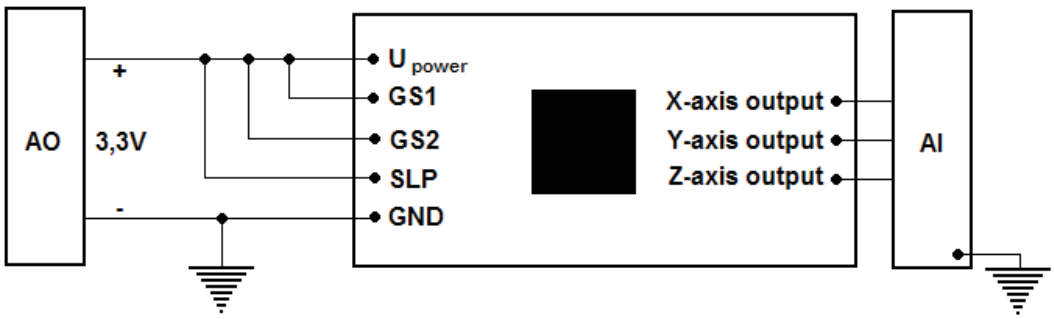

Figure 2.20. *The connection diagram for an accelerometer in the project setup. GS1 and GS2 are put to logical high so that a sensitivity of 6g is obtained, SLP is also put to logical high so that "sleep mode" is deactivated. The same reference ground is used.*

There are eight pins given by Table 2.5 present on the accelerometer IC. The pins GS1 and GS2 control the sensitivity by putting logical high and logical low on them, high-high ($3{,}3V - 3{,}3V$) corresponds to 6*g*. The SLP pin is connected to logical high so that "sleep mode" is deactivated. The connection diagram for an accelerometer in this setup is given by Figure 2.20.

Accelerometer IC pins		
Pin 1	U_{power}	$3{,}3V$
Pin 2	GND	Ground
Pin 3	X-axis output	$0\text{-}3{,}3V$
Pin 4	Y-axis output	$0\text{-}3{,}3V$
Pin 5	Z-axis output	$0\text{-}3{,}3V$
Pin 6	GS1	$0/3{,}3V$
Pin 7	GS2	$0/3{,}3V$
Pin 8	SLP	$0\text{-}3{,}3V$

Table 2.5. *The accelerometer IC pins.*

3. Automation Studio

The programming environment that all programs for control are created in is called Automation Studio. It is also the software tool that is used for hardware management and diagnostics.

Automation Studio is a B&R licensed product and is described as an integrated software development environment [5]. It is a vast tool that includes functionality for many elements of an automation application. Among other areas and components it facilitates [5]:

- Project management
- Programming
- Hardware management
- Integrated visualization
- Motion control
- Diagnostics
- Integrated safety technology
- Simulation
- Help system
- Fieldbus systems
- Communication
- Real-time operating system
- Remote maintanance

In this chapter only project relevant topics of Automation Studio will be discussed. No detailed presentation will be given as that would go beyond the scope of this thesis. Many parts and functionalities that have been encountered in this project will also be left out since they are probably not in the interest of most readers of this thesis. The purpose of this chapter is only to give a brief overview of some used functions so that implementation aspects may be easier to understand.

3.1 Project and hardware management

The user interface is divided into various sections, one section is the project explorer. The project explorer is divided into three tabs; Logical view, Configuration view and Physical view. In Logical view the software organization is presented as a branch structure, all programs in a project can be accessed through folders and it is also possible to create and access documentation for the programs. In the same project different configurations for both software and hardware may be desired, e.g., it is possible that one PLC type is replaced by another. Different configurations are stored and can be activated and deactivated in Configuration view. For a selected configuration all the involved hardware is shown in Physical view. It is for instance possible to access and change module properties, servo drive properties, PLC properties etc.

Among many configuration interfaces for the PLC, the software configuration interface can be accessed through Physical view. There it is possible to activate created programs in parallel cyclic tasks for the PLC and it is also possible to change the task class cycle time (see Chapter 2 Section 2.5) as multiples of the minimum cycle time of the considered PLC.

3.2 Programming

Another section of the user interface is the editor. All programming is conducted in this section. Automation Studio supports many different programming languages; Ladder diagram (LD), Sequential Function Chart (SFC), Function Block Diagram (FBD), Continuous Function Chart (CFC) which are all graphical programming languages [5]. Text based languages Automation Studio supports are Structured Text (ST), Instruction List (IL), ANSI C, C++ and B&R Automation Basic [5]. LD, SFC, FBD, CFC, ST and IL are defined and included in the IEC 61131-3 standard. The IEC 61131-3 standard is an international standard for industrial control programming created by the International Electrotechnical Commission. Its purpose is to harmonize the way people design and operate industrial controls [2].

The languages used for this project has been ANSI C and Structured Text. ANSI C allows the user to call function blocks and access variables from IEC languages [5]. Mixing languages within a project can therefore be done quite conveniently. It is said that the hardware equivalent of a function block is an integrated circuit (IC); like an IC a function block generates certain outputs from certain inputs. Inputs and outputs can for a function block be e.g. objects or variables.

Automation Studio integrates many different tools and function blocks, among these are PLCopen function blocks for positioning tasks [5]. PLCopen function blocks conform to the IEC 61131-3 standard and are e.g. complemented by B&R function blocks. The ANSI C programs that are created within Automation Studio are compiled by the GNU C compiler. Only programs that are activated in the PLC (see Section 3.1) are compiled when the “compile command” is given.

3.3 NC Test, Trace and monitor mode

Network Command Test (NC Test) is a powerful tool when it comes to axis motion tests. From the NC Test center it is possible to give commands to an ACOPOS to commence movements on an axis as specified in the NC Test Structure interface (see Figure 3.1). It is also possible to trace motor and ACOPOS states through the NC oscilloscope. As on a real multi channel oscilloscope it is possible trigger on different signals and there are also possibilities for signal treatment (e.g. FFT) and other operations on chart data. Through the NC Test structure interface it is possible to adjust axis limits, servo drive control parameters etc.

When different tasks (implemented programs) are running on a PLC (or a PLC simulator) it is possible to put Automation Studio in “monitor mode”. Variables can then be changed on the PLC and obtained from the PLC. This is done through the variable monitor. The trace function can be used to record changes on variable values on the target system synchronously with respect to the

task class [6]. It is possible, like in the NC Test interface, to trigger a trace on different variables and then conduct various kinds of operations on the data. Figure 3.1 shows a screen shot of Automation Studio when a NC Test window is opened with the Structure interface shown. Figure 3.2 shows a screen shot of Automation Studio when a NC Test window is opened with the Trace interface shown.

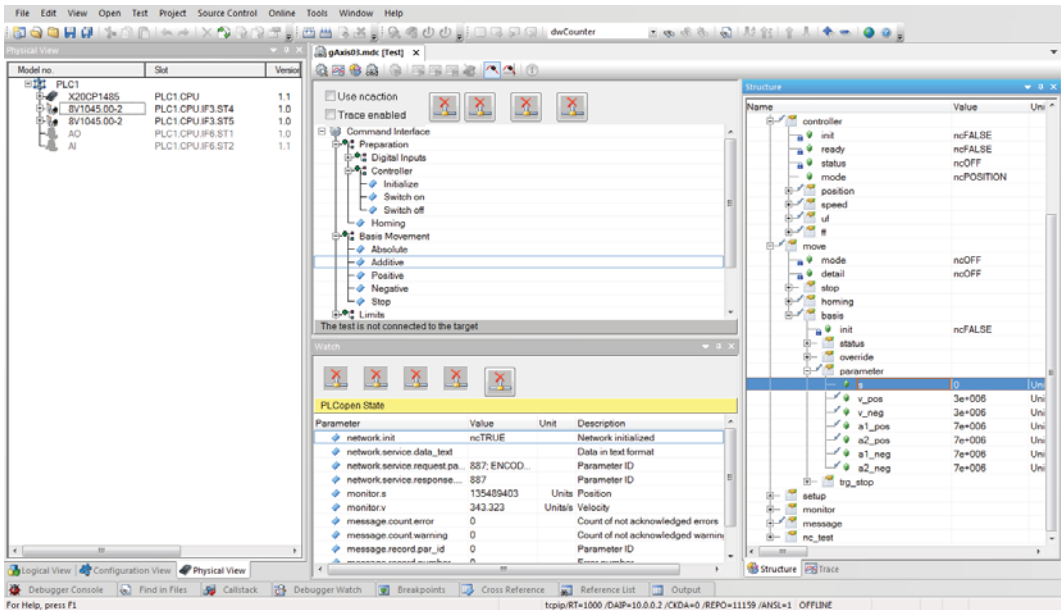

Figure 3.1. *A screen shot of Automation Studio when a NC Test window is opened with the Structure interface shown. The physical view in the project explorer is seen on the left side of the screen.*

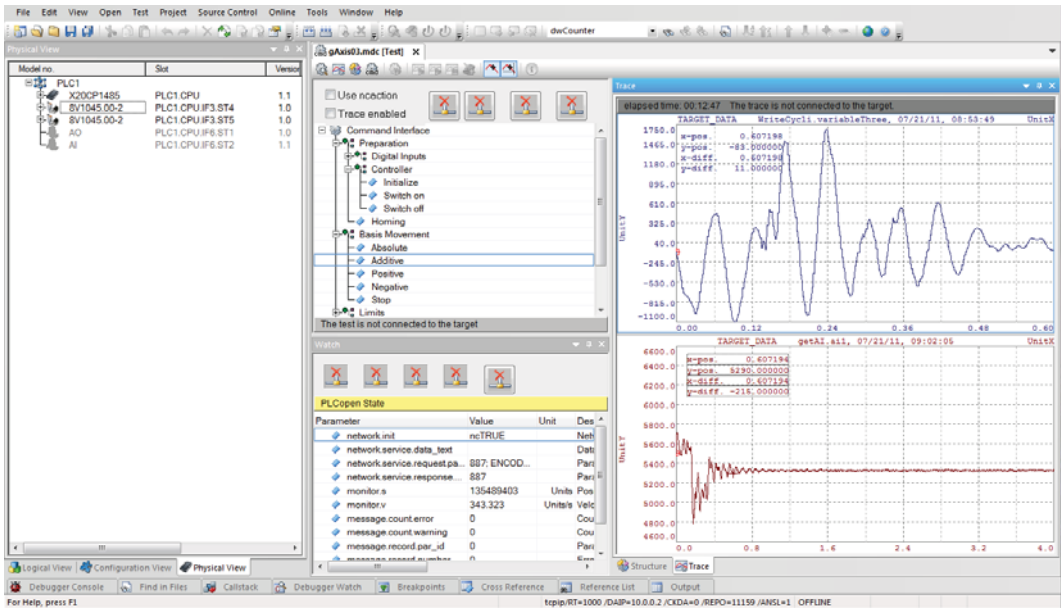

Figure 3.2. *A screen shot of Automation Studio when a NC Test window is opened with the Trace interface shown. The logical view in the project explorer is seen on the left side of the screen.*

4. Servo drive control concept

In this chapter elementary servo drive control will be presented. Then a more detailed discussion will follow for the servo drive system at hand, namely the ACOPOS. The motor states; axis position, axis speed, axis acceleration etc. are of interest to control in a wide set of different applications.

It is important that the control is accurate. If many machines are synchronized in a complex structure this becomes clearly evident. Imagine a plant where one machine, e.g. a conveyor belt, fails to maintain a certain speed. The task for this conveyor belt is to deliver an unfinished product to another machine for processing. This other machine "knows" that it can expect an unfinished product at a very specific rate, therefore the machine triggers its process at a certain point in time. If the machine does not get the unfinished product and the process is triggered, then in the next cycle it will have two products to process. This may not be possible and a "pile up" becomes a fact and that could lead to a chain reaction that propagates further back in the plant. Naturally, various kinds of sensors can prevent such a scenario. In the case of the conveyor belt an optical sensor that "sees" if the unfinished product enters the machine or not could be present. The machine would then wait for a boolean value to become true before it starts processing the unfinished product. The problem is that sensors are often expensive and can also break. It is somewhat desirable to avoid many sensors if possible. Therefore accurate servo speed control is important in the conveyor belt example above. The example can easily be generalized to numerous applications.

It is also very important for the position control of the servo motor axis to be accurate. We can easily imagine what happens if the moving unit of an overhead crane goes further than its physical position limit, e.g. a wall or another machine. Applications involving portal robots serve as very good examples. One application with a portal robot could, as already indicated, be a pick and place task. The robot Y-axis should enter a machine and grab an unfinished product with an expensive grip tool mounted on the tip of the axis. It should then transfer the unfinished product to another machine for further processing. Usually the geometric margins are small in all directions for such applications, therefore accurate positioning is crucial. If the positions of the involved servo axes are slightly inaccurate, the worst case scenario could be damaging of the expensive tool.

4.1 Basic motor control

The discussion in this section is mostly based on information from Elmaskinsystem, G. Olsson et al. [7].

The need of measuring more states than just the current in a servo can be illustrated by the following example. Assume that it is desired to control the rotational speed of the axis on a DC motor. This can be done by applying a certain voltage over the armature (rotor). The rotational speed will remain the same as long as no load is applied. In the case of a load disturbance on the axis, the rotational speed will decrease if the same voltage is kept over the armature. The voltage needs to be raised in order to keep the desired rotational speed, i.e., speed measurements and feedback of such are needed. For a DC motor in stationarity, i.e., constant magnetic field, the produced motor torque T_{motor} can be described by Equation 4.1

$$T_{motor} = \kappa i_a \qquad (4.1)$$

where κ is a motor constant and i_a is the current through the armature. The dynamics of the rotor can be described by Equation 4.2.

$$L_a \frac{di_a}{dt} = u_a - \kappa \omega - R_a i_a \qquad (4.2)$$

where L_a is the armature inductance, u_a the voltage over the armature, ω the rotational speed and R_a the armature resistance. As stationary conditions apply, Equation 4.2 can be reduced to Equation 4.3.

$$u_a - \kappa \omega - R_a i_a = 0 \qquad (4.3).$$

Under the same conditions, the torque equation for the loaded DC motor becomes

$$T_{motor} = \kappa i_a = T_{load},$$

i.e., Equation 4.4 is imposed with help of Equation 4.3.

$$T_{motor} = -\frac{\kappa^2}{R_a}\omega + \frac{\kappa}{R_a} u_a = T_{load} \qquad (4.4).$$

Equation 4.4 can be described as in Figure 4.1 where the function between torque and rotational speed has been drawn.

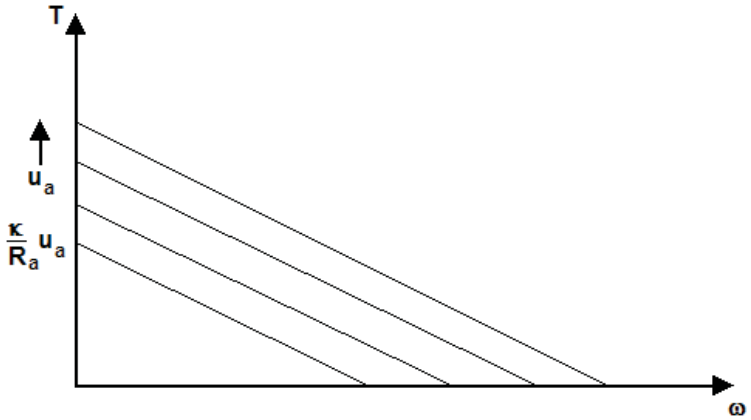

Figure 4.1. *The relation between the rotational speed and the armature torque when steady state conditions apply.*

Figure 4.1 is based upon the assumption that stationary conditions apply. However, it motivates the need of negative feedback of the angular speed. If a torque load is introduced (higher torque) the angular speed will decrease accordingly (on one of the lines). To compensate for this torque and to reach back to the desired angular speed it is necessary to "step over to the next line", this is done by increasing the armature voltage. The angular speed can either be estimated through Equation 4.3 as

$$\omega = \frac{u_a - R_a i_a}{\kappa}$$

with measurements of the armature voltage and current, or measured with various kinds of encoders. The latter alternative is preferable since Equation 4.3 applies for steady state conditions only.

For a synchronous motor, as is used in this project, the current-torque-speed relationships are a lot more cumbersome to derive. The synchronous motor structure is described in Chapter 2, Section 2.3. The problem has to be formulated with vectors expressed in stator and armature coordinates, more about how this is done can be found in G. Olsson et al. [7]. What the DC motors and the synchronous motors have in common is that, when used in servo applications, current control is needed.

A synchronous motor gives a defined *maximum* torque at the present angular speed. The angular speed of the rotor ω depends on the frequency of the power signals to the stator and the number of poles (phases) according to

$$\omega = \frac{2}{p}\omega_{power}$$

where p is the number of poles and ω_{power} is the angular frequency of the power signal. Given a 50*Hz* signal the motor illustrated in Chapter 2 (Figure 2.10) would run at 2000*rpm*. The angular speed of a synchronous motor can thus be controlled by the frequency of the power signal. The torque however, still needs to be

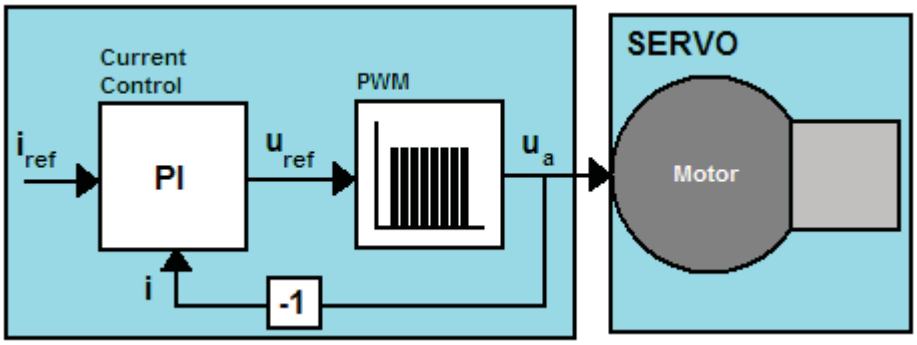

Figure 4.2. *Shows the standard solution for current control in a DC motor. The reference current* i_{ref} *is compared to the real current i and adjusted with a PI controller. Usually the reference voltage* u_{ref} *is transformed to a PWM signal.*

controlled. The standard solution for this is PI controllers for both the DC motor and the synchronous motor. For the DC motor this is done quite straight forwardly, the only feed back signal needed is the armature current according to Figure 4.2. Usually, the voltage to the armature u_a have the form of a pulse width modulated (PWM) signal. A PWM signal is actually just a quickly switched square wave. In all equations above where u_a has occurred, it has been assumed that u_a is a continuous voltage. This is however not a bad approximation as long as the switching frequency is fast enough. Figure 4.3 illustrates the principles of a PWM signal.

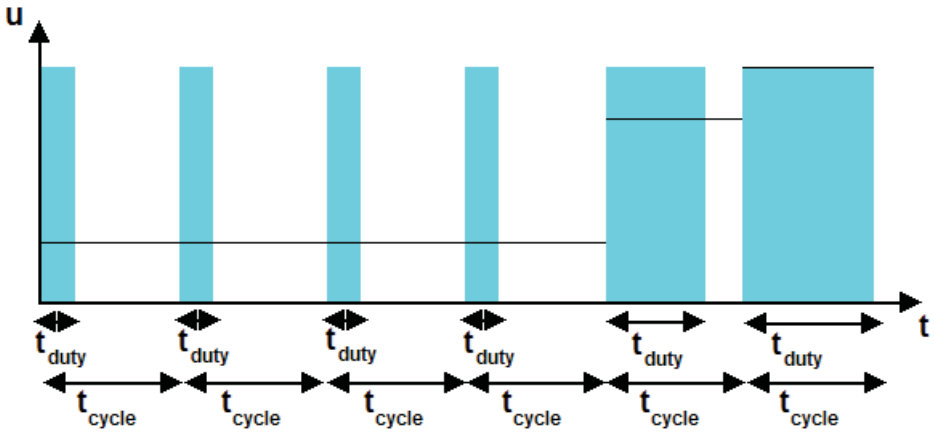

Figure 4.3. *Shows a simplistic illustration of the principles of a pulse width modulated signal.*

By varying the duty time t_{duty} (the "on time") of the constant voltage it is possible to achieve a "mean value" voltage that is lower. If $t_{duty}=t_{cycle}$, i.e., if the duty time equals the switching cycle time, the maximum voltage is obtained. This means that the voltage is usually adjusted as a fraction between the duty time and the switching cycle time in a microprocessor. Naturally, the "mean voltage" will not be constant between the cycles but if the switching frequency is fast and the "load" has a certain inertia this is not a problem.

We also see that it is possible to create a "mean signal" with a sinusoid appearance (as is needed for synchronous motors) by means of a PWM signal. The current control is as already stated different for synchronous motors. According to G. Olsson et al. [7], the reference currents in a stator and a rotor are calculated in a rotating coordinate system. The absolute value of the current vector and its angle relative to the rotor are needed for obtaining a certain torque. These currents are converted to two phase components expressed in stator coordinates.

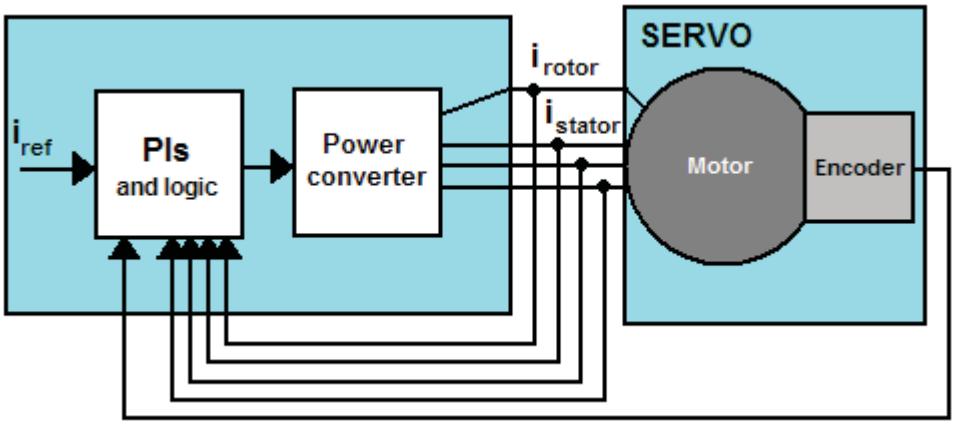

Figure 4.4. *A schematic illustration of the current control in a synchronous motor.*

The two phase components are then transformed to three phase currents for the stator poles. PI regulators are most often used for controlling these currents by adjusting a PWM voltage. To measure the rotor angle for current control an encoder can be used. Figure 4.4 shows a schematic block diagram for current control of a synchronous motor. With the differences and the similarities between the somewhat trivial current control of a DC motor and the bit more advanced current control of a synchronous motor in mind, it is fair to make the following approximation.

The differential equation between the current reference i_{ref} and the angular speed of the motor axis ω can be expressed as Equation 4.5.

$$\tau_c \dot{\omega} + \omega = K_c i_{ref} \qquad (4.5)$$

where τ_c is the collected system time constant and K_c is a gain factor. Equation 4.5 is a good approximation as long as the PI controller(s) are well tuned. Motor dynamics such as e.g. resonances are not considered in the model. In the Laplace domain Equation 4.5 can be rewritten as Equation 4.6.

$$\Omega(s) = \frac{K_c}{\tau_c s + 1} I_{ref}(s) = G_c(s) I_{ref}(s) \qquad (4.6)$$

in other words, the feedback of current and PI control makes the assembled systems in Figures 4.2 and 4.4 have the properties of low pass filters between reference current and angular speed. This approximation will be made throughout the rest of this thesis and the current control of the motors will not be studied in further detail.

4.1.1 Axis angular speed control and axis angle control

As already stated, the angular speed needs to be measured or estimated and controlled by means of some kind of feed back. So does the angle ϕ in many

applications and especially in the application that constitutes the subject of this project. In most industrial servo drives this is done with cascaded PI controllers. A simplistic block diagram (for a DC motor) is shown in Figure 4.5.

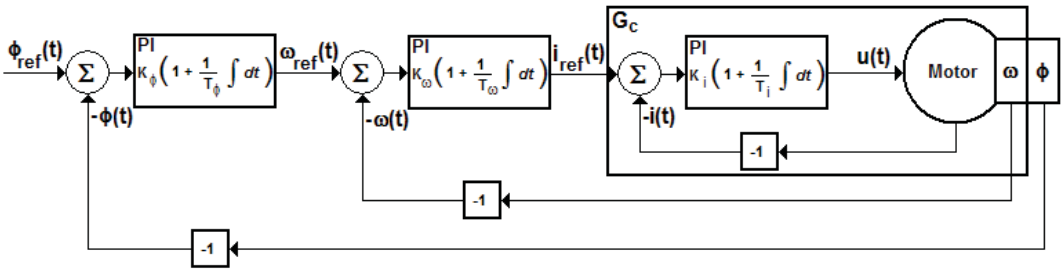

Figure 4.5. *A block diagram with a simple cascaded PI control structure for current, axis angular speed and axis angle of a DC motor. The adjustable controller parameters are* $K_\phi, K_\omega, K_i, T_\phi, T_\omega$ *and* T_i. *Note that it is only for the synchronous motor that encoder feedback signals are needed for current control.*

Figure 4.5 shows the basic control strategy for a DC motor, the main difference with respect to the synchronous motor is the current control. Angular speed control and angle control can be considered analogous between the two motor types. We see that if the integral parts are neglected in all the three controllers, a state feed back scenario arises and the control signal *u* is given according to

$$\begin{aligned} u &= ((K_\phi(\phi_{ref} - \phi) - \omega)K_\omega - i)K_i = \\ &= K_\phi K_\omega K_i(\phi_{ref} - \phi) - K_\omega K_i \omega - K_i i = \\ &= -\bar{L}\boldsymbol{x} \end{aligned}$$

where

$$\bar{L} = \begin{pmatrix} -K_\phi K_\omega K_i & K_\omega K_i & K_i \end{pmatrix} \quad \text{and} \quad \boldsymbol{x} = \begin{pmatrix} \phi_{ref} - \phi \\ \omega \\ i \end{pmatrix}.$$

Cascaded PI-controllers are well established and have proven to work very well in servo applications. The tuning of the regulators starts in the inner fastest loop (the slave) and proceeds towards the outer slowest loop (the master).

The main reason for using a cascaded structure is that in this way disturbances acting on an inner signal can be handled faster and before they appear in an outer signal. Another advantage with a cascaded structure is that the dynamics of the process that the master controller should regulate is simpler. Instead of working towards the whole process, the process is divided into simpler subprocesses each regulated by their own controller [8].

The controllers implemented in practice are naturally not as simple as might be suggested by Figure 4.5. There are most often signal filtering, feed

forward, integral anti wind-up schemes etc. Of course there are also nonlinearities present such as control signal limitations (both physical and in software).

4.2 ACOPOS control

So far basic motor control has been presented. A general description of the ACOPOS hardware is found in Chapter 2, Section 2.4. In this section the aim is to give an overview of some of the functionalities of the ACOPOS firmware. The discussion is mostly based on information from ACOPOS Control Concept and Adjustment [9]. The ACOPOS is as already mentioned where the motor control occurs. In normal operation the PLC only provides set points to the ACOPOS. The control strategy in the servo drive system implements cascaded PI controllers with the general structure as in Section 4.1. The current controller will not be studied in further detail due to reasons already explained. Instead a simplistic approach close to the DC motor will be taken; PI controllers regulate a voltage to the stator based on current measurements, even though it has been explained that some kind of rotor angle measurement is needed too. A schematic block diagram of the control structure, roughly as it is described in The Basics Of Motion Control [1], is found in Figure 4.6.

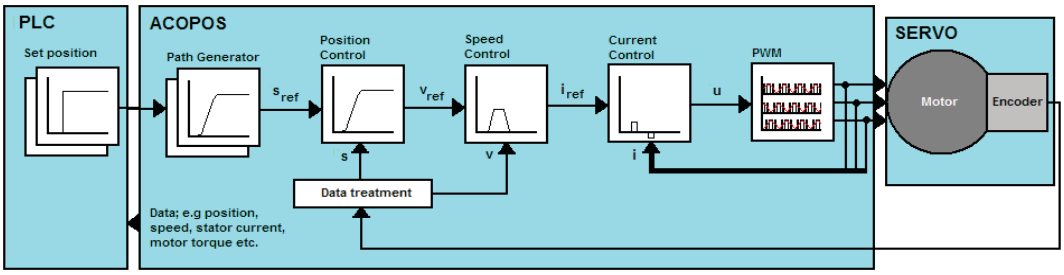

Figure 4.6. *Block diagram of the control structure in the ACOPOS firmware. Set points for acceleration, speed and position are given by the PLC. These are then combined and shaped to a position profile in the "Path Generator". The position and speed controllers are both PIs. To obtain an estimate of the angular speed the position signal from the encoder is differentiated.*

From Figure 4.6 we see that a movement on the motor axis is initiated from set points in the PLC; acceleration, speed and position. These set points are then used to calculate a position profile that acts as the reference values for the position controller. The path is calculated in what is called the "Path Generator".
The position reference is then compared with the actual position and an angular speed reference for the speed controller is calculated by the position PI. The actual speed is estimated by differentiation of the actual position. This value is then compared with the reference value and a current reference is calculated for the current controller by the speed PI.

In Figure 4.6 the angle position is denoted as *s* and the angular speed is denoted as *v*. The units that the quantities are given in in the project setting are based on one revolution of the motor axis corresponding to the *integer* 36000*units* (see Chapter 2, Subsections 2.1.1 and 2.2.1). This means that

$$s = \textbf{round}\left(\frac{36000}{2\pi}\phi\right) \quad \text{and} \quad v = \textbf{round}\left(\frac{36000}{2\pi}\omega\right).$$

4.2.1 The Path Generator

A position reference shaped as a step is neither physically realistic nor gentle for the motor and the mechanics to follow. During project configuration the motor and mechanical limitations are taken into consideration. Acceleration and speed limits are set accordingly and if the mechanical process has limits on the position, those are also set accordingly. There are seven parameters that can be set to configure a movement to a certain position by the PLC (see Table 4.1).

Parameter	Description
s	Target position
v_pos	Speed in positive direction
v_neg	Speed in negative direction
a1_pos	Acceleration in positive direction
a2_pos	Deceleration in positive direction
a1_neg	Acceleration in negative direction
a2_neg	Deceleration in negative direction

Table 4.1. *Parameters that can be set by the PLC to shape a position profile in the ACOPOS Path Generator.*

An example of how a profile is generated with

$$\begin{aligned} s &= 10000\,units \\ v\_pos &= 10000\,units/s \\ v\_neg &= 10000\,units/s \\ a1\_pos &= 100000\,units/s^2 \\ a2\_pos &= 100000\,units/s^2 \\ a1\_neg &= 100000\,units/s^2 \\ a2\_neg &= 100000\,units/s^2 \end{aligned}$$

is illustrated in Figure 4.7. For the profile calculation in Figure 4.7 the only needed acceleration values are a1_pos, a2_pos and the only needed speed value is v_pos since the movement is conducted in the positive direction.

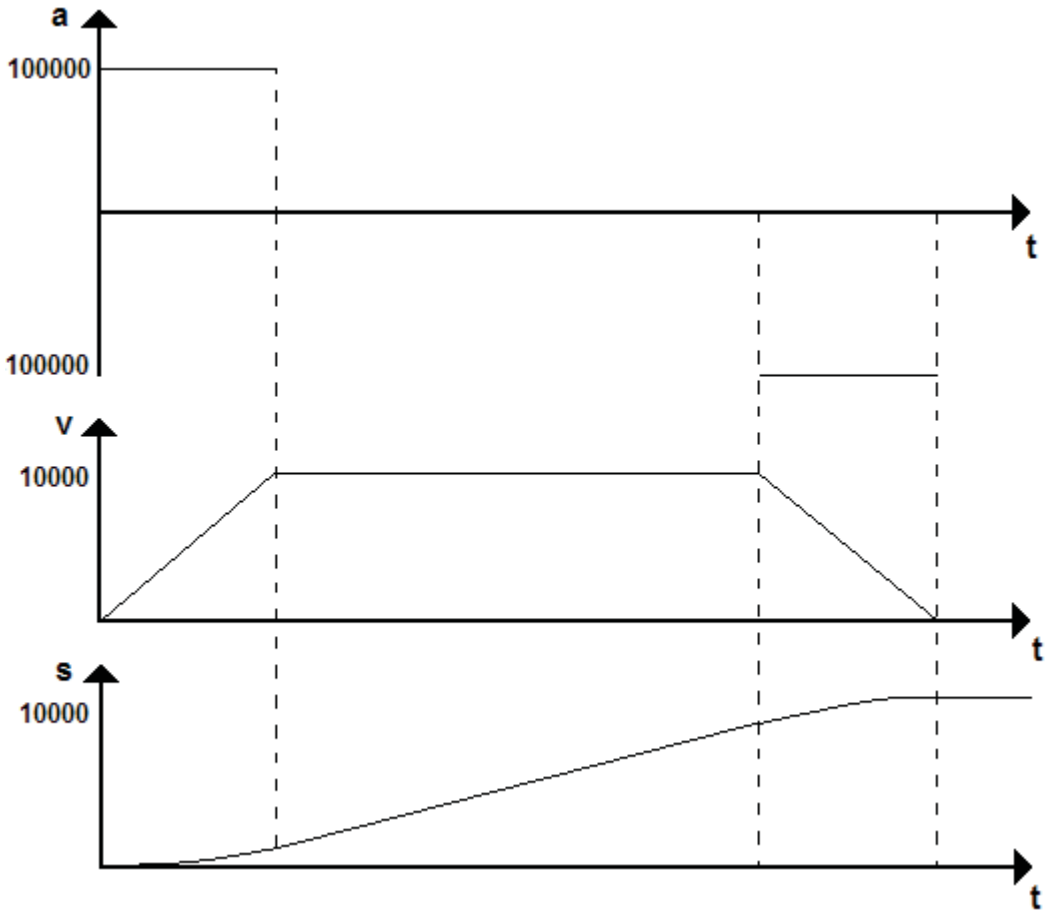

Figure 4.7. *An example of a position reference profile provided by the Path Generator.*

As we see in Figure 4.7 the acceleration derivatives equals positive and negative infinity. This demands a high change of torque which is not gentle to the motor and the mechanics. The high torque slopes will most often also cause vibrations in the mechanical structure [9]. It is these vibrations that constitute the subject for this thesis and the topic of Chapter 5. A standard remedy for vibrations is so called jolt limitation. When configuring the parameters in Table 4.1 it is also possible to set the parameter t_jolt which is called “jolt filter time”. Figure 4.8 shows a profile with an active jolt filter.

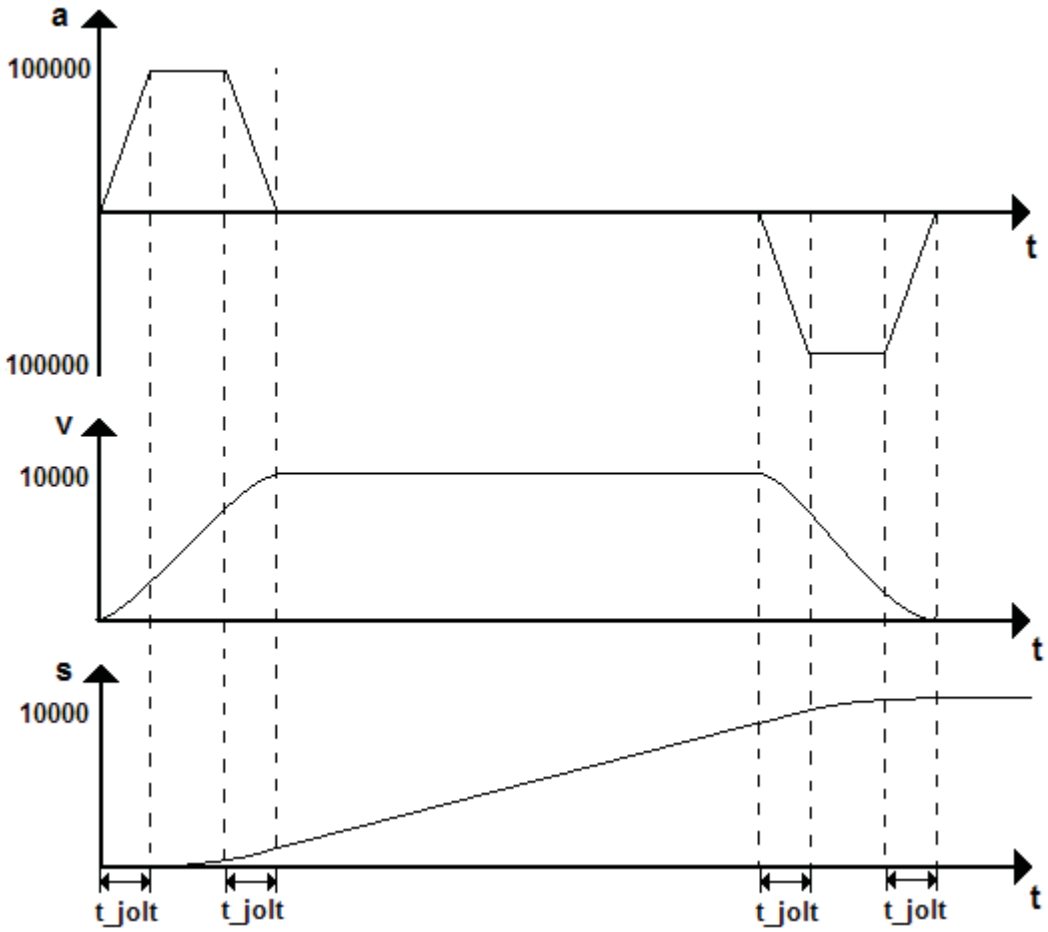

Figure 4.8. *An example of a position reference profile generated with active jolt filter.*

As we see in Figure 4.8, the parameter t_jolt defines the time it takes for the acceleration to reach its final value. Moreover, the time for the position reference to reach its final value is increased by t_jolt compared to a profile without jolt limitation.

The Path Generator runs at a cycle time of 0,4*ms,* i.e., a new reference value for the position controller is obtained every new cycle.

4.2.2 Position control

The position controller (or the master) is as already mentioned implemented as a PI controller. Instead of a derivative part as in a PID controller, it implements feed forward to the speed controller (or the slave). It is of course possible to deactivate both the feed forward part and the integral part. Figure 4.9 shows a schematic block diagram of the position controller roughly as it is shown in ACOPOS Control Concept and Adjustment [9].

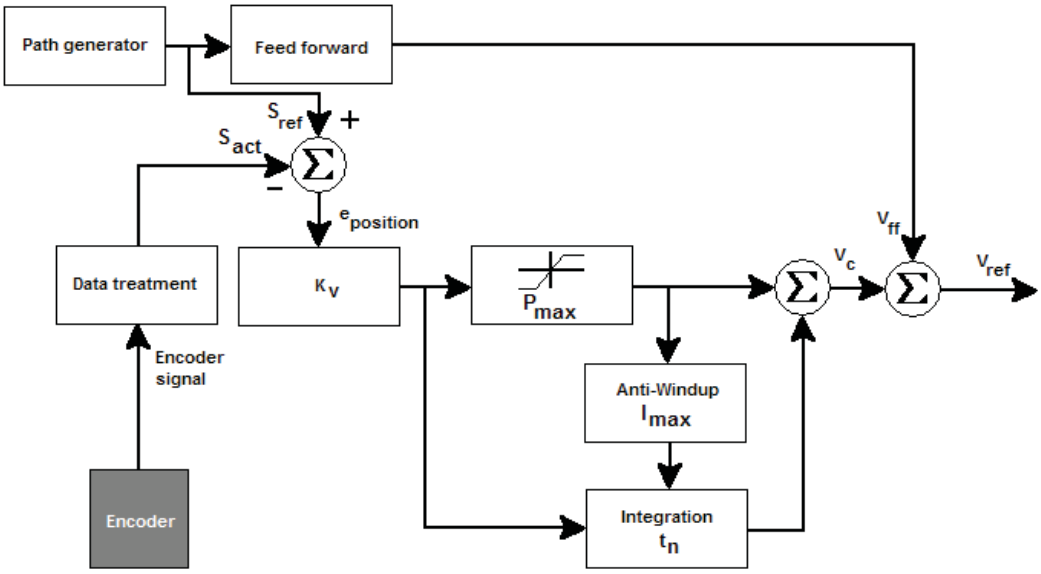

Figure 4.9. *A block diagram of the position controller. The reference values* s_{ref} *from the Path Generator goes through a feed forward model to obtain a feed forward reference speed* v_{ff}. *The actual position* s_{act} *is subtracted from* s_{ref} *and the error is scaled by gain factor* K_v, *this signal is then integrated and the obtained signal is added to the scaled error. The signals* v_c *and* v_{ff} *are added to obtain the reference speed* v_{ref} *for the speed controller. Note that there is an anti-wind up scheme and a P-part limitation implemented in the controller.*

The cycle time of the position controller is the same as the cycle time of the Path Generator, i.e., 0,4*ms*.

4.2.3 Speed control

The speed controller (position controller slave) is also implemented as a PI controller. It runs at a cycle time of 0,2*ms* and therefore interpolation of the reference value v_{ref} from the position controller is conducted. Limitation of the reference signal as well as limitation of the controller output is possible. It is also possible to add a signal v_{addref} to the reference v_{ref}. According to ACOPOS Control Concept and Adjustment [9] the limitation of the controller output is implemented for two reasons; the ACOPOS should not output more current to the motor than the motor can handle (the motor peak current) and the motor stator current should not exceed the ACOPOS peak current. The lower rated peak current of the two is chosen as the limit value.

After being differentiated, the encoder signal is low pass filtered to obtain the “actual speed” v_{act}. The low pass filter is needed since differentiation of noise enhances it significantly and signal noise is most often of high frequency nature.

A notch filter is implementable on the controller output. However it is important not to use it at low frequencies. Disturbances on the process is often of low frequency nature. If the notch is set at a particular disturbance frequency, that disturbance is not handled by the controller. In other words, it would be possible to move the process manually at the notch frequency, thus the notch filter should

only be used for high frequencies. Figure 4.10 shows a schematic block diagram of the speed controller roughly as it is shown in ACOPOS Control Concept and Adjustment [9].

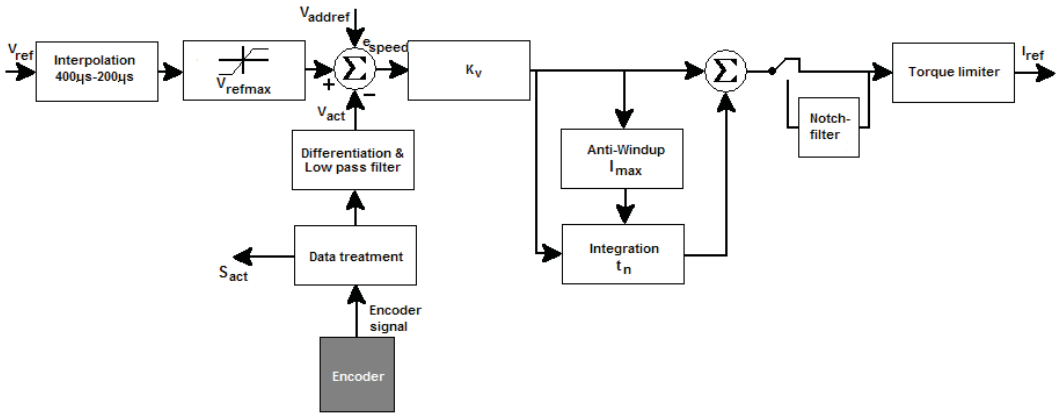

Figure 4.10. *A block diagram of the speed controller. The reference value* v_{ref} *from the position controller is interpolated and limited to obtain the speed reference. The actual speed* v_{act} *is subtracted from the speed reference and the error is then scaled by the gain factor* K_v. *Note that it is possible to add a signal* v_{addref} *to the speed reference. The scaled error is integrated and the obtained signal is then added to the scaled error. The current reference* i_{ref} *is limited based on ACOPOS or motor peak current ratings before it is sent to the current controller.*

It should be mentioned that the current controller runs at a cycle time of 0,05-0,2*ms* depending on the PWM frequency to the output stage. It is also possible to add a signal i_{addref} to the reference current i_{ref}.

5. Vibration phenomenon

In Chapter 4 the importance of accurate servo control is stressed. If we only would consider theoretically rigid structures in the sense of classical mechanics, accurate control of the motor states would be enough. The important states of the rest of the process would be proportional to those of the motor.

The subject of this thesis arises because distortion terms have to be added to the motor angle position, speed and acceleration in order to describe the same states of the whole process. As an interesting observation and perhaps as a bit of a detour from the subject, it can be said that there cannot exist any rigid structures even in theory. Imagine a beam that hits a wall with one of its ends. Since no material information can travel faster than the speed of light the information that a wall has struck one end must take some time to reach the other end. Hence, material distortion must take place.

In Chapter 2 the position distortions of the tool point are described by

$$\boldsymbol{P}_{\boldsymbol{dist}} = (x_{dist}, y_{dist}, z_{dist}).$$

In this chapter focus lies on the appearance of these distortions. First a more general discussion will be given and then a more specific discussion for the portal robot and for the experimental setup will be given.

5.1 Vibrations in flexible beams and in belts

It is possible to make mathematical modeling of flexible beams very detailed but as there is no need for this here the discussion is kept quite simple. The main concern in the problem at hand is the tool point deviation from the reference value. How the beam structure behaves is not of interest if the behavior is not seen at the tool point. This can also be motivated with the fact that in applications similar to portal robots the highest amplitudes are seen at the boundaries of the beams opposite to the motor side boundaries. Canceling vibrations of one end point is often analogous to canceling vibrations of the whole beam. Figure 5.1 illustrates how the problem for one beam can be approximated when exposed to a force in the z direction.

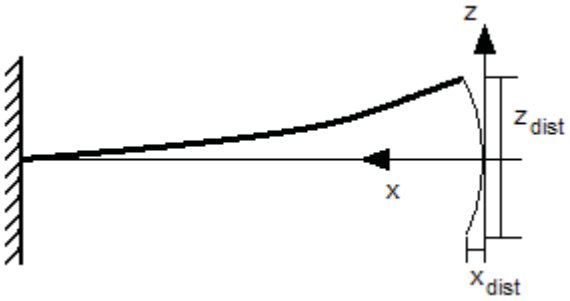

Figure 5.1. *Bending of a fixed flexible beam considered in a plane. The end point deviations* x_{dist} *and* z_{dist} *are marked.*

An approximate graph of the distortions $x_{dist}(t)$ and $z_{dist}(t)$ is shown in Figure 5.2.

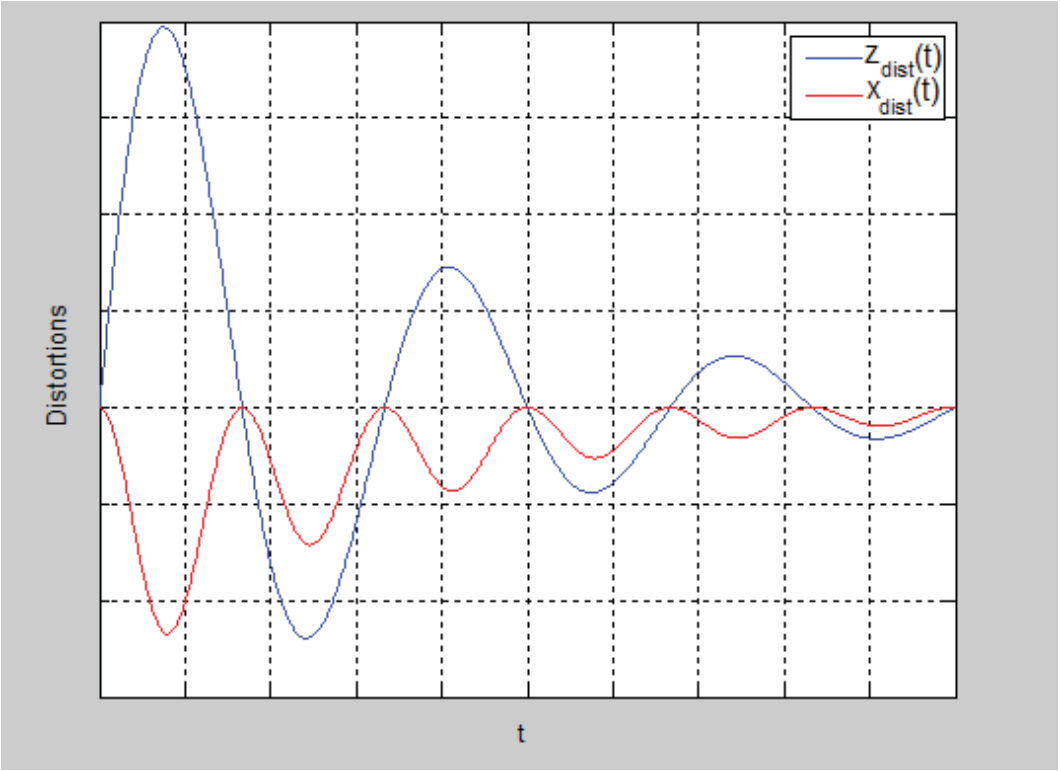

Figure 5.2. *An example of the approximate appearance of distortions in a beam that is exposed to an impulse force in the z direction. The x direction function oscillates with the double frequency of the z direction function.*

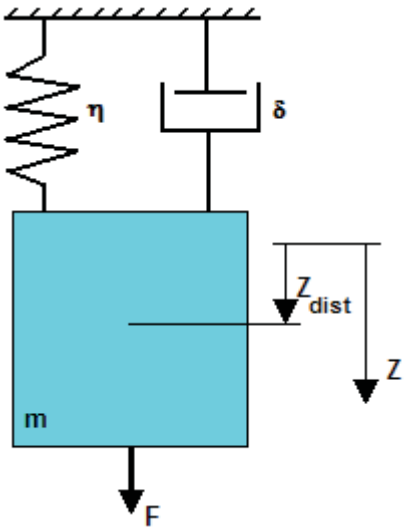

Figure 5.3. *A mass-spring-damper process exposed to an external force F.* η *is the spring constant,* δ *the damper constant and m the mass.*

Even though an applied impulse force in one direction on a beam gives rise to

distortions in two dimensions, it is only needed to consider the largest component since the smallest is very small and does not appear if the largest does not appear. In Figures 5.1 and 5.2 the largest component corresponds to z_{dist}.

Given this, the distortion z_{dist} may be approximated with the output of a mass-spring-damper system according to Figure 5.3. The force balance equation for the process in Figure 5.3 can be written as Equation 5.1,

$$m\ddot{z}_{dist}+\delta\dot{z}_{dist}+\eta z_{dist}=F \qquad (5.1)$$

or instead with the motor torque T_{motor} as Equation 5.2.

$$m\ddot{z}_{dist}+\delta\dot{z}_{dist}+\eta z_{dist}=k_{motor}T_{motor} \qquad (5.2)$$

where $k_{motor}\ [1/m]$ is the conversion constant between the applied force and the corresponding torque. Equation 5.2 can be rewritten as

$$\begin{aligned} &m s^2 Z_{dist}(s)+\delta s Z_{dist}(s)+\eta Z_{dist}(s)=k_{motor}T_{motor}(s) \Leftrightarrow \\ &Z_{dist}(s)=\frac{k_{motor}}{m s^2+\delta s+\eta}T_{motor}(s)=G_{dist}(s)T_{motor}(s) \end{aligned}$$

in the Laplace domain. The second order transfer function $G_{dist}(s)$ can be rewritten on the more conventional form of Equation 5.3

$$G_{dist}(s)=\frac{K\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\ ,\ (0<\zeta<1) \qquad (5.3)$$

where ω_n is the natural (resonance) frequency of the process, ζ is called the relative damping [8] and *K* is a gain factor. We see that

$$k_{motor}=K m\omega_n^2,\ \delta=2m\zeta\omega_n \text{ and } \eta=m\omega_n^2.$$

The benefit of using Equation 5.3 to describe the process is that the natural frequency and the relative damping of the process can be accessed directly in the transfer function. The same discussion holds for describing vibrations in the belts that are strapped around the pulleys, i.e., a simple mass spring damper system with different parameters is fully sufficient. However, distortions does not take place in two dimensions as for a beam. All oscillations appear in the direction of the force is applied in.

5.2 Vibrations of the portal robot tool point

As already stated the portal robot consists of three beams with belts and pulleys. There can be several sources for vibrations. However the most apparent and critical ones are the Y-axis, the X-axis, the X-axis belt, the Y-axis belt and the Z-axis belt. The Z-axis is very rigid and well supported (see Figure 2.4) so it is not likely that it constitutes a source of vibrations worth mentioning.

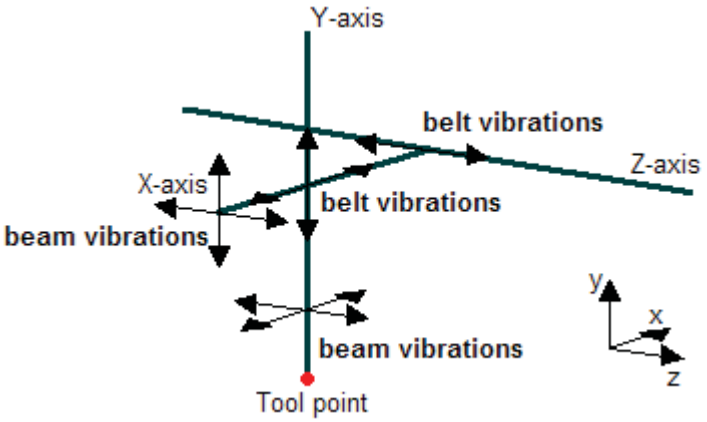

Figure 5.4. *The sources of vibrations for the tool point. The secondary component that bending of beams give rise to is neglected.*

Figure 5.4 shows the potential vibration sources that are worth considering. The secondary term (orthogonal to the primary term) that arises from bending of beams is neglected. This is because the latter component is very small and, as already stated in Section 5.1, it may be seen as a result of the first component. E.g. consider the X-axis beam vibrations when the X-axis is moved along the Z-axis. It is only the z_{dist} component that is illustrated in the figure, the smaller term in the *x* direction is neglected. Table 5.1 shows roughly how the vibrations in Figure 5.4 are connected to movements of the three servos.

		X-axis	Y-axis	Z-axis
Belt vibrations	x	X-axis servo	-	-
	y	-	Y-axis servo	-
	z	-	-	Z-axis servo
Beam vibrations	x	-	X-axis servo	-
	y	Y-axis servo	-	-
	z	Z-axis servo	Z-axis servo	-

Table 5.1. *A rough overview of the connections between the vibrations in Figure 5.4 and the servo motors. The Z-axis is considered very rigid and non-flexible.*

The characteristics of the X-axis and Y-axis vibrations will of course change

depending on the position of the tool point relative to the axes, i.e., the parameters K, ζ and ω_n in the corresponding transfer functions will change. E.g. the beam vibrations of the Y-axis will be different if the tool point is located at $y=0$ or if it is located at $y=y_1$ where $y_1<0 \in \mathbb{R}$.

It is in the opinion of the author, that an assumption that the principles of superposition may yield a fair approximation for the total distortion of the tool point as each axis distortion have wavelike behavior. Considering z_{dist} of the tool point this means roughly that

$$z_{dist}(t)=z_{dist}^{Y-axis\ beam}(t)+z_{dist}^{X-axis\ beam}(t)+z_{dist}^{Z-axis\ belt}(t) \qquad (5.4).$$

Looking at Table 5.1 and considering Equation 5.4 we see that the tool point distortions in the z direction are only induced by the Z-axis servo.

On the portal robot used for this project, each beam has different cross sections and characteristics. Figure 5.5 shows a close-up of the the X-axis and the Y-axis.

Figure 5.5. *A close-up of the X-axis to the left and the Y-axis to the right.*

The only case studied for this thesis is the case in Equation 5.4, i.e., distortions of the tool point along z induced by the Z-axis servo. This means that the bending force on the beams is applied in the z direction orthogonally to the other axes. The only resonance frequencies that have to be studied are those along z of the considered beam. Since the cross sections of the beams are such that they are non-symmetrical with respect to the center point, the resonance frequencies are different in two directions. The beam cross sections are symmetrical along the center planes. Even though the case of several axis movements have not been studied in this project, it is believed that the approximation below is fair since the distortions are small and no plastic deformation takes place.

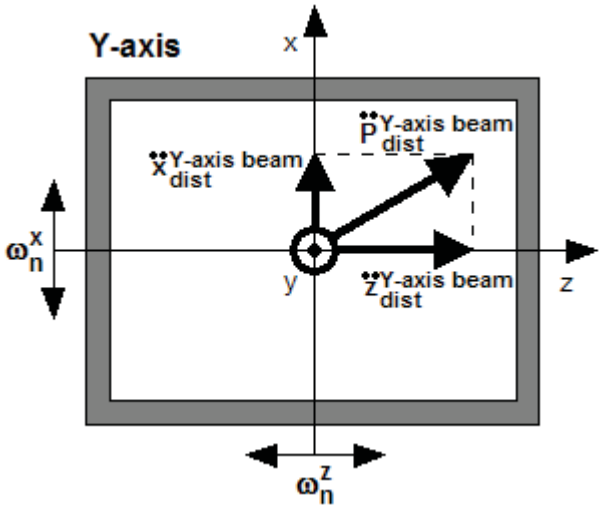

Figure 5.6. *A cross section of the Y-axis. The beam is exposed to a force from both the X-axis servo and the Z-axis servo.*

Figure 5.6 approximates the beam dynamics when the X-axis servo and the Z-axis servo accelerates or decelerates at the same time. It is assumed that the beam distortion $\boldsymbol{P}_{\boldsymbol{dist}}^{\boldsymbol{Y-axis\,beam}}$ under this condition can be described as Equation 5.5.

$$\boldsymbol{P}_{\boldsymbol{dist}}^{\boldsymbol{Y-axis\,beam}} = (x_{dist}^{Y-axis\,beam}, 0, z_{dist}^{Y-axis\,beam}) \qquad (5.5)$$

where $x_{dist}^{Y-axis\,beam}$ oscillates with the natural frequency of the beam in the x direction ω_n^x and $z_{dist}^{Y-axis\,beam}$ oscillates with the natural frequency of the beam in the z direction ω_n^z. This means that oscillations at the frequency ω_n^x are induced by the X-axis servo and that oscillations at the frequency ω_n^z are induced by the Z-axis servo. The reasoning above is however something that has to undergo practical experiments to be verified.

5.2.1 Measurements on the portal robot

As already mentioned, the focus in this project has been on distortions in the z direction according to Equation 5.4. Measurements have been taken in order to identify these distortions. A very important issue is if the vibrations can be observed in other signals than accelerometer signals.

For the tests, the tool point have been positioned at

$$\boldsymbol{P}_{\boldsymbol{tool}} = (-x_{max}, -y_{max}, k_z s_z + z_{dist}).$$

This means that the Z-axis servo is active and that the worst case scenario is investigated. Positioning the tool point at the maximum x and y coordinates yields the longest lever arms. The test conditions for the Y-axis can be illustrated as in Figure 5.7.

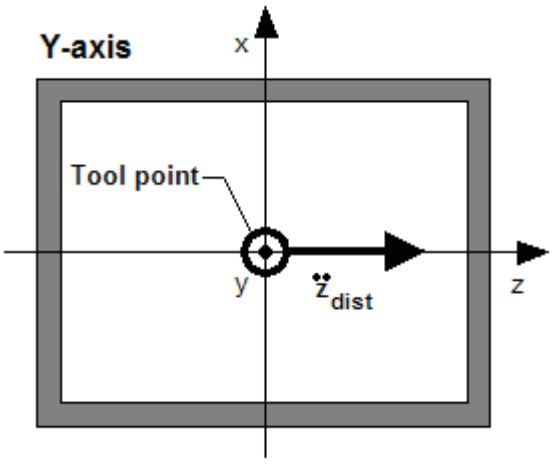

Figure 5.7. *Test conditions for the tool point.*

Consulting Equation 5.4 we see that z_{dist} roughly consists of three terms; $z_{dist}^{Y-axis\ beam}$, $z_{dist}^{X-axis\ beam}$ and $z_{dist}^{Z-axis\ belt}$. In other words bending of the Y-axis along *z,* bending of the X-axis along *z* and stretching of the Z-axis belt. It may be of interest from a mechanical perspective to investigate which term is superior. For this purpose two accelerometers have been attached to the robot; one on the tool point and another on the free end of the X-axis. Figure 5.8 shows the accelerometer measurements of the tool point and of the free end point of the X-axis $(x=-x_{max})$.

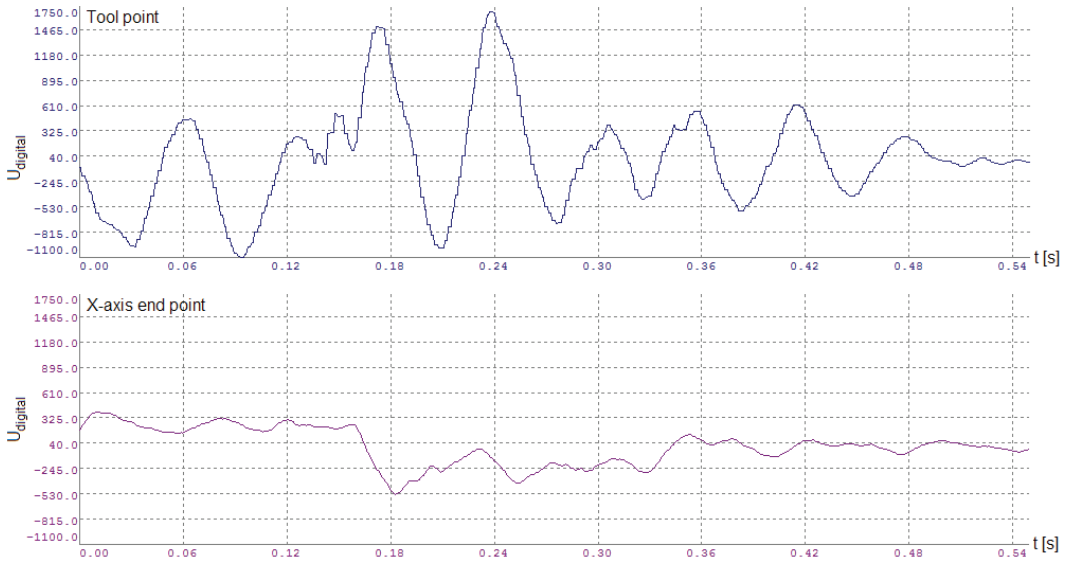

Figure 5.8. *Shows accelerometer measurements of the tool point and of the free end point of the X-axis.*

The results in Figure 5.8 suggest that the superior term of z_{dist} is $z_{dist}^{Y-axis\ beam}$, or in other words that the main distortion of the tool point position originates from bending of the beam. This can also be observed simply by looking at the process

when it operates. It is somewhat also expected since the Y-axis is much weaker than the X-axis.

The movement profile for the Z-axis servo used for obtaining the data in Figure 5.8 has been;

$$s = 144000\,units$$
$$v\_pos = 5 \cdot 10^6\,units/s$$
$$a1\_pos = 7 \cdot 10^6\,units/s^2$$
$$a2\_pos = 7 \cdot 10^6\,units/s^2$$

or in Z-axis coordinates (see Chapter 2, Subsection 2.1.1)

$$z_{ref} = 144000\,k_z = 0{,}2563\,m$$
$$\dot{z}_{ref} = 5 \cdot 10^6\,k_z = 8{,}9\,m/s$$
$$\ddot{z}_{ref} = \pm 7 \cdot 10^6\,k_z = \pm 12{,}46\,m/s^2.$$

The accelerometer signal $U_{digital}$ is given as

$$U_{digital} = U_{acc\ output} - U_{neutral\ digital}$$

(see Chapter 2, Section 2.7). As already stated, it is important to see if the vibrations can be seen in other available signals and not just in the accelerometer signals. The other signals can e.g. be the Z-axis servo stator current i_{stator} and the Z-axis servo position lag error $e_{position}$. For this purpose a profile with the same speed and acceleration set points, but with half the position set point as in Figure 5.8 have been used and traces were taken for i_{stator} and $e_{position}$. This profile has been identified to induce much vibration in the process. Figure 5.9 shows the measurements of i_{stator}, $e_{position}$ and $U_{digital}$. As we see in the figure, vibrations are observed in more signals than the accelerometer signals. A parameter that is of particular interest is the frequency, i.e., the natural frequency f_n of (mostly) the Y-axis. To obtain this frequency and to be certain that it is the same in all signals, Fast Fourier Transforms (FFTs) are conducted for the data in Figure 5.9. It could also be mentioned that vibrations are seen in more signals than the ones presented, e.g. they can be observed in the speed error e_{speed} and the motor torque T_{motor}. Since e_{speed} suffers from noise (differentiated signal) and since T_{motor} is calculated with the help of current and angle measurements (and therefore has approximately the same appearance as i_{stator}) those signals are not considered here. The FFTs in Figure 5.10 show peaks at approximately $14{,}15Hz$ which suggests that it is the same vibrations that are observed in all signals. This means that the natural frequency for the Y-axis in the z direction is $f_n = 14{,}15\,Hz$ or $\omega_n = 2\pi\,14{,}15\,rad/s = 88{,}9\,rad/s$ when $\boldsymbol{P}_{tool} = (-x_{max}, -y_{max}, k_z s_z + z_{dist})$.

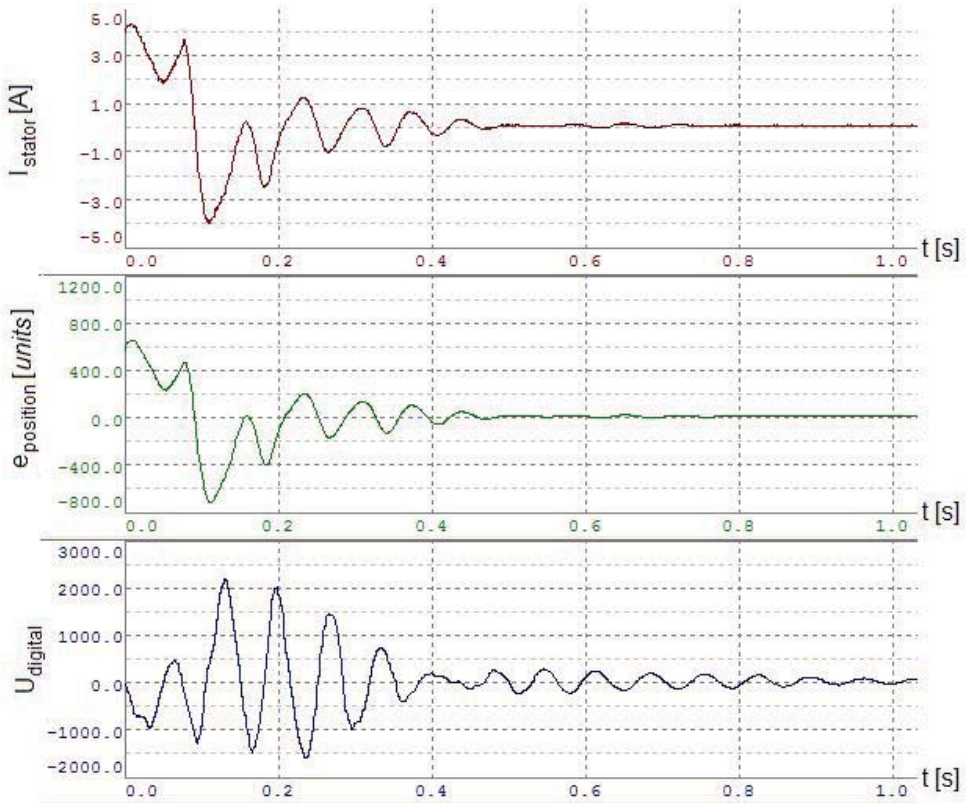

Figure 5.9. *Stator current, position lag error and tool point accelerometer measurements.*

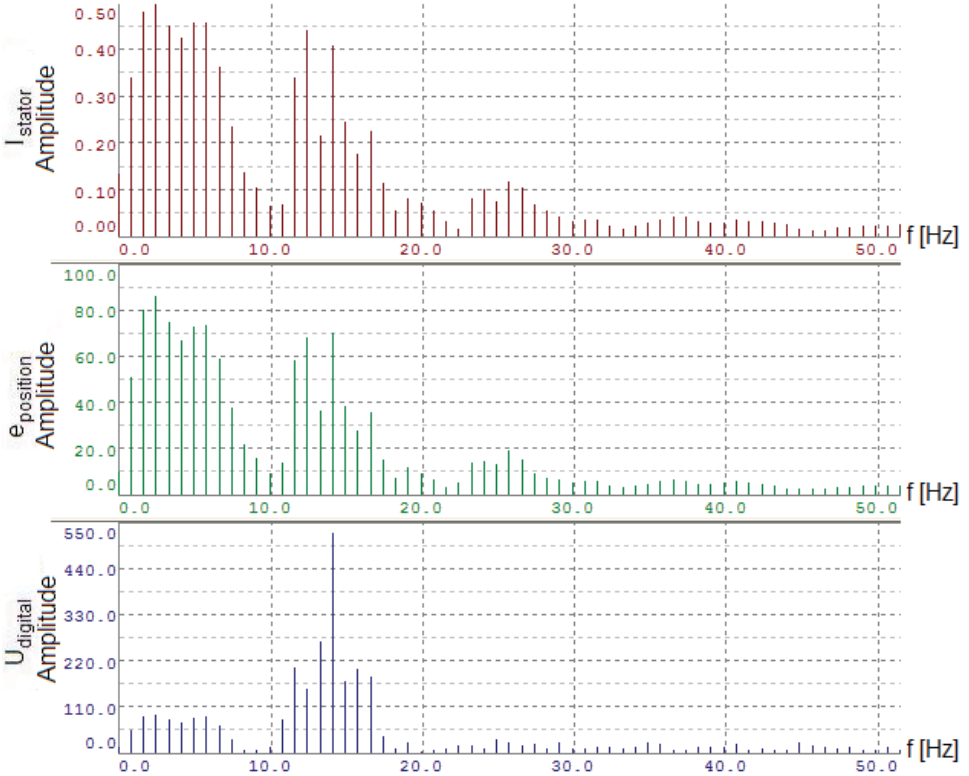

Figure 5.10. *FFTs of stator current, position lag error and accelerometer measurements.*

5.3 Vibrations of the experimental setup

The discussion in this section is in general very similar to the discussion about the portal robot tool point in Section 5.2. As the experimental setup consists of a belt and pulley system like the portal robot, the sources of vibrations are in principle the same. However, the process is much simpler since there is only one flexible beam and one belt to consider. The sources of vibrations are illustrated in Figure 5.11.

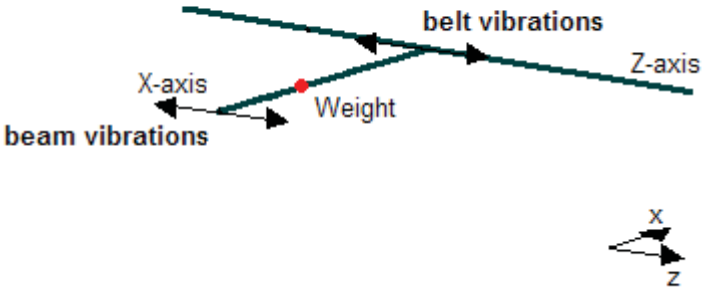

Figure 5.11. *The sources of vibrations for the experimental setup.*

For reasons already discussed, the secondary distortion component in the *x* direction that beam vibrations give rise to is neglected. It is of course only the movements of the Z-axis servo that induce vibrations. Furthermore we can consider the belt vibration term as very small, which is clearly seen during operation of the process. This is only of interest from a mechanical viewpoint, from a control perspective it is only of interest if the vibrations can be measured and what actuator induces them. Figure 5.12 shows a close-up of the X-axis.

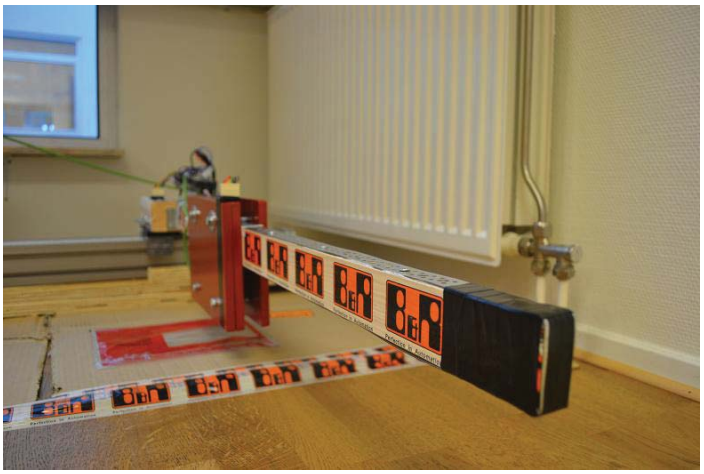

Figure 5.12. *A close-up of the X-axis.*

As the weight can be moved manually along the X-axis it is possible to change

the characteristics of the vibrations. In this presentation the weight has been fixed, i.e., we consider one transfer function here.

5.3.1 Measurements on the experimental setup

The position for the weight during the tests below can be described as

$$\boldsymbol{P}_{\boldsymbol{weight}} = (a\,,\, c_z s_z + z_{dist})$$

and as the belt vibration term is much smaller than the beam vibration term,

$$z_{dist} \approx z_{dist}^{X-axis\,beam}.$$

Figure 5.13 shows measurements of the stator current i_{stator}, the position lag error $e_{position}$ and the digital accelerometer voltage $U_{digital}$ (the accelerometer was attached to the weight). The movement profile for the Z-axis servo was

$$\begin{aligned} s &= 144000\,units \\ v\_pos &= 5\cdot 10^6\,units/s \\ a1\_pos &= 7\cdot 10^6\,units/s^2 \\ a2\_pos &= 7\cdot 10^6\,units/s^2 \end{aligned}$$

or in Z-axis coordinates (see Chapter 2, Subsection 2.2.1)

$$\begin{aligned} z_{ref} &= 144000\,c_z = 0{,}2722\,m \\ \dot{z}_{ref} &= 5\cdot 10^6\,c_z = 9{,}45\,m/s \\ \ddot{z}_{ref} &= \pm 7\cdot 10^6\,c_z = \pm 13{,}23\,m/s^2. \end{aligned}$$

As we see in Figure 5.13 vibrations are observed in both the stator current i_{stator} and the position lag error $e_{position}$ besides the accelerometer signal $U_{digital}$. This is analogous to the considered portal robot. What remains to investigate is the frequency of the vibrations, i.e., the natural frequency of the X-axis at the considered weight position. The FFTs of stator current, position lag error and accelerometer signal in Figure 5.14 show peaks at about 9*Hz*. This suggests that it is the same vibrations that are observed in all signals. In other words the natural frequency f_n in the *z* direction of the beam is $f_n = 9\,Hz$ or $\omega_n = 2\pi 9\,rad/s = 56{,}5\,rad/s$ at the considered weight position.

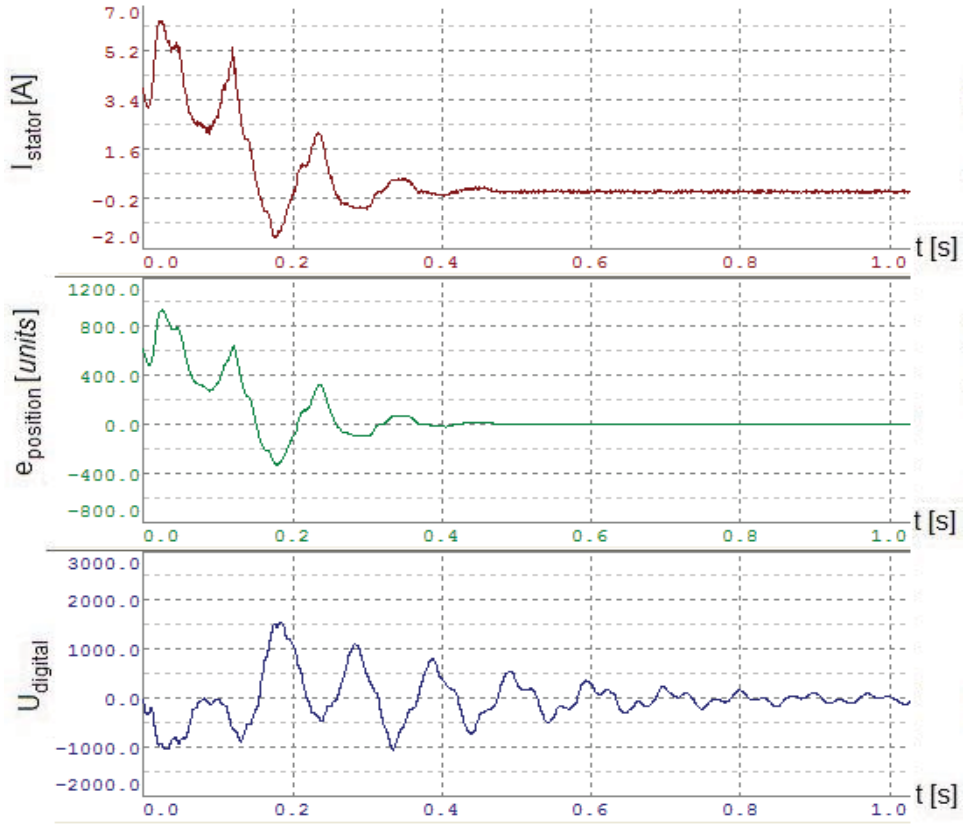

Figure 5.13. *Stator current, position lag error and weight accelerometer measurements.*

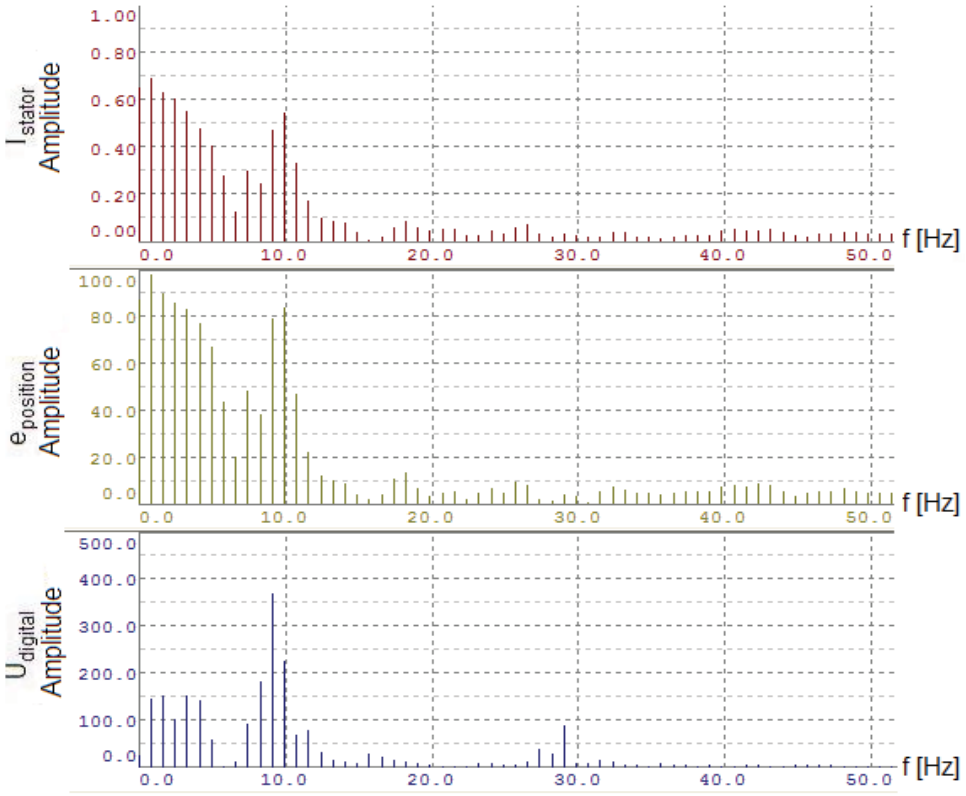

Figure 5.14. *FFTs of stator current, position lag error and accelerometer measurements.*

5.4 External vibrations in PID control loops

In Sections 5.2 and 5.3 we have seen that it is possible to observe the vibrations in the position lag error $e_{position}$ and the stator current i_{stator}. Furthermore it is said that the vibrations are seen in more servo drive signals than those presented, e.g. the motor torque T_{motor} and the speed error e_{speed}.

A question is whether this would be possible if the controlled signal followed the reference signal almost exactly (with a stiffly tuned controller)? The answer is of course no because e.g. the lag errors would then be close to zero at all times. If this were the case would the vibrations disappear? The answer here is of course also no, the vibrations due to bending and stretching would become even worse. This scenario could be described by the following example. Imagine holding a flyswatter and moving your arm very rapidly and then making an abrupt stop. The flyswatter would bend and vibrate before coming to a complete stop and you would probably not feel much of the vibrations. Now imagine that instead of holding the flyswatter with your hand you hold it between your thumb and index finger making the same movement. Not only will the vibrations be felt, they will probably also be reduced a bit since the fingers will give in to the vibrations. It is exactly the same case with a stiffly tuned PID controller and a softly tuned PID controller. The softly tuned controller "sees" the vibrations as it gives in to them a bit whereas the stiffly tuned controller makes the exact opposite. As a conclusion we may say that we want good reference tracking of the servos that a stiffly tuned controller yields, and at the same time we want a low amount of mechanical vibrations that a softly tuned controller yields.

Chapter 6 describes various control strategies. In S. L. Dickerson et al. [10] the analogy between a physical spring-damper system and a PD position controller is made. An example is given with position feedback control of a mass-spring system where the context is very similar to the topic of this thesis. In the example the spring is compared with the P part of the controller (proportional to the deviation) and the damper is compared with the D part of the controller (proportional to the derivative of the deviation). Using the same basic idea but applying it to the example given in this presentation, we see why a softly tuned controller is preferable when it comes to identifying the mechanical vibrations with servo drive signals. Why on the other hand, a stiffly tuned controller is preferable when it comes to identifying the mechanical vibrations with e.g. an accelerometer. The servo drive controller is of course not as simple as a conventional PD or PID controller, it consists of cascaded PI controllers with reference feed forward to the speed controller as we have seen in Chapter 4.

The aim of this section has only been to give an intuitive understanding for the coupling between servo motor control and mechanical vibrations.

6. Control strategy

In this chapter various control strategies are presented. The aim is to motivate why position reference filtering is preferable compared to other strategies.

As already stated in Chapter 5 the focus of this thesis has been on one axis only, the discussion will therefore be held with this in mind. In industry there is always a request to keep the amount of sensors down in a process. This is natural because sensors increase the economic cost and also because sensors can break and cause expensive secondary faults, as discussed in Chapter 4. The desire to decrease the number of sensors leads to a demand of mathematical models or other mathematical methods to replace the loss of information that a sensor would give. This demand has been present also for this project, i.e., no accelerometers or strain gauges should be involved for the vibration control and this leaves us with the servo drive signals as the only signals to be utilized. This fact is important. In Chapter 5 it is stated that stiff control leads to the situation where axis vibrations are not observed as much in the lag errors as they are with soft control. A contradiction arises which has to be considered when choosing control method. Figure 6.1 shows position lag error $e_{position}$ data gathered from the experimental setup for two different position controller proportional amplifications. It is seen that the vibrations are observed less with the more stiffly tuned controller.

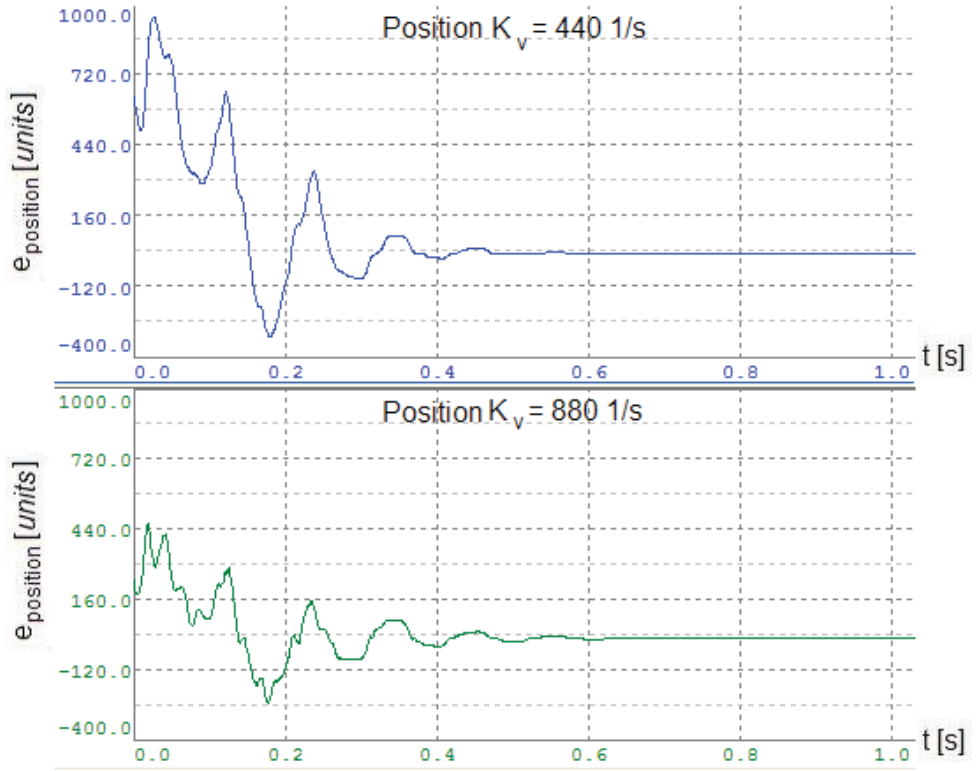

Figure 6.1. *Position lag error data from the experimental setup at two different proportional amplifications in the position controller.*

The fact that the "real" arm vibrations cannot be observed leads to the conclusion that if a feedback based method should be used, a model of the vibrations is needed. In S. L. Dickerson et al. [10] an overall process block diagram for the

servos and the free endpoint of one axis is described as in Figure 6.2 (the figure is slightly changed in this presentation).

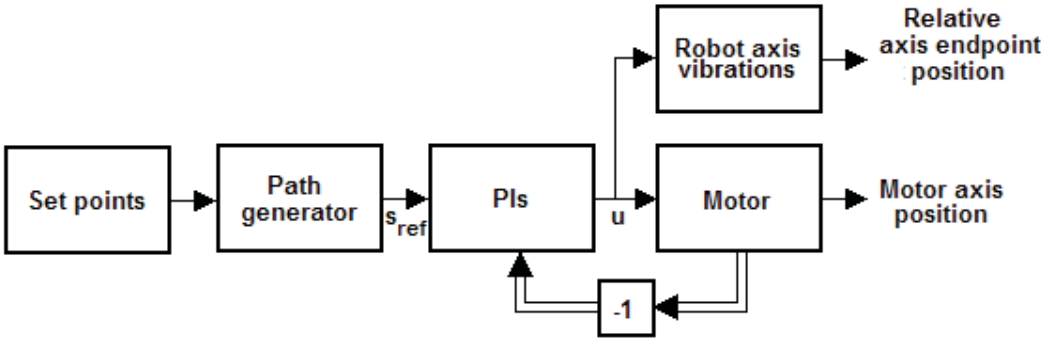

Figure 6.2. *Overall process block diagram. Note that stiff control is assumed since the motor axis position is unaffected by the robot axis position in the figure.*

Several potential control strategies exist. They can be divided in to three main categories: feedback control, feed forward control and command shaping. As is also stated in S. L. Dickerson et al. [10] feedback control is many times a good idea since it allows active damping of the vibrations. Reference feed forward and command shaping can only prevent vibrations from occurring. To design a totally new controller is not realistic as it would exceed the scope of this thesis. The servo drive control structure has lots of implemented functionalities that would be to cumbersome to adapt to a new controller and good performance of such a controller would still not be certain. Moreover, when it comes to servo control, cascaded PIs are well established in industry and have proven to work very well in most applications. Focus in this thesis lies therefore on the possibilities to modify the existing control structure so that it takes the mechanical vibrations into account (with no additional sensors).

6.1 Model based feedback control

As already mentioned, the main benefit of using feedback control is that the vibrations can be actively damped. This means that if e.g. a load disturbance occurs the vibrations that the load induces can be treated by the controller. A drawback with feedback control is that an error has to be present in order for the controller to react. How model based feedback control can be accomplished is illustrated in Figure 6.3. In the controller design, feedback is done from the estimated axis endpoint position vibrations. A good idea could of course be to also have feedback of speed vibrations to the speed loop and feedback of acceleration vibrations to the current loop. However, the purpose with this controller is just to make a general investigation of model based feedback control. The question at this stage is how the model should be constructed.

Considering vibrations in one dimension of space, a simple mass-spring-damper model can describe oscillations with one frequency component very well. In Section 5.1 we see that the transfer function G_{dist} between torque and position of such a process can be written on the conventional form

$$G_{dist}(s)=\frac{K\,\omega_n^2}{s^2+2\zeta\,\omega_n s+\omega_n^2}.$$

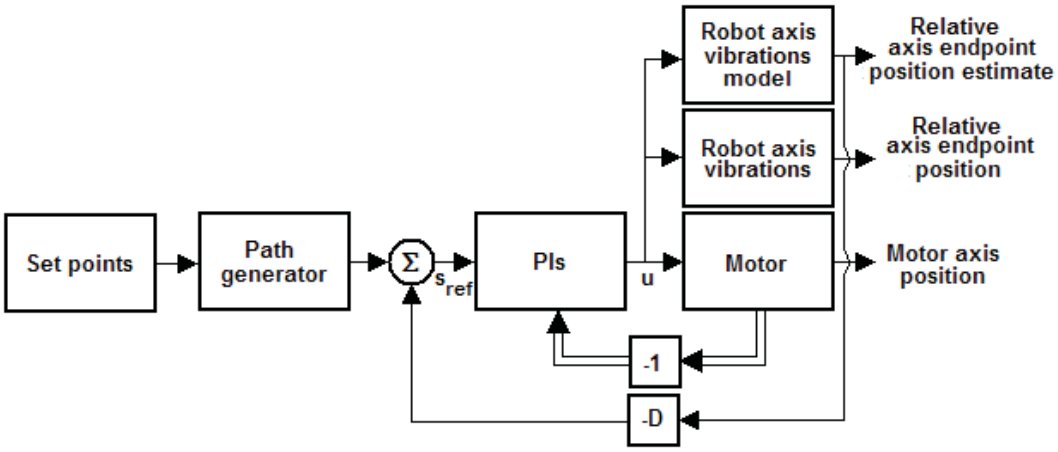

Figure 6.3. *Model based feedback control of the mechanical vibration position. D is the state feedback factor.*

This model could then be used in the block "Robot axis endpoint model" in Figure 6.3. Note that in the figure, for illustrative reasons, the input to the model is the PWM voltage. It may as well be the actual stator current instead. Even though the current-torque relationship for a synchronous motor is not as easy formulated as it is for a DC motor, the stator current i_{stator} and the motor torque T_{motor} have approximately the same appearance except for a factor in the considered signal range. It is therefore fair to use i_{stator} as input to the model G_{dist}. The estimated position distortion $\hat{Z}_{dist}(s)$ of the free axis endpoint can thus be obtained by Equation 6.1,

$$\hat{Z}_{dist}(s)=\frac{K\,\omega_n^2}{s^2+2\zeta\,\omega_n s+\omega_n^2}I_{stator}(s) \qquad (6.1).$$

As we have seen it is no problem to obtain the structure of the transfer function for the vibrations. The problems arise with the determination of transfer function parameters. It is no problem to obtain ω_n, this we have seen in Chapter 5. For the portal robot $\omega_n=88{,}9\,rad/s$ and for the experimental setup $\omega_n=56{,}5\,rad/s$ at the considered axis position configurations. But how should the parameters K and ζ be determined?

Having the structure of the transfer function a so called grey box identification of the process could be made. Grey box identification is essentially used for parameter identification whereas black box identification is used when the whole process is unknown. The problem is, however, that since no additional sensors are desired no sufficient z_{dist} data can be obtained for e.g. linear least squares estimation of the parameters in a discrete time model of Equation 6.1. The closest we get to z_{dist} is the servo position lag error $e_{position}$ and the problem is

that the vibrations are only observed in $e_{position}$, the lag error does not represent z_{dist}. One simple solution could be to manually isolate the vibrations in the lag error signal and adjust the parameters in Equation 6.1 so that $\hat{z}_{dist}$ corresponds to the isolated signal as well as possible. This solution would only serve to test the feedback idea as the process parameters may be different for various operating states, loads etc. Another aspect is that a more general solution may be desired in industry, since servo drives are used in many different applications where vibrations occur. In S. L. Dickerson et al. [10] it is said about the feedback method that a good mathematical model of the process is needed and it is also stated that we would have a hard time constructing a fast model for feedback. Why a good model is needed is clear; there is no real data that can be used for compensation of model errors. It would be to cumbersome to create a new model for every new application where servo drives are included. However, an approximative model has been created for the considered portal robot merely to test the feedback idea. The model was created by means of the proposed manual method. This means that the vibrations were isolated in the position lag error and the parameters were then adjusted so that a fairly good match between the model output and the lag error vibrations was obtained. The reference signal used for obtaining the lag error data was

$$
\begin{aligned}
s &= 144000\, units \\
v\_pos &= 5 \cdot 10^6\, units/s \\
a1\_pos &= 7 \cdot 10^6\, units/s^2 \\
a2\_pos &= 7 \cdot 10^6\, units/s^2
\end{aligned}
$$

moreover the controllers were very stiffly tuned so that the mechanical vibrations could be easily observed on the robot axes.

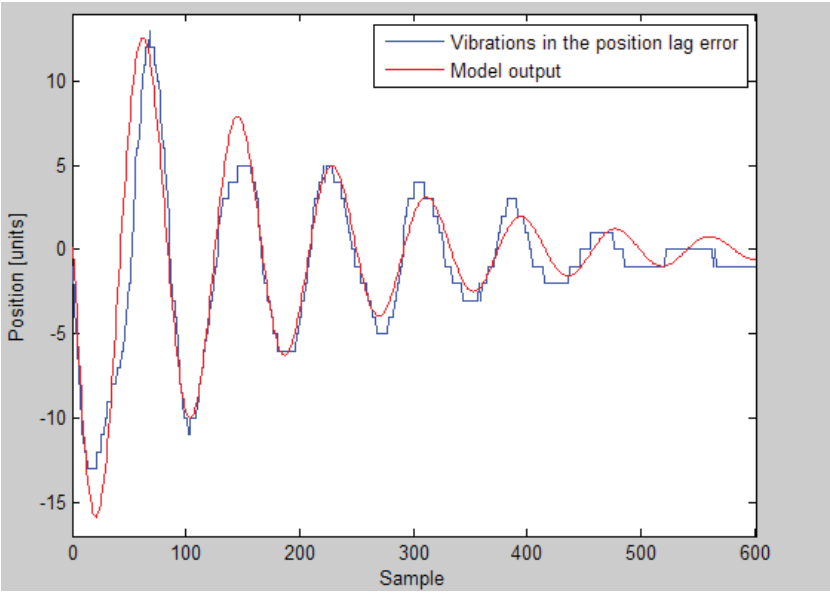

Figure 6.4. *Comparison between model output and vibrations observed in the lag error.*

Note that the position scale in Figure 6.4, compared with the scale in Figures 5.9 and 5.13, is vastly lower. This has to do with the stiffly tuned controllers. The model was simulated in Matlab Simulink and the parameters were adjusted in Matlab until the appearance in Figure 6.4 was obtained. The transfer function is found in Equation 6.2.

$$G_{dist} = \frac{231905}{s^2 + 14\,s + 9000} \qquad (6.2)$$

6.1.1 Feedback model discretization

In order to implement the model in a real system it has to be discretized due to the sampled nature of a PLC. The system in Equation 6.1 can be written on the state-space form of Equation 6.3 introducing the states $x_1 = \dot{\hat{z}}_{dist}$ and $x_2 = \hat{z}_{dist}$.

$$\begin{aligned} \dot{x} &= A\,x + B\,i_{stator} \\ \hat{z}_{dist} &= C\,x \end{aligned} \qquad (6.3)$$

where $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $A = \begin{pmatrix} -2\zeta\omega_n & -\omega_n^2 \\ 1 & 0 \end{pmatrix}$, $B = K\begin{pmatrix} \omega_n^2 \\ 0 \end{pmatrix}$ and $C = \begin{pmatrix} 0 & 1 \end{pmatrix}$. The discrete-time counterpart of the system in Equation 6.3 can be written as Equation 6.4.

$$\begin{aligned} x(kh+h) &= \Phi\,x(kh) + \Gamma\,i_{stator}(kh) \\ \hat{z}_{dist}(kh) &= C\,x(kh) \end{aligned} \qquad (6.4)$$

where k is the time instant and h is the sample time (task class cycle time in the PLC). Using the zero-order hold (ZOH) method for sampling the system yields that

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{At}\,dt\,B \end{aligned}$$

or equivalently

$$\Phi = \boldsymbol{L^{-1}}(sI - A)^{-1} = \boldsymbol{L^{-1}}\begin{pmatrix} s + 2\zeta\omega_n & \omega_n^2 \\ -1 & s \end{pmatrix}^{-1} = \boldsymbol{L^{-1}}\frac{1}{s^2 + 2\zeta\omega_n + \omega_n^2}\begin{pmatrix} s & -\omega_n^2 \\ 1 & s + 2\zeta\omega_n \end{pmatrix}$$

and

$$\Gamma = \int_0^h \boldsymbol{L^{-1}}(sI - A)\,dt\,B = \int_0^h \boldsymbol{L^{-1}} \frac{1}{s^2 + 2\zeta\,\omega_n + \omega_n^2} \begin{pmatrix} s & -\omega_n^2 \\ 1 & s + 2\zeta\,\omega_n \end{pmatrix} dt\,K \begin{pmatrix} \omega_n^2 \\ 0 \end{pmatrix}.$$

Executing the ZOH conversion in Matlab with the parameters in Equation 6.2 and $h = 0{,}8\,ms$, the resulting system matrices become

$$\Phi = \begin{pmatrix} 0{,}986 & -7{,}153 \\ 0{,}0007948 & 0{,}9971 \end{pmatrix},\ \Gamma = \begin{pmatrix} 184{,}3 \\ 0{,}0739 \end{pmatrix} \text{ and } C = \begin{pmatrix} 0 & 1 \end{pmatrix}.$$

Thus the transfer function $H_{dist}(z)$ from the sampled current $i_{stator,\,k}$ to the sampled distortion estimate $\hat{z}_{dist,\,k}$ can be expressed as Equation 6.5.

$$H_{dist}(z) = \frac{0{,}0739\,z + 0{,}07362}{z^2 - 1{,}983\,z + 0{,}9889} \qquad (6.5).$$

With the help of Equation 6.5 an expression for calculating the distortion estimate with the PLC is given as Equation 6.6.

$$\hat{z}_{dist,\,k+1} = 0{,}0739\,i_{stator,\,k} + 0{,}07362\,i_{stator,\,k-1} + 1{,}983\,\hat{z}_{dist,\,k} - 0{,}9889\,\hat{z}_{dist,\,k-1} \qquad (6.6).$$

6.1.2 Feedback experiments and conclusions

Initial experiments with feedback control were conducted on the portal robot with the exact same profile that was used for modeling, i.e.,

$$\begin{aligned} s &= 144000\,units \\ v\_pos &= 5 \cdot 10^6\,units/s \\ a1\_pos &= 7 \cdot 10^6\,units/s^2 \\ a2\_pos &= 7 \cdot 10^6\,units/s^2. \end{aligned}$$

More about the implementation is described in Chapter 7. Figure 6.5 shows accelerometer data for the process without feedback and with feedback. During these experiments the accelerometers were attached to the free endpoint of the X-axis, the Y-axis vibrations can still be observed. Even though the signal with feedback does not look that damped compared to the signal without feedback, it is seen on the Y-axis that there are less vibrations. This can also be observed in the FFTs in Figure 6.6 for the accelerometer data. The state feedback coefficient was set to $D = 5$. That means that the signal subtracted from the position reference every cycle was given by

$$5\,\hat{z}_{dist,\,k+1} = 5(0{,}0739\,i_{stator,\,k} + 0{,}07362\,i_{stator,\,k-1} + 1{,}983\,\hat{z}_{dist,\,k} - 0{,}9889\,\hat{z}_{dist,\,k-1}).$$

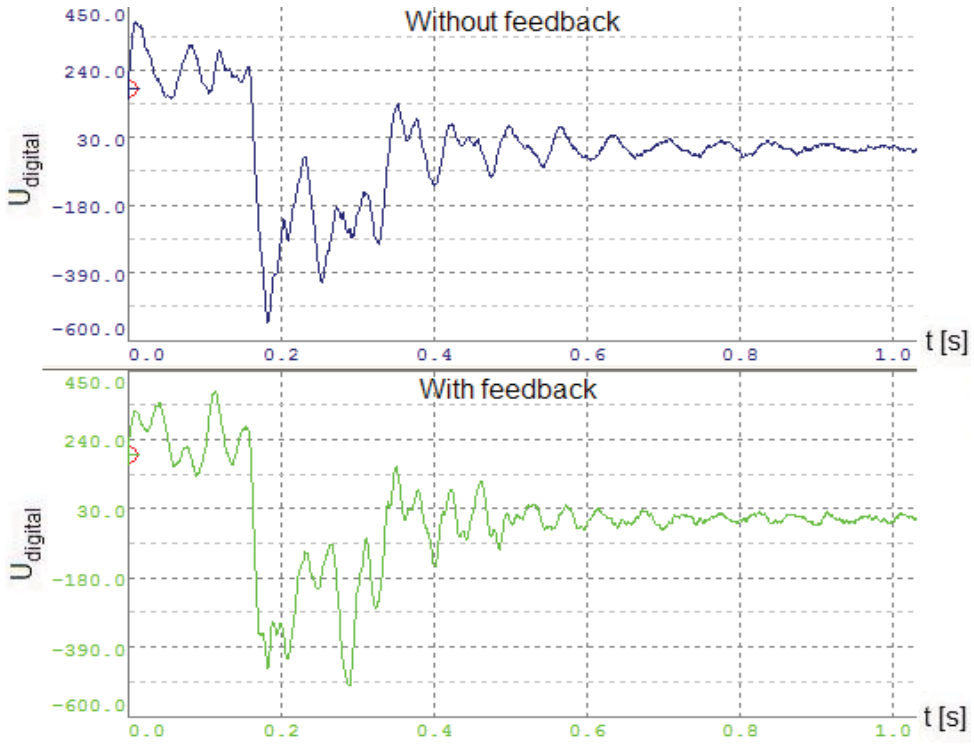

Figure 6.5. *Accelerometer data for the process without feedback and with feedback.*

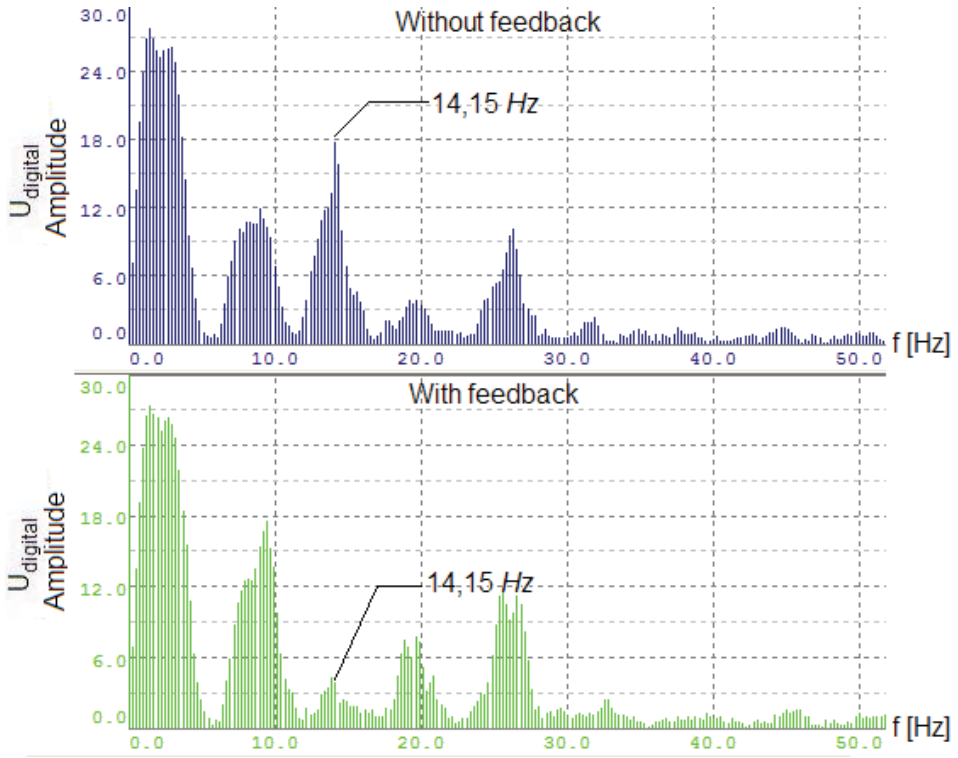

Figure 6.6. *FFTs of accelerometer data.*

In the FFTs in Figure 6.6 we see that the peak at 14,15*Hz* is damped. Does this mean that feedback of distortion estimates is sufficient? The answer is no, the problems discussed earlier remain. Several profiles have been tested with the same model. For some profiles feedback control works quite well but for others it is not satisfying at all. For those profiles the feedback method may even induce more vibrations in the process and this is of course because the model error is too large. Since it is hard to create a good general model based feedback method and since real distortion data is not available, it may be wise to consider another solution than feedback. The main loss is that vibrations can not be actively damped with another solution.

6.2 Feed forward control

A feed forward strategy for the considered processes could be to pass the reference signal to a model and then pass feed forward terms to control variables. Figure 6.7 shows a block diagram of the process with feed forward applied.

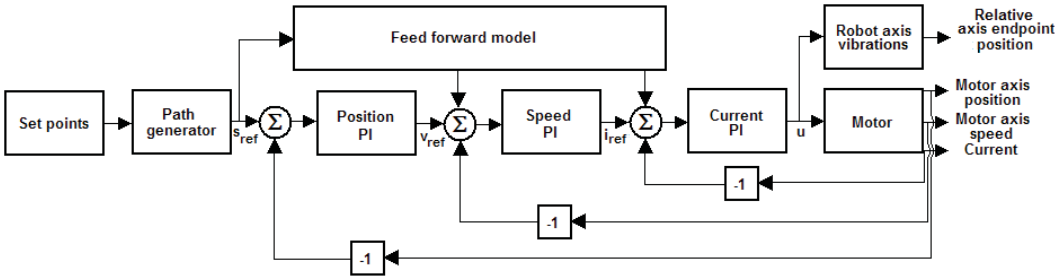

Figure 6.7. *Overall process block diagram with a feed forward structure. Note that a simplistic approach is taken to the current controller.*

The feed forward structure in Figure 6.7 can be used for making servo systems faster and optimized [7]. In the considered processes, the feed forward terms would have to compensate for mechanical vibrations by changing the control variables before the vibrations occur. In either way a good model of the process would be needed and that leaves us facing the same problems as we do with feedback control. It should be possible to adapt a robust control strategy for a variety of robots at different operating states. In addition, as have already been stressed many times, no extra sensors are desired.

6.3 Command shaping

Vibrations in the process are induced by the servo motors, or more exactly through the reference signal to the servos. An analogy could be made by carrying a cup of coffee from the kitchen to your desk. Imagine being stressed having to finish a paper by a certain date, you grab your cup and walk towards your desk without thinking about the coffee in the cup. The well deserved coffee ends up on the floor (hopefully not on a carpet) and somebody else in the room starts to yell at you, calling you clumsy etc. and finally says that you should be more careful. What that the other person really wants to say is that you should consider the natural frequency of the coffee in the cup and try not to move the cup with that frequency too much. The same basic idea could apply for the mechanical

vibrations. I.e., by telling the servos not to induce the natural frequency of the vibrations, less vibrations will appear.

The question you ask the other person may be: how should I know the natural frequency of the coffee? The question is justified. Some a priori knowledge is needed in order for you to move fast to the desk without spilling out the coffee.

As we have seen in Chapter 5, by looking at available servo drive signals we can determine the natural frequency of the undesired oscillations. This section deals with how we can use that fact to prevent mechanical vibrations from occurring.

6.3.1 Jolt limitation

The Path Generator in the ACOPOS is the command shaper at hand. With speed and acceleration references it creates an "S-like" position reference from step set points. However, in normal operation it does not take process resonances into consideration while shaping the reference curve. As already mentioned in Chapter 4, the standard remedy to prevent mechanical vibrations is so called jolt limitation. By reshaping the acceleration set point so that it obtains a trapezoid appearance instead of a step appearance, the resulting position reference becomes smoother to follow. The method is based on a rule of thumb: the jolt filter time t_jolt is set equal to the period time of the undesired vibration (t_jolt defines the time it takes for the acceleration to reach its final value). The question is if it is possible to create an algorithm that gives better performance than jolt limitation.

6.3.2 Notch filter

Telling the servos not to induce vibrations in the rest of the process can be done by oppressing the frequency content at the natural frequency of the mechanics in the reference signal. This may be realized by adding a notch filter between the Path Generator and the position PI. Figure 6.8 shows a block diagram of the process with a notch filter applied as proposed.

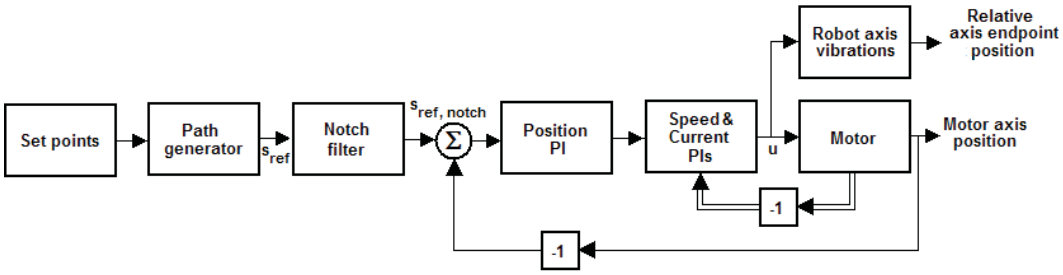

Figure 6.8. *Overall process block diagram with a notch filter implemented between the Path Generator and the position PI.*

A notch filter removes signal contents, or in other words signal energy. The effects are that critical energy is not induced and at the same time the notched profile gets slower than the initial signal. It is the same scenario with jolt limitation, as mentioned in Chapter 4 the limited profile gets slowed down by t_jolt. The gain with the two methods (even though slower profiles are generated) is that the time it takes to wait out transients decreases. The question is if the

transients are longer than the increased time it takes for a treated profile to reach its final value.

6.4 Notch filter design

The reason for implementing notch filters is that a robust and general control design with few adjustable parameters is desired as there is little a priori knowledge about the process. We know the frequency of the vibrations and we wish to tell how much energy at this frequency that should be removed from the reference signal. In the Laplace domain, a typical transfer function for a notch filter between two signals is given as a fraction between two second order polynomials as in Equation 6.7.

$$F_{notch}(s)=\frac{s^2+2\frac{\omega_n}{q}s+\omega_n^2}{s^2+2\frac{\omega_n r}{q}s+\omega_n^2} \qquad (6.7)$$

where q and r are adjustable parameters. We have to be careful choosing these parameters so that a desirable filter is obtained. A desirable filter, a filter that is attenuating, is fast and non-oscillatory. Looking at the denominator of Equation 6.7, the poles are given by Equation 6.8.

$$s=-\frac{r}{q}\omega_n\pm\sqrt{(\frac{r^2}{q^2}-1)}\omega_n \qquad (6.8).$$

From the first term in Equation 6.8 we observe that as long as $\{(r,q)\in\mathbb{R}^2: r>0, q>0\}$ or $\{(r,q)\in\mathbb{R}^2: r<0, q<0\}$, i.e., as long as r and q are positive or negative real numbers at the same instant the stability criterion is satisfied. However, in order to get a desired phase behavior of the filter, i.e., no phase distortion at more frequencies than the ones close to the notch frequency we only consider parameters $\{(r,q)\in\mathbb{R}^2: r>0, q>0\}$. If $r<1$ an undesirable amplification instead of a desirable notch is obtained by the filter. Given this, in order to get the characteristic notch and not an amplification or unity, $r\gg 1$. If $r\gg q$ the resulting filter has the characteristics of a band-reject filter. Another situation that remains to investigate is $r<q$. Consulting Equation 6.8, it is observed that for values $\frac{r}{q}<1$ the pole map will have the appearance as in Figure 6.9. Figure 6.9 indicates that the filter output will oscillate as we see by the imaginary term of the poles. The vector from the origin to the pole (and the vector from the origin to the corresponding mirror pole), will be of length $2\pi f_n$. To get rid of filter induced oscillations, the imaginary term of the poles has to be cancelled. In other words a double pole on the negative real axis is desired. Consulting Equation 6.8 it is seen that this applies when $r=q$. The filter transfer function can thus be rewritten as Equation 6.9.

$$F_{notch}(s)=\frac{s^2+2\frac{\omega_n}{q}s+\omega_n^2}{s^2+2\,\omega_n s+\omega_n^2}=\frac{s^2+2\frac{\omega_n}{q}s+\omega_n^2}{(s+\omega_n)^2} \qquad (6.9).$$

This kind of filter outputs a fast, stable and non-oscillatory response. Table 6.1 is a summary of how the filter parameters should be chosen.

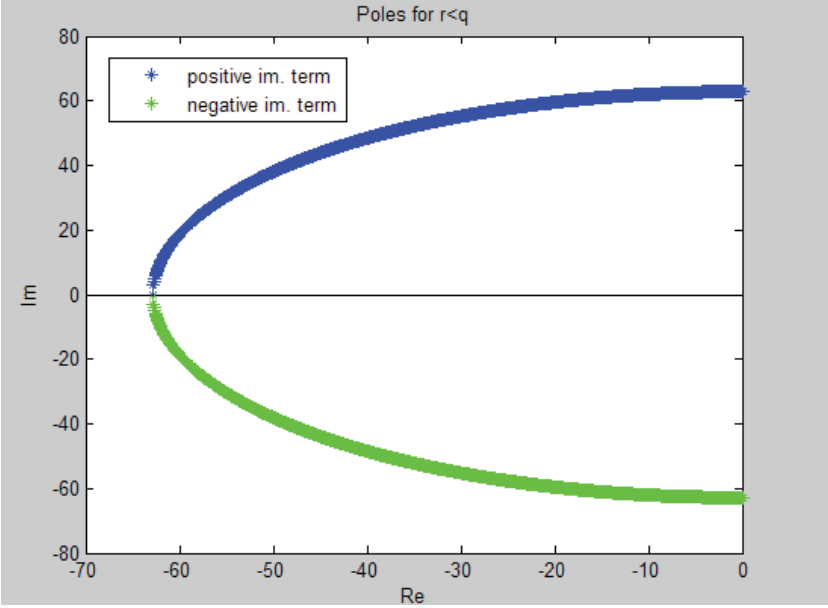

Figure 6.9. *The poles given by Equation 6.8 with* $f_n=10\,Hz$, *swept at different values* $0<r/q<1$. *The length of the vector (and, of course, its mirror image) is always* $2\pi f_n\approx 62{,}8\,rad/s$. *Note that* $r\gg 1$ *even though it is not understood by this plot.*

Parameters for the filter in Equation 6.7	Description
$\omega_n=2\pi f_n$	$f_n>0$, the undesired natural frequency.
$r=q$	For a non-oscillating filter response.
$q>0$	For a desired phase behavior.
$q\gg 1$	To get a notch instead of unity or an amplification.
This means:	$\omega_n=2\pi f_n>0,\ r=q\wedge q\gg 1$.

Table 6.1. *The filter parameters should be chosen according to the table in order to get a good and safe notch filter.*

Figure 6.10 shows the Bode diagram of a notch filter according to Table 6.1 with $f_n=10\,Hz$ and $q=600$.

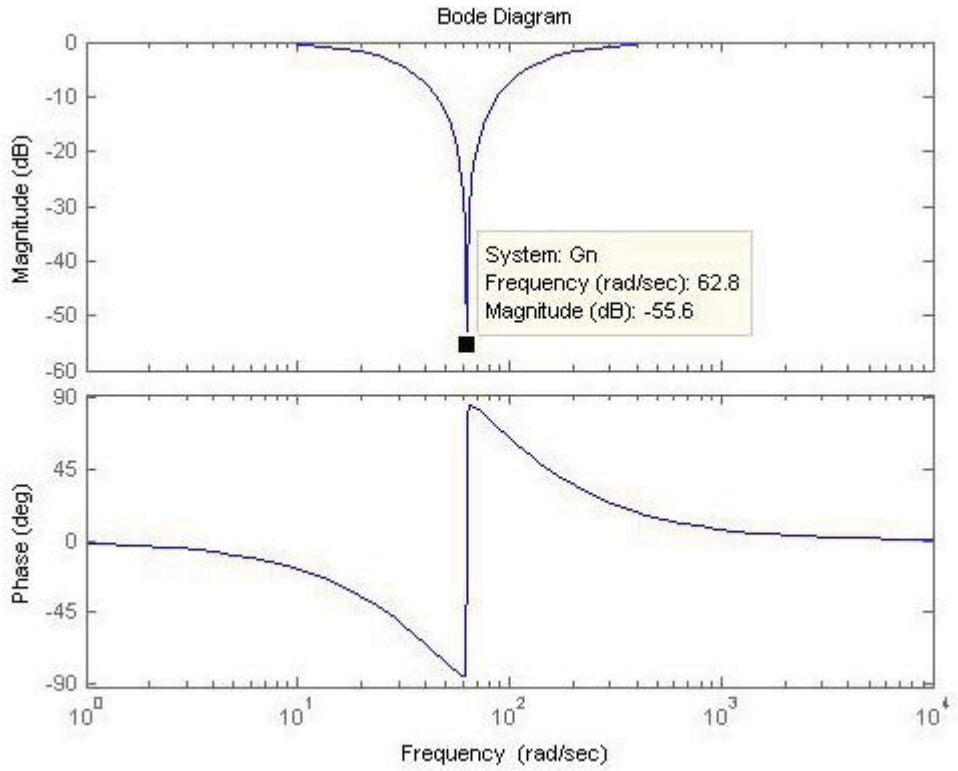

Figure 6.10. *The Bode diagram of a notch filter with* $f_n = 10\,Hz$, $q = 600$ *and thus* $\omega_n = 2\pi f_n \approx 62{,}8\,rad/s$.

A transfer function expressed in the Laplace domain is obviously not implementable in our case due to the sampled nature of a microprocessor. The filter must be expressed with the z-transform in order to be useful.

6.4.1 Discretization and discretization aspects

When implementing the notch filter in a real process we want it to be easy to set damping parameter *q* and notch frequency f_n. In other words the parameters of the discrete-time transfer function should be expressed in parameters of the continuous transfer function and the sample time *h*. The discrete-time transfer function parameters should then be calculated in a C program so that the desired filter is obtained. Exact discretization is many times cumbersome to do (matrix exponentials and integral operations). Different approximation methods exist for translating the Laplace transform to the z-transform: the forward difference, the backward difference and Tustin's approximation. Special care must be taken choosing the method.

The forward difference can transform a stable continuous-time system into an unstable sampled system. The backward difference will always yield a stable sampled system from a stable continuous-time system. There are however also unstable continuous-time systems which will appear stable in the sampled domain, when using the backward difference. Tustin's approximation will always map the stability regions of the two domains correspondingly [11].

Figure 6.11 illustrates the Laplace transform – z-transform pole stability region mappings with the different methods (see also B. Wittenmark et al. [11]).

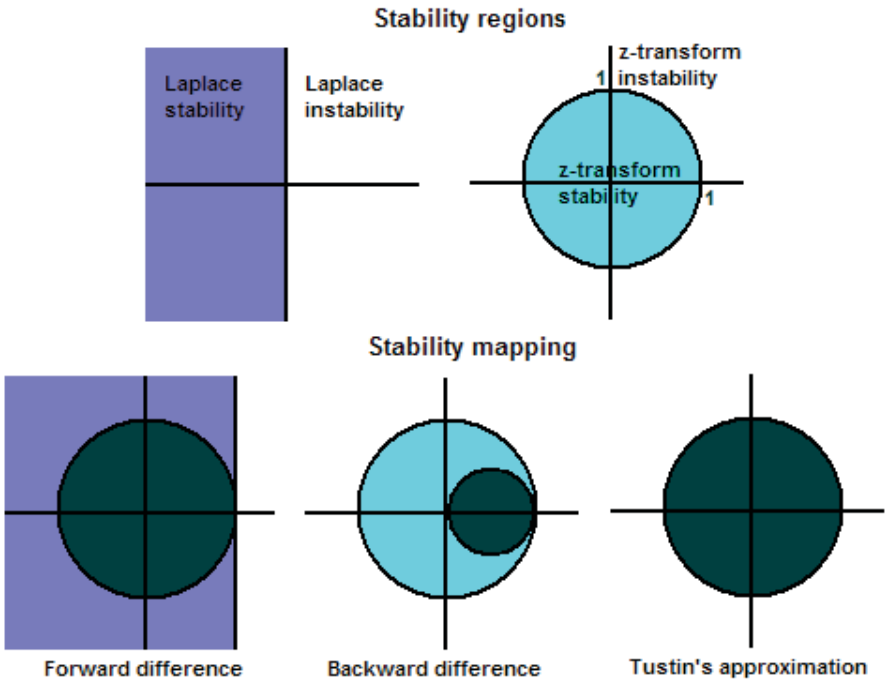

Figure 6.11. *The stability regions for poles within the Laplace- and z-transforms. The mappings of the two different regions are also illustrated by the figure.*

The three approximation methods are expressed as:

$$s=\frac{z-1}{h} \quad \textit{Forward difference}$$

$$s=\frac{z-1}{zh} \quad \textit{Backward difference}$$

$$s=\frac{2(z-1)}{h(z+1)} \quad \textit{Tustin's approximation}.$$

From Figure 6.11 we understand that Tustin's approximation is the safest choice for translating the transfer function expressed by Equation 6.9. The forward difference method should be avoided since one would like to have the possibility of changing all the filter parameters including the sample time *h*, without making the filter unstable. The z-transform translation of Equation 6.9 is expressed as Equation 6.10 (Tustin's approximation has been used).

$$H_{notch}(z)=\frac{B(z)}{A(z)}=\frac{b_0z^2+b_1z+b_2}{a_0z^2+a_1z+a_2} \qquad (6.10)$$

where

$$b_0 = 4 + 4\frac{\omega_n}{q}h + \omega_n^2 h^2, \quad b_1 = 2\omega_n^2 h^2 - 8, \quad b_2 = \omega_n^2 h^2 - 4\frac{\omega_n}{q}h + 4,$$
$$a_0 = 4 + 4\omega_n h + \omega_n^2 h^2, \quad a_1 = 2\omega_n^2 h^2 - 8, \quad a_2 = \omega_n^2 h^2 - 4\omega_n h + 4.$$

The algorithm for the filter could be implemented in a computer program as stated in Equation 6.11.

$$\Xi^{filtered} = \frac{b_0 z^2 + b_1 z + b_2}{a_0 z^2 + a_1 z + a_2}\Xi \quad \Leftrightarrow \quad (a_0 z^2 + a_1 z + a_2)\Xi^{filtered} = (b_0 z^2 + b_1 z + b_2)\Xi \quad \Rightarrow$$

$$a_0 \xi_{k+2}^{filtered} + a_1 \xi_{k+1}^{filtered} + a_2 \xi_k^{filtered} = b_0 \xi_{k+2} + b_1 \xi_{k+1} + b_2 \xi_k \quad \Rightarrow$$

$$\xi_k^{filtered} = \frac{-a_1 \xi_{k-1}^{filtered} - a_2 \xi_{k-2}^{filtered} + b_0 \xi_k + b_1 \xi_{k-1} + b_2 \xi_{k-2}}{a_0} \qquad (6.11).$$

ξ is the signal that the filter has its effect upon and *k* is an instant (in time). As we see from Equation 6.11 the filter output will always be one sample delayed with respect to the unfiltered signal.

It should be noted that Equation 6.10 is based upon an approximation. Therefore the notch frequency might differ slightly from the intended one using the discrete-time filter in Equation 6.11. This phenomenon is called frequency warping [11], the effect may not give rise to any problems but we should keep it in mind that it certainly can. Below an explanation is given to why this effect appears and how to partially avoid it, the general discussion follows the discussion in B. Wittenmark et al. [11]. Consider the continuous-time transfer function *G.* The discrete-time transfer function (translated with Tustin's approximation) is given by

$$H(e^{i\omega h}) = \frac{1}{i\omega h}(1 - e^{-i\omega h})G\left(\frac{2(e^{i\omega h} - 1)}{h(e^{i\omega h} + 1)}\right) = G\left(\frac{2(e^{i\omega h/2} - e^{-i\omega h/2})}{h(e^{i\omega h/2} + e^{-i\omega h/2})}\right).$$

(Here *z* has been replaced by $e^{i\omega h}$ for the case with transmission of sinusoids. The first two factors arise due to sample and hold [11]). The argument of G can be rewritten as

$$\frac{2(e^{i\omega h/2} - e^{-i\omega h/2})}{h(e^{i\omega h/2} + e^{-i\omega h/2})} = \frac{2\mathrm{i}\sin\left(\frac{\omega h}{2}\right)}{h\cos\left(\frac{\omega h}{2}\right)} = \frac{2\mathrm{i}}{h}\tan\left(\frac{\omega h}{2}\right).$$

We would normally write the continuous-time transfer function as

$$G(\omega' i).$$

However, due to Tustin's approximation, the frequency ω' becomes distorted according to

$$\omega' = \frac{2}{h}\tan\left(\frac{\omega h}{2}\right) \Rightarrow \omega = \frac{2}{h}\arctan\left(\frac{\omega' h}{2}\right) \approx \omega'\left(1 - \frac{(\omega' h)^2}{12}\right).$$

Consequently, as long as the frequency is low and/or the sample time is short, the frequency distortion will be small. Figure 6.12 illustrates the effects of frequency warping with a notch filter sampled at two different sampling rates. As already mentioned, there are means to partially avoid the effects of frequency warping. This is called prewarping [11]. Prewarping has the effect that the distortion is canceled at a specific frequency, e.g. the resonance frequency of some mechanics in the case of a notch filter. The used approximation method, i.e., Tustin's approximation, is modified according to

$$s = \frac{\omega_n}{\tan(\omega_n h/2)} \cdot \frac{z-1}{z+1} \qquad (6.12).$$

With the modification expressed in Equation 6.12 there is no distortion at exactly ω_n. There will however be a distortion at other frequencies [11].

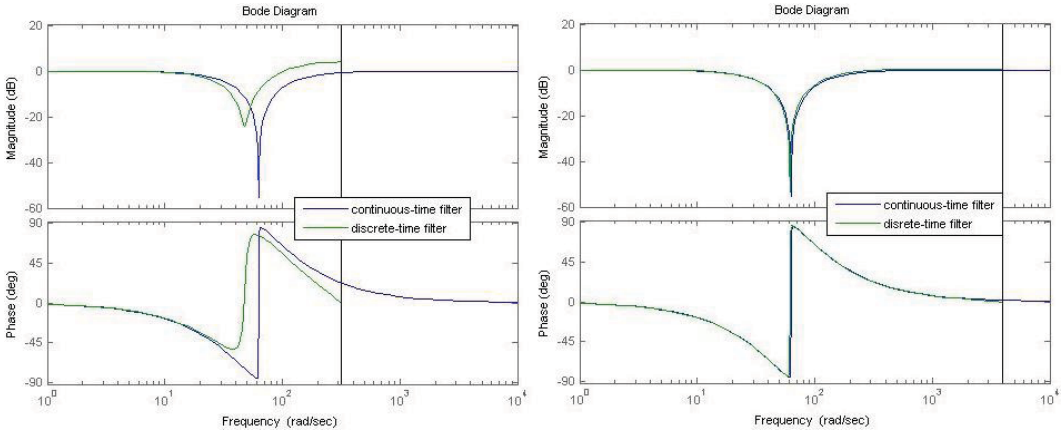

Figure 6.12. *Shows the same notch filter, $f_n = 10\,Hz$ and $q = 600$ at different sampling rates. To the left $h = 10\,ms$ and to the right $h = 0{,}8\,ms$. The figure shows that we must keep the effect of frequency warping in mind when choosing sampling rate. The frequency value has the same effect as the sampling rate for this undesired phenomenon, i.e., the distortion is bigger at higher frequencies.*

It is also relevant to discuss the case when there are several resonances in the

considered system. For this case it is natural and straight forward to use cascaded notch filters. However, the more filters that are introduced, the more cumbersome filter parameter calculations get. In the Laplace domain, cascaded notch filters are the same as

$$F_{multiple\,notch}(s)=F^{1}_{notch}(s)\cdot F^{2}_{notch}(s)\cdot\ldots\cdot F^{n}_{notch}(s).$$

For a double notch filter, this means

$$F_{double\,notch}(s)=\frac{s^2+2\frac{\omega_1}{q_1}s+\omega_1^2}{(s+\omega_1)^2}\cdot\frac{s^2+2\frac{\omega_2}{q_2}s+\omega_2^2}{(s+\omega_2)^2}=\frac{s^4+\beta_0 s^3+\beta_1 s^2+\beta_2 s+\beta_3}{s^4+\alpha_0 s^3+\alpha_1 s^2+\alpha_2 s+\alpha_3} \quad (6.13)$$

where

$$\beta_0=2\frac{\omega_1}{q_1}+2\frac{\omega_2}{q_2},\ \beta_1=\omega_1^2+\omega_2^2+\frac{4\omega_1\omega_2}{q_1 q_2},\ \beta_2=\frac{2\omega_1\omega_2^2}{q_1}+\frac{2\omega_1^2\omega_2}{q_2},$$
$$\beta_3=\alpha_3=\omega_1^2\omega_2^2,\ \alpha_0=2\omega_1+2\omega_2,\ \alpha_1=\omega_1^2+\omega_2^2+4\omega_1\omega_2,\ \alpha_2=2\omega_1\omega_2^2+2\omega_1^2\omega_2.$$

Using Tustin's approximation (with no prewarping) on Equation 6.13, the implementable discrete-time transfer function can be expressed as Equation 6.14.

$$H_{double\,notch}(z)=\frac{b_0 z^4+b_1 z^3+b_2 z^2+b_3 z+b_4}{a_0 z^4+a_1 z^3+a_2 z^2+a_3 z+a_4} \quad (6.14)$$

where

$$b_0=\beta_3 h^4+2\beta_2 h^3+4\beta_1 h^2+8\beta_0 h+16,$$
$$b_1=4\beta_3 h^4+4\beta_2 h^3-16\beta_0 h-64,$$
$$b_2=6\beta_3 h^4-8\beta_1 h^2+96,$$
$$b_3=4\beta_3 h^4-4\beta_2 h^3+16\beta_0 h-64,$$
$$b_4=\beta_3 h^4-2\beta_2 h^3+4\beta_1 h^2-8\beta_0 h+16,$$

$$a_0=\alpha_3 h^4+2\alpha_2 h^3+4\alpha_1 h^2+8\alpha_0 h+16,$$
$$a_1=4\alpha_3 h^4+4\alpha_2 h^3-16\alpha_0 h-64,$$
$$a_2=6\alpha_3 h^4-8\alpha_1 h^2+96,$$
$$a_3=4\alpha_3 h^4-4\alpha_2 h^3+16\alpha_0 h-64,$$
$$a_4=\alpha_3 h^4-2\alpha_2 h^3+4\alpha_1 h^2-8\alpha_0 h+16.$$

With the same notation (except for filter parameters) and reasoning as for Equation 6.11, the double notch filter can be implemented in a computer program according to Equation 6.15.

$$\xi_k^{filtered} = \frac{1}{a_0}\left(\sum_{n=0}^{4} b_n \xi_{k-n} - \sum_{n=1}^{4} a_n \xi_{k-n}^{filtered}\right) \qquad (6.15).$$

Again, we note that the filtered signal is delayed one sample with respect to the unfiltered signal. Implementing a double notch filter (or more) would also be possible by creating a single notch filter for each frequency and then using the output of the first filter as input to the next filter. The calculations would then have to be done within the same task cycle, otherwise a delay of one sample is added for each frequency. If a prewarped solution is desired, it should be straight forward to do so. In that case, several modified Tustin's approximations are needed (one for each frequency). By means of notch filtering the reference signal, a potentially good result is obtained. It is only the natural frequency of the vibrations that has to be known, the damping factor q can be chosen empirically.

7. Implementation

This chapter deals with implementation aspects concerning the control strategies described in Chapter 6. Both the model based feedback solution and the notch filters were implemented in the PLC. Since the servo control is done by the ACOPOS this means that there are some issues that have to be considered. One issue could for instance be time delays, however no problems have been encountered in connection with these and therefore they will not be discussed. As have already been mentioned in previous chapters, the task class cycle time for both feedback control and notch filter output generation was set to 0,8*ms*. The reason for doing this has been to match the Powerlink communication time between the PLC and the ACOPOS. In Chapter 4 it is mentioned that the position control loop runs at a cycle time of 0,4*ms*. Interpolation similar to the one between the position loop and the speed loop would be a good idea to do. This has however not been done and therefore it remains as a task for the future. No visual effects arise due to the different cycle times. It is however possible to hear the quantization in the servo motors. Method specific implementation aspects are treated in the sections below.

7.1 Model based feedback

As we have seen in Chapter 6, the distortion estimate from the model is subtracted from the position reference. Looking at the block diagram of the position controller in Figure 4.9 we see that there is no additive parameter before the amplification factor K_v as there is in the speed controller (v_{addref}) and in the current controller (i_{addref}). Having an additive parameter it is possible to let the ACOPOS read from a variable in the PLC cyclically and set this parameter accordingly. Because this is not the case for the position controller another solution has been created to get the same effect.

It is possible to create a cam-coupling between the real servo and a virtual servo. This essentially means that the virtual servo acts as master and the real servo acts as slave. Then it is possible to adjust the offset between the two servos; by letting the offset act as feedback signal the real servo will follow the virtual servo except for the additive parameter. The situation can be illustrated as in Figure 7.1.

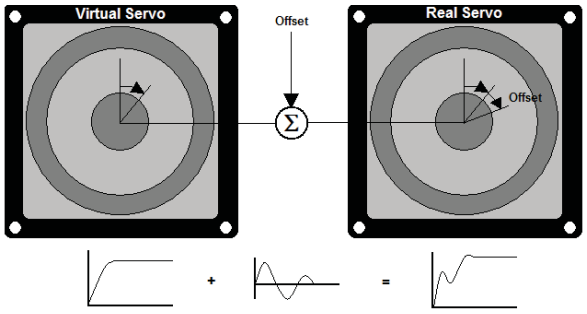

Figure 7.1. *Cam-coupling between a master servo and a slave servo. The real servo follows the reference of the virtual servo except for an offset.*

The cam-coupling is achieved with an already made Structured Text (ST) program, the only thing that needs to be done is to link the offset with the model feedback. The model, the current reading program and the program that sets the offset parameter have been implemented in C. Figure 7.2 illustrates the program hierarchy.

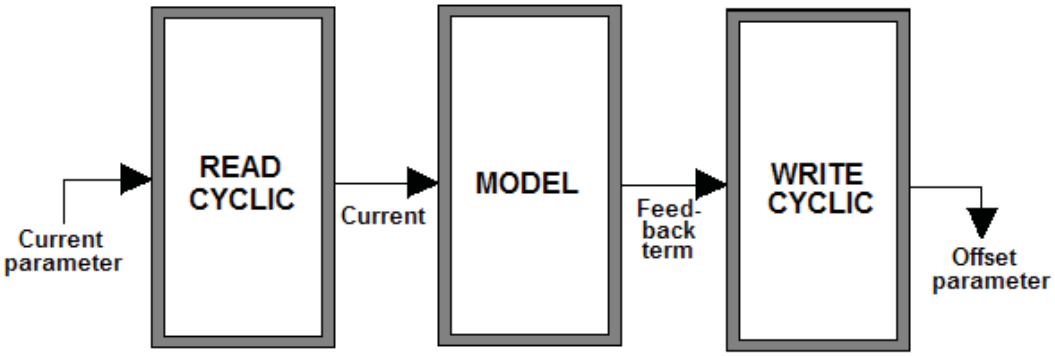

Figure 7.2. *Feedback control program hierarchy.*

Read Cyclic, Model and Write Cyclic in Figure 7.2 are set as parallel tasks with the same task class cycle time in the PLC.

7.2 Notch filters

The main issue regarding implementation of command shaping notch filters was the question of how to get in between the Path Generator and the position loop, i.e., without reprogramming the ACOPOS firmware. It is only the Path Generator that owns writing access to the reference position parameter, however, it is of course possible to read from the parameter. If it would have been possible to decouple the Path Generator from the position loop and if we were granted writing access to the reference position parameter then a solution could be to read the Path Generator parameter, pass the variable through our notch filter and then write the new variable to the position reference parameter. Since this has not been an option a solution with a virtual servo similar to the solution in Section 7.1 was created. Instead of letting the real servo follow the virtual servo except for an offset as before, the reference following was deactivated and only the additive offset remained active.

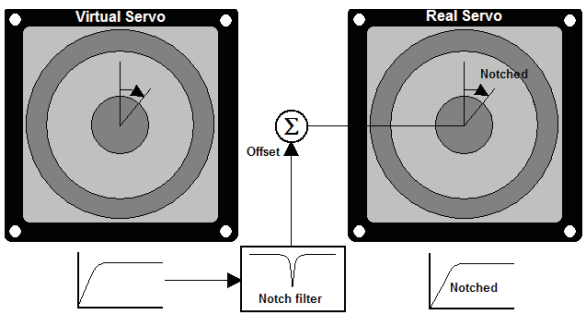

Figure 7.3. *The set position of the virtual servo is passed through a notch filter, the filtered signal is then passed to the real servo via the offset.*

The set position parameter value of the virtual servo was stored in a variable, this variable was then passed through the notch filter and the output was written to the offset. In this way the position reference reaching the real servo has been a notched position reference profile from the virtual servo. Figure 7.3 illustrates the scenario. The notch filter, the virtual position reading program and the program that sets the offset parameter has been implemented in C. Figure 7.4 illustrates the program hierarchy.

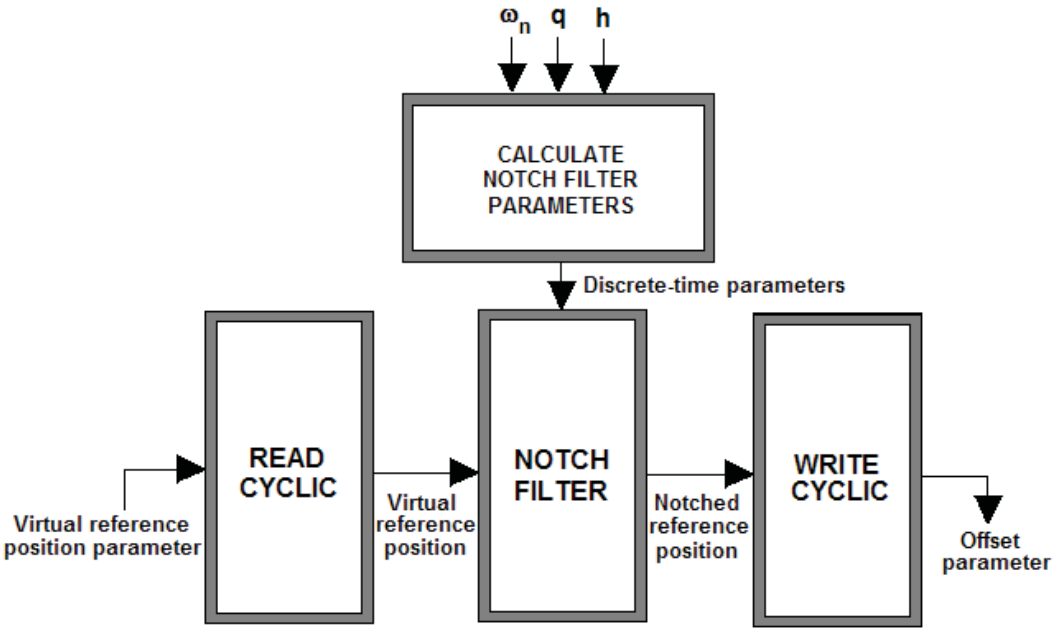

Figure 7.4. *Notch filter program hierarchy.*

Read Cyclic, Notch Filter and Write Cyclic in Figure 7.4 are set as parallel tasks with the same task class cycle time in the PLC (0,8*ms*). Calculate Notch Filter Parameters runs at a lower task class cycle time (100*ms*) as the real-time demand is not high for this task.

8. Results and discussion

In this chapter results based on accelerometer, stator current and lag error data for command shaping notch filters are presented. Both the experimental setup and the portal robot (described in Chapter 2) have been used to produce the results. Furthermore, a comparison to the jolt limitation method is conducted on the portal robot in order to find out which method to prefer. A discussion is given in connection with the results and also in the last section. Both single and double notch filters (see Chapter 6) have been tested. No prewarping was implemented on the filters since the sample time is very short and the considered natural frequencies are in the range of 10*Hz* (see Figure 6.12 for further motivation). Moreover, the proportional amplification K_v in the position controller has been altered to see the effects on both lag error and vibrations.

8.1 Results from the experimental setup

Experiments have been conducted by moving the X-axis along the Z-axis rapidly and then giving it an abrupt stop so that (mainly the X-axis beam) vibrations could be studied. The weight position was set so that the natural frequency of the vibrations should have a value of about 9*Hz* as in previous chapters. The final value of the position reference has been altered but the speed and acceleration set points have remained constant according to

$$\begin{aligned} v\_pos &= 5 \cdot 10^6 \, units/s \\ a1\_pos &= 7 \cdot 10^6 \, units/s^2 \\ a2\_pos &= 7 \cdot 10^6 \, units/s^2 \end{aligned}$$

or in Z-axis coordinates

$$\begin{aligned} \dot{z}_{ref} &= 5 \cdot 10^6 c_z = 9{,}45 \, m/s \\ \ddot{z}_{ref} &= \pm 7 \cdot 10^6 c_z = \pm 13{,}23 \, m/s^2 . \end{aligned}$$

Figures 8.1 and 8.2 show the results for a final value of the position reference at 36000 *units* or 0,068*m* unfiltered and filtered. The proportional amplification in the position controller was set to $K_v = 440 \; 1/s$. Figures 8.3 and 8.4 show the same results for the same position reference but with the double proportional amplification in the position controller, i.e., $K_v = 880 \; 1/s$. The filter that was used has been calculated with $f_n = 9 \, Hz$ or $\omega_n = 56{,}5 \, rad/s$ and $q = 600$. The accelerometer was attached to the weight as described earlier.

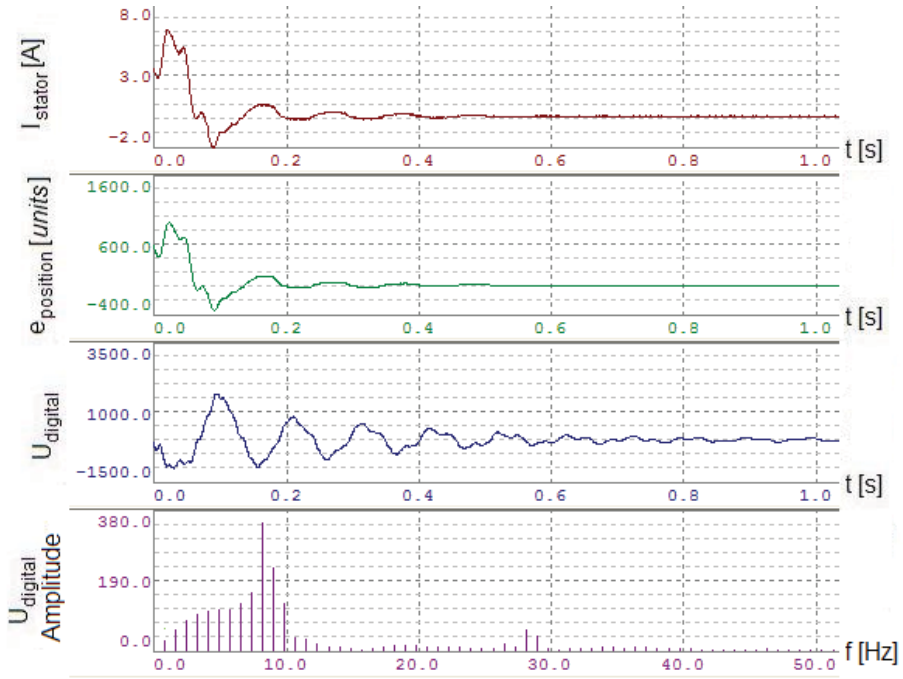

Figure 8.1. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

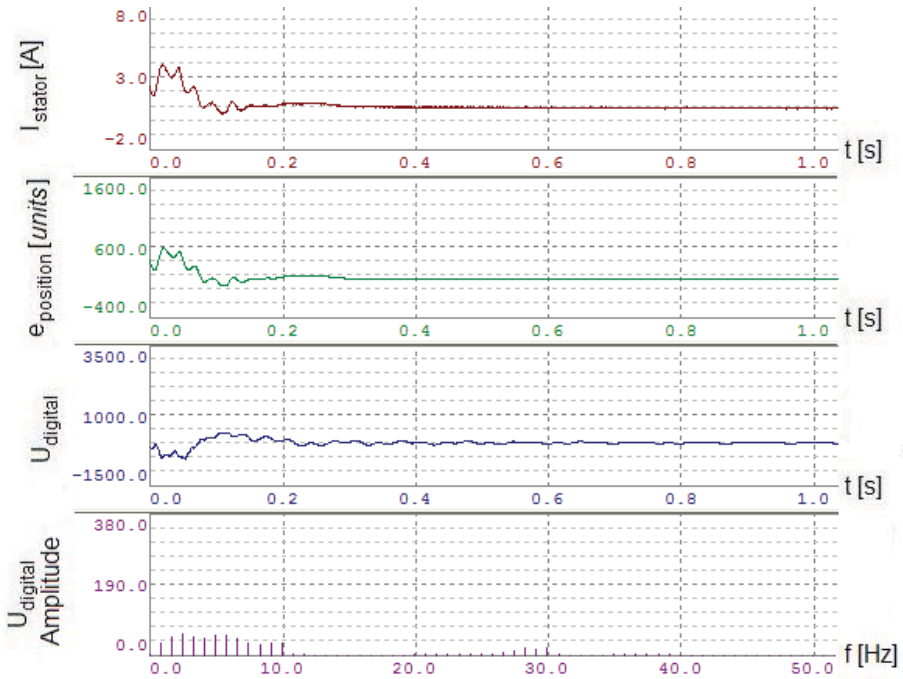

Figure 8.2. *Filtered results, single notch 9Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

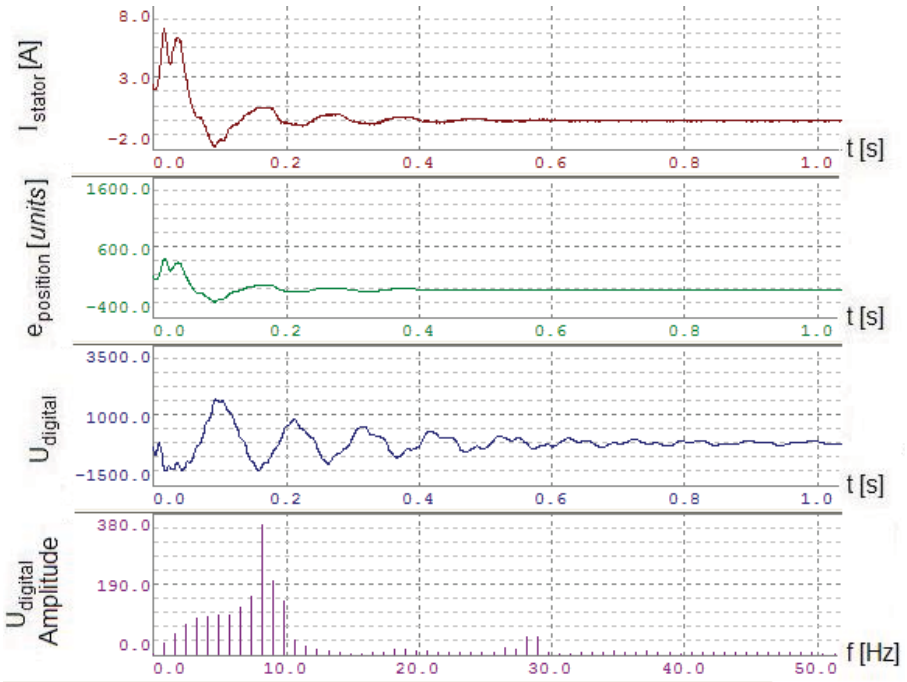

Figure 8.3. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

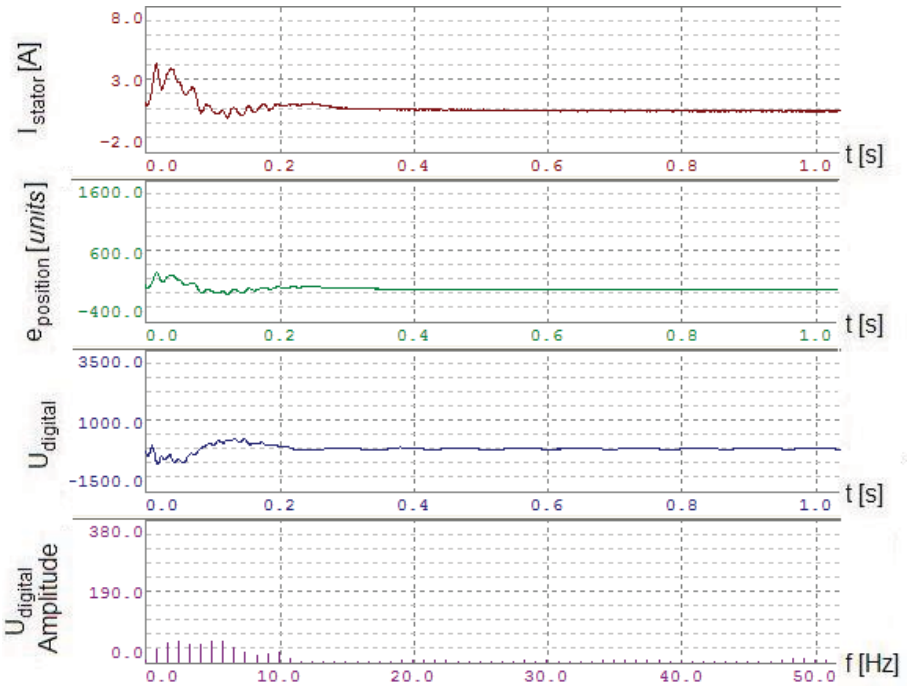

Figure 8.4. *Filtered results, single notch 9Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

Figures 8.5 and 8.6 show the results for a final value of the position reference at 72000 *units* or 0,1361*m* unfiltered and filtered. The proportional amplification in the position controller was set to $K_v = 440\ 1/s$. Figures 8.7 and 8.8 show the

same results for the same position reference but with the double proportional amplification in the position controller, i.e., $K_v = 880\ 1/s$.

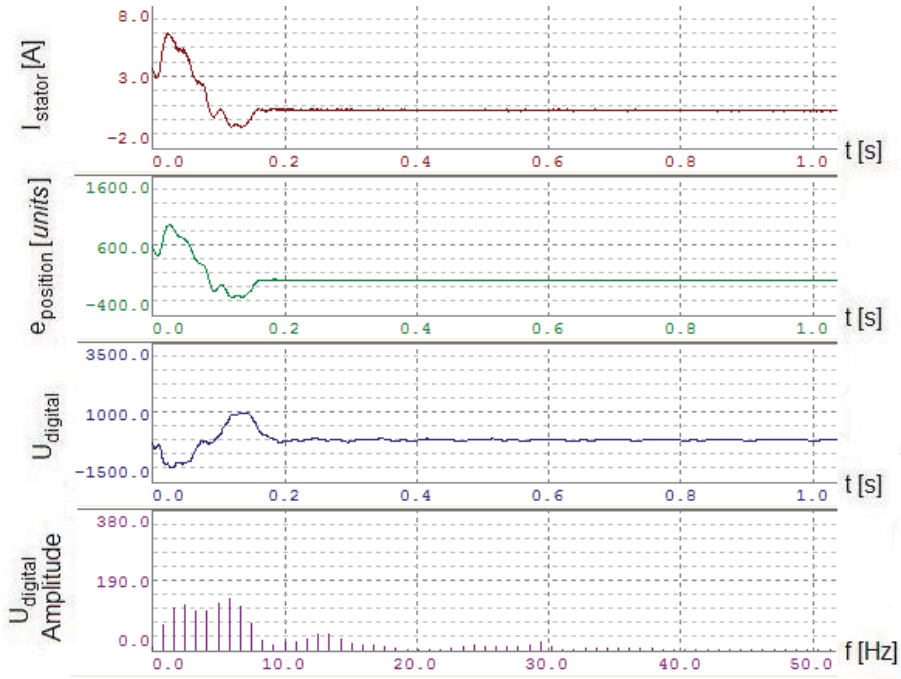

Figure 8.5. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

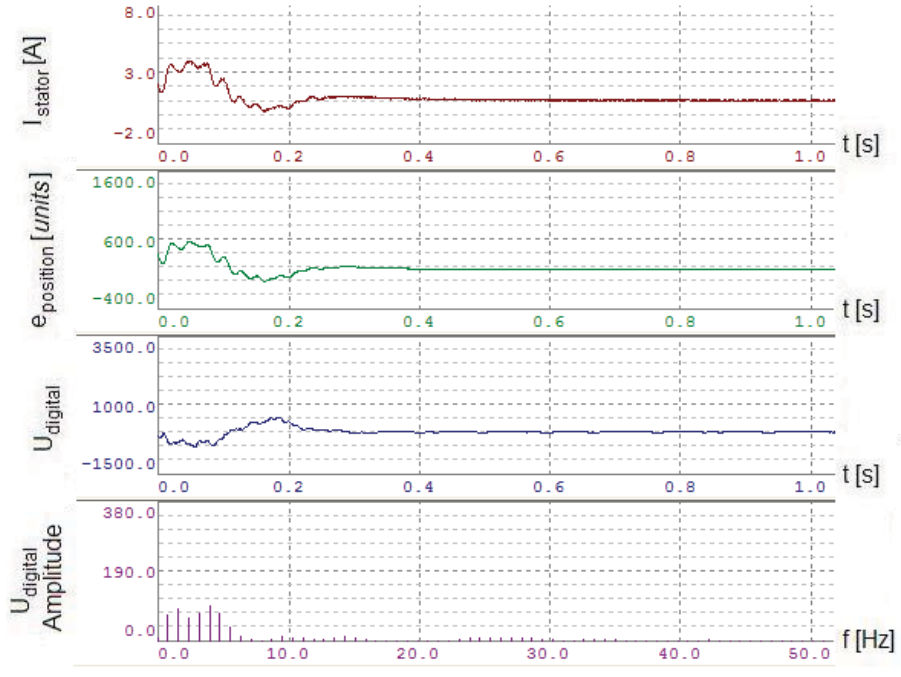

Figure 8.6. *Filtered results, single notch 9Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

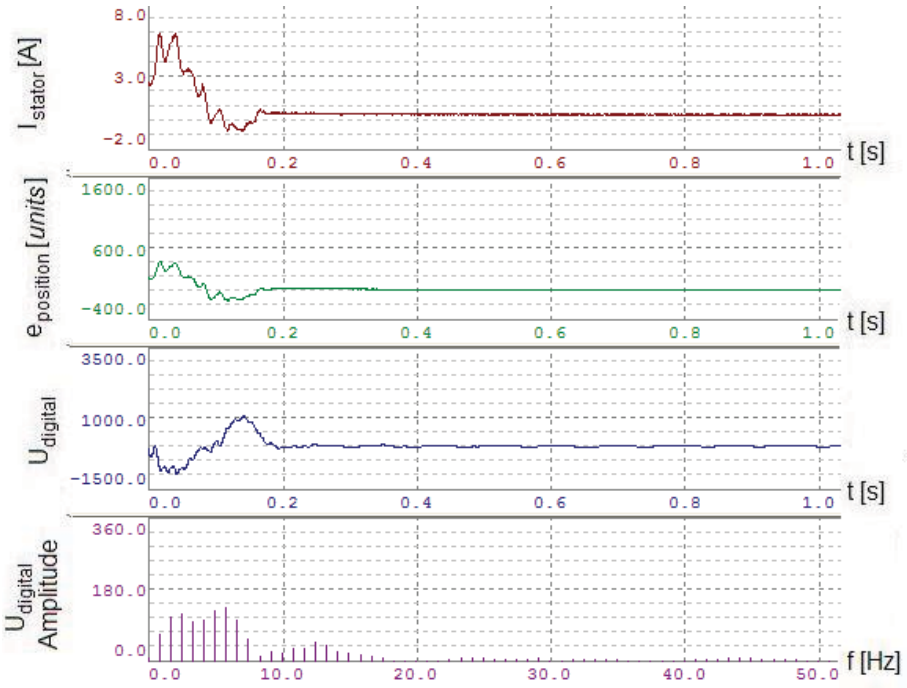

Figure 8.7. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

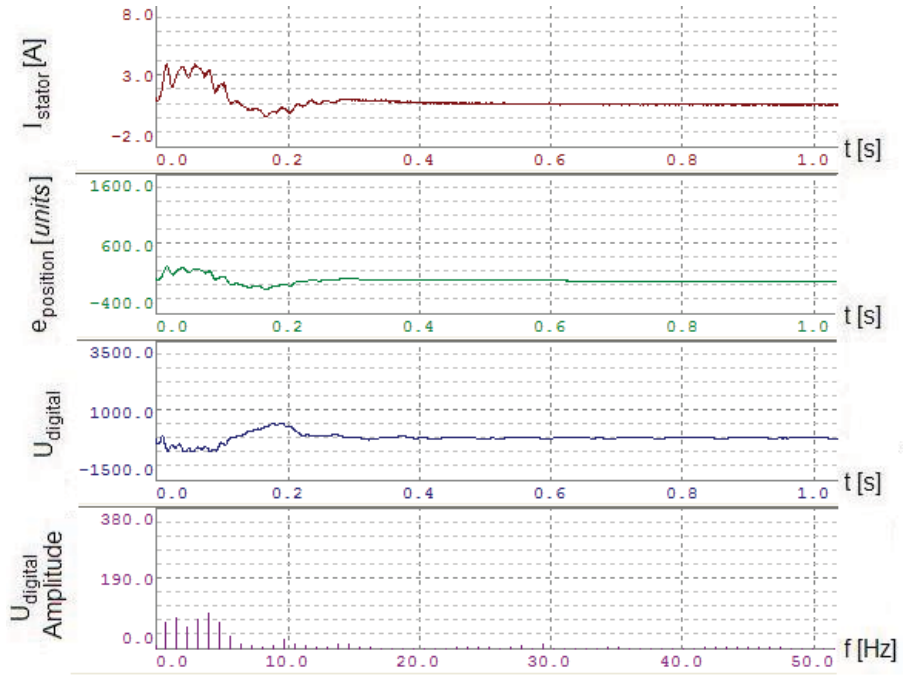

Figure 8.8. *Filtered results, single notch 9Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

Figures 8.9 and 8.10 show the results for a final value of the position reference at 144000 *units* or 0,2722*m* unfiltered and filtered. The proportional amplification in the position controller was set to $K_v = 440\ 1/s$. Figures 8.11 and 8.12 show the

same results for the same position reference but with the double proportional amplification in the position controller, i.e., $K_v = 880\ 1/s$.

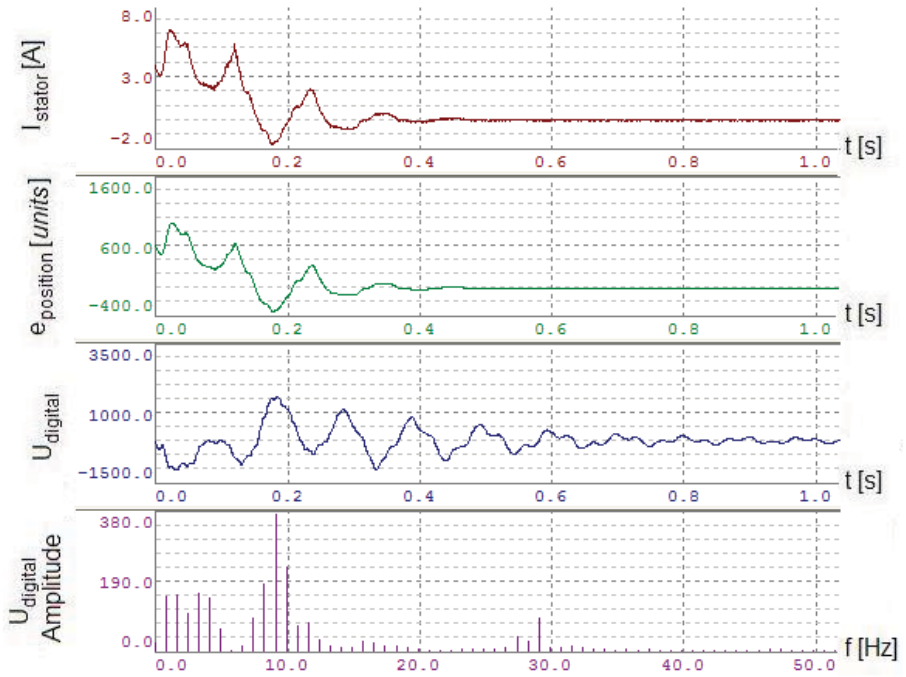

Figure 8.9. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\ units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

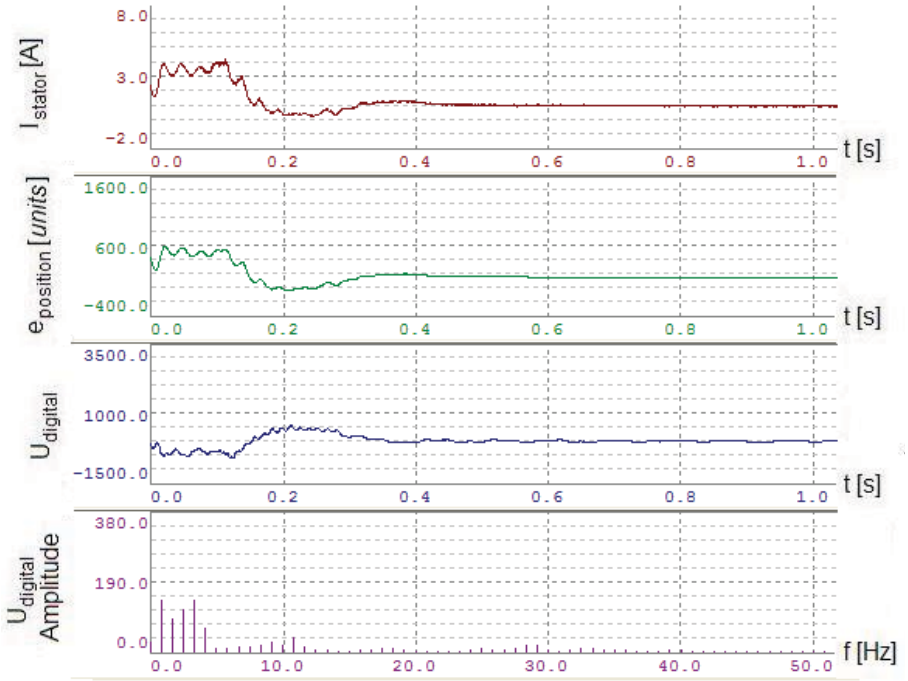

Figure 8.10. *Filtered results, single notch 9Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\ units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

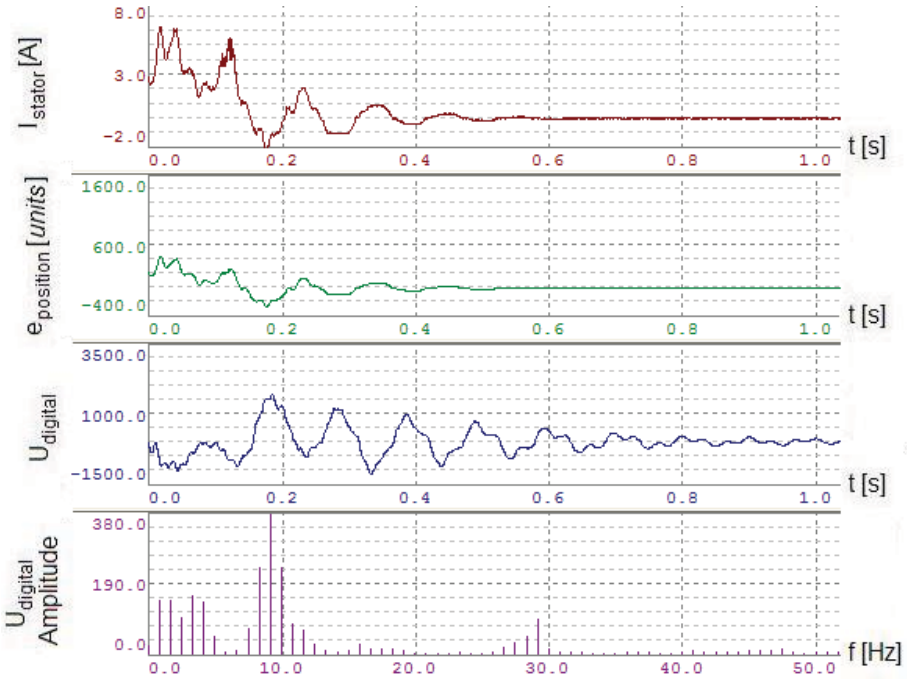

Figure 8.11. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 880\,1/s$.

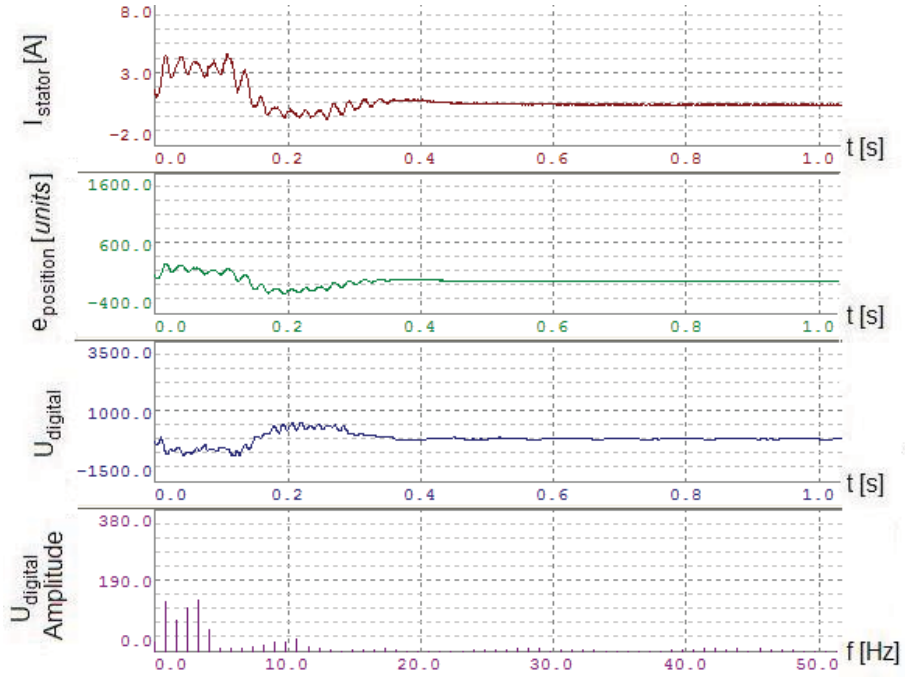

Figure 8.12. *Filtered results, single notch 9Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 880\,1/s$.

If we compare Figure 8.1 to Figure 8.5 where the only difference is *s*, we see that it is not all reference profiles that induce much vibrations. However, the notch filter is still effective at (and around) the notch frequency and it reduces the

bending of the X-axis a bit as we see in Figure 8.6. The same discussion applies to Figure 8.3 and Figure 8.7 where the proportional amplification in the position controller is doubled. As we see in the results above, e.g. in Figures 8.9 and 8.10, there is a higher frequency component in the signals. Some experiments with a double notch filter were conducted in order to see if it was possible to attenuate these vibrations in addition to the other. Figure 8.13 shows the results for a filter with $q_1 = q_2 = 600$, $f_1 = 14{,}15\,Hz$ and $f_2 = 30\,Hz$. The position controller proportional amplification was set to $K_v = 440\ 1/s$ and the final position was set to $s = 144000\,units;$ in other words the results in Figure 8.13 should be compared to the results in Figure 8.10 where the same controller parameters were used.

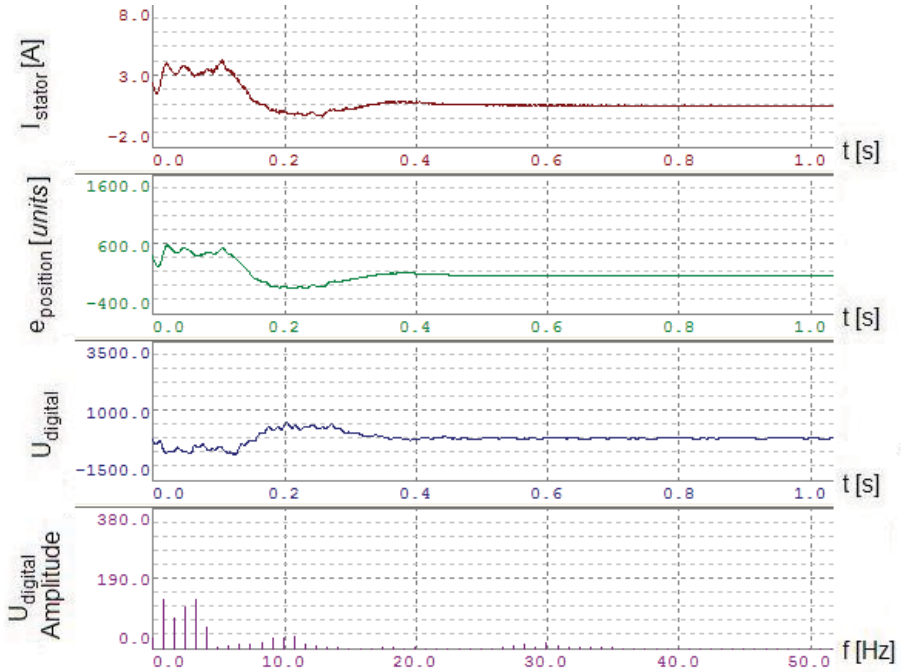

Figure 8.13. *Filtered results, double notch 9Hz and 30Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

Comparing the results in Figures 8.10 and 8.13, we see that there are less vibrations in the signals that were generated with the double notch filter. However, the difference is not very large and the higher frequency component may be controller induced.

8.2 Results from the portal robot

Experiments have been conducted analogously to the experiments in the previous section, i.e., by moving the X-axis along the Z-axis rapidly and then giving it an abrupt stop so that (mainly the Y-axis beam) vibrations could be studied. The tool point position was set so that the natural frequency of the vibrations should have a value of about 14,15*Hz* as in previous chapters. However, there are some frequency components in the signals slightly higher than 14,15*Hz*. The final value of the position reference has been altered but the speed and acceleration set points

have remained constant according to

$$v\_pos = 5 \cdot 10^6 \, units/s$$
$$a1\_pos = 7 \cdot 10^6 \, units/s^2$$
$$a2\_pos = 7 \cdot 10^6 \, units/s^2$$

or in Z-axis coordinates

$$\dot{z}_{ref} = 5 \cdot 10^6 \, k_z = 8{,}9 \, m/s$$
$$\ddot{z}_{ref} = \pm 7 \cdot 10^6 \, k_z = \pm 12{,}46 \, m/s^2.$$

Figures 8.14 and 8.15 show the results for a final value of the position reference at 36000 *units* or 0,064*m* unfiltered and filtered. The proportional amplification in the position controller was set to $K_v = 440 \; 1/s$. Figures 8.16 and 8.17 show the same results for the same position reference but with the double proportional amplification in the position controller, i.e., $K_v = 880 \; 1/s$. The filter that was used has been a double notch filter calculated with $f_1 = 14{,}15 \, Hz$ or $\omega_1 = 94{,}2 \, rad/s$, $f_2 = 16{,}15 \, Hz$ or $\omega_2 = 101{,}5 \, rad/s$ and $q_1 = q_2 = 1600$. The accelerometer was attached to the tool point as described earlier.

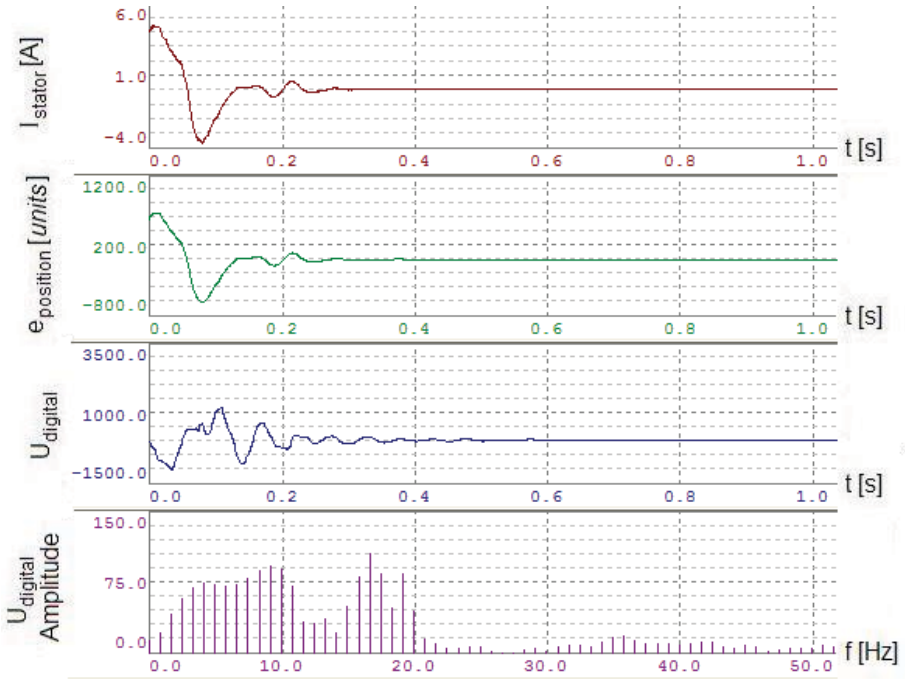

Figure 8.14. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000 \, units$ *and position controller proportional amplification* $K_v = 440 \; 1/s$.

Figures 8.18 and 8.19 show the results for a final value of the position reference at

72000 *units* or 0,128*m* unfiltered and filtered. The proportional amplification in the position controller was set to $K_v = 440\ 1/s$. Figures 8.20 and 8.21 show the same results for the same position reference but with the double proportional amplification in the position controller, i.e., $K_v = 880\ 1/s$.

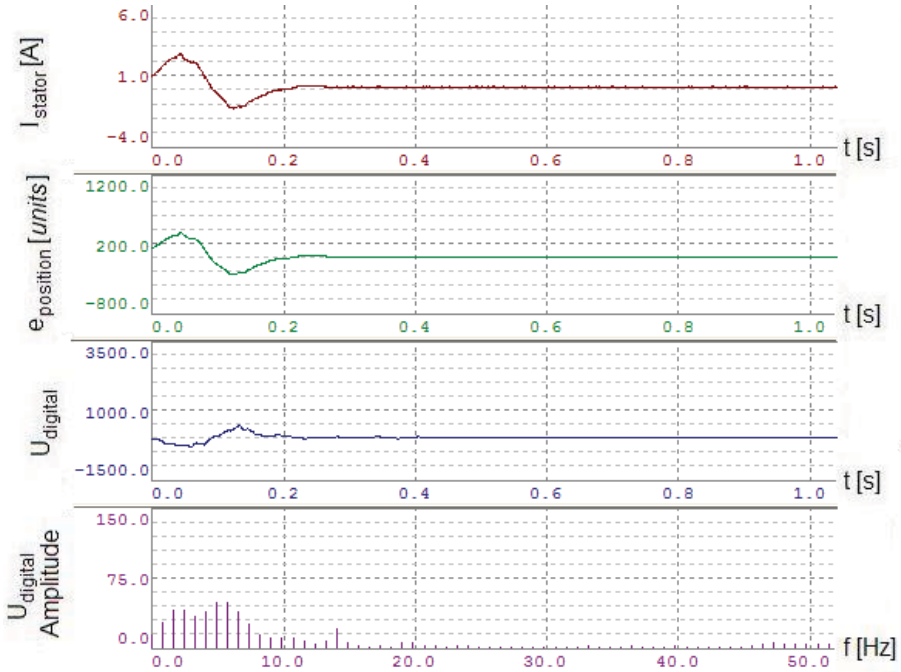

Figure 8.15. *Filtered results, double notch 14,15Hz and 16,15Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000\ units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

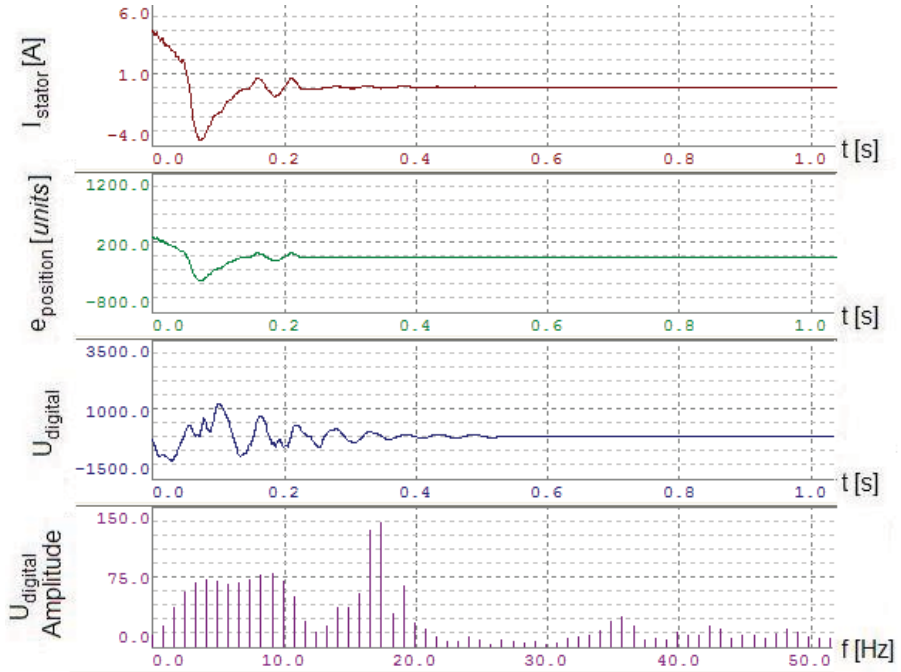

Figure 8.16. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000\ units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

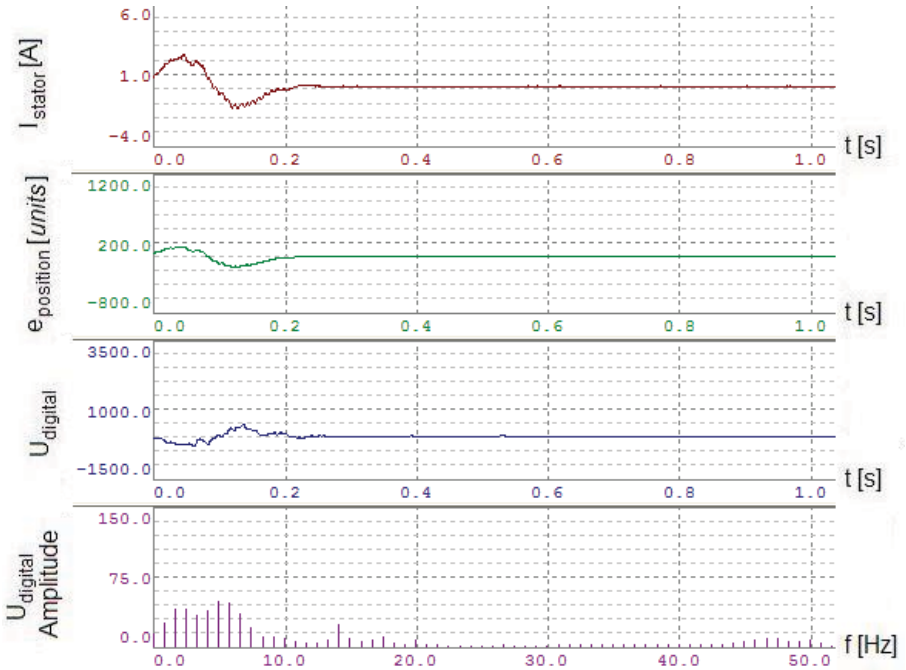

Figure 8.17. *Filtered results, double notch 14,15Hz and 16,15Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 36000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

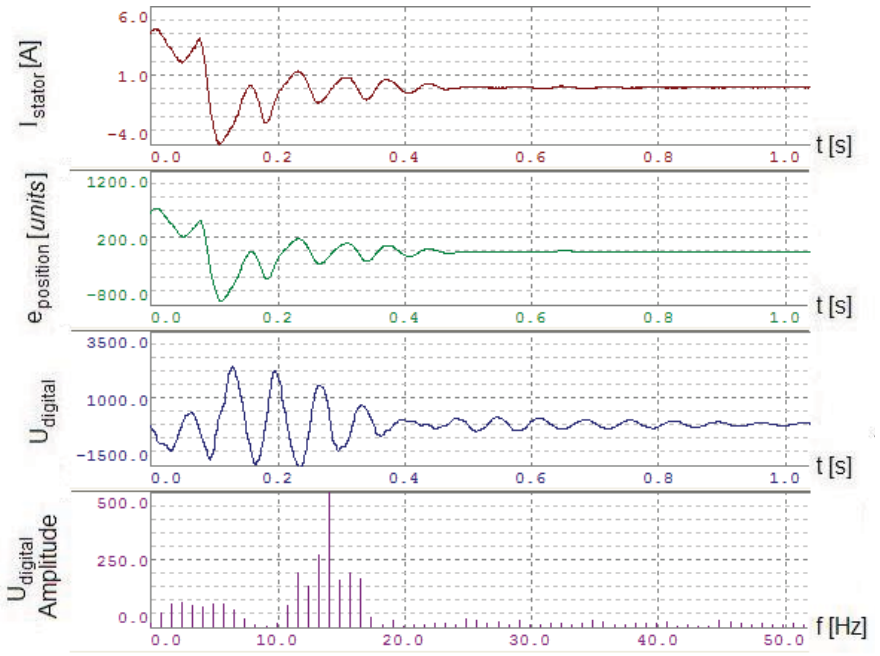

Figure 8.18. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

As we see in Figures 8.18 and 8.20, the profile used to generate the results would be very beneficial for identification of the natural frequency by just observing encoder or current data. The controller parameters for the data in Figure 8.18 (larger lag error) are better suited for this purpose though.

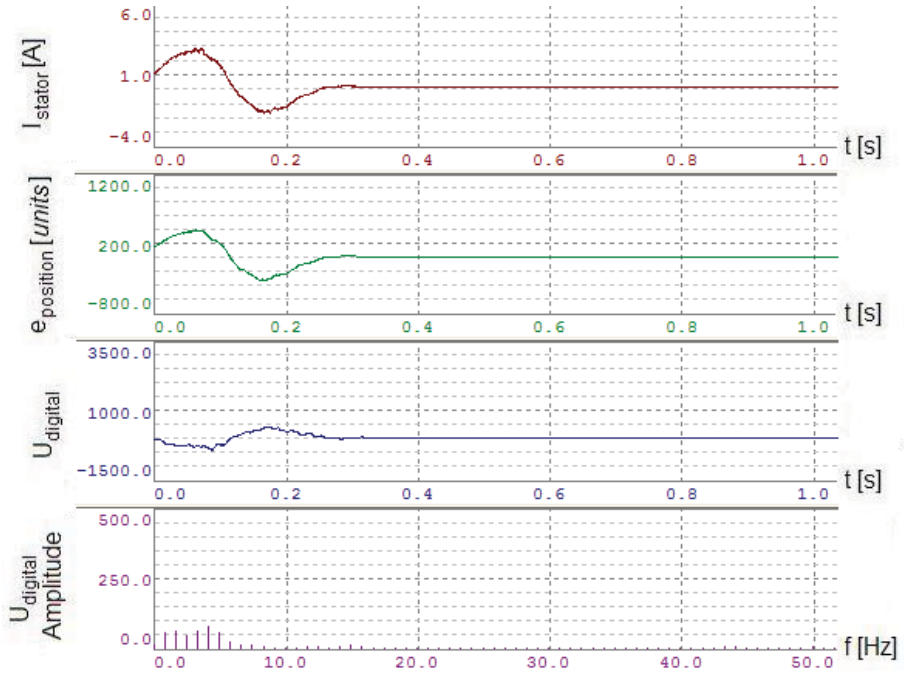

Figure 8.19. *Filtered results, double notch 14,15Hz and 16,15Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

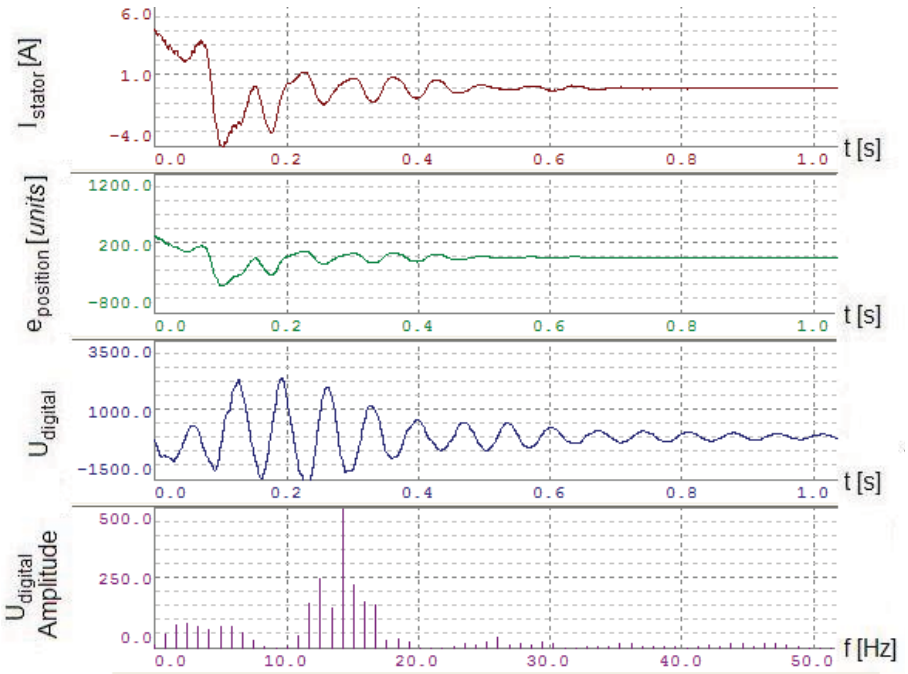

Figure 8.20. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

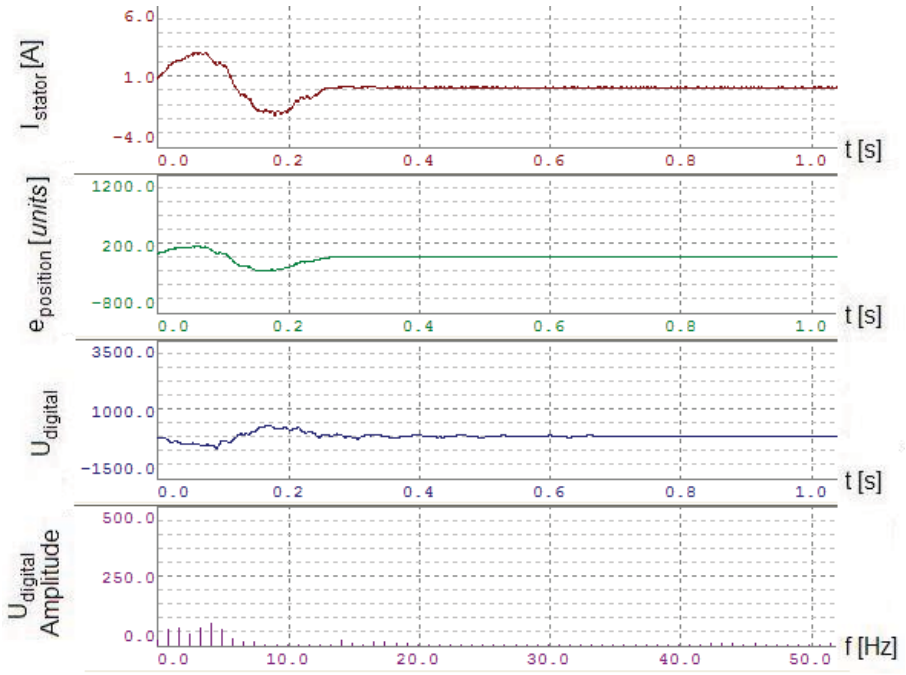

Figure 8.21. *Filtered results, double notch 14,15Hz and 16,15Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 72000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

Figures 8.22 and 8.23 show the results for a final value of the position reference at 144000 *units* or 0,256*m* unfiltered and filtered. The proportional amplification in the position controller was set to $K_v = 440\ 1/s$. Figures 8.24 and 8.25 show the same results for the same position reference but with the double proportional amplification in the position controller, i.e., $K_v = 880\ 1/s$.

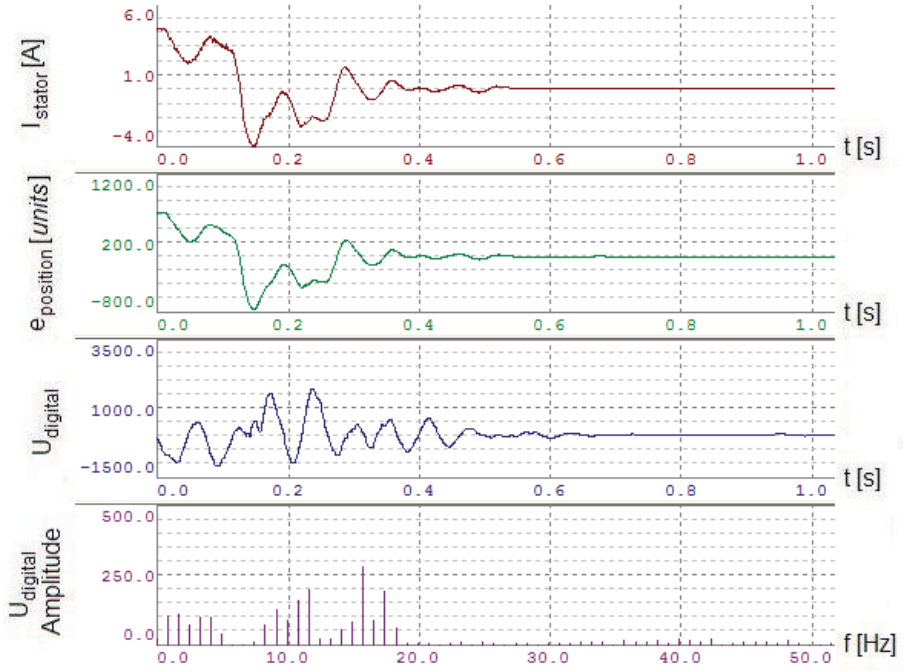

Figure 8.22. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

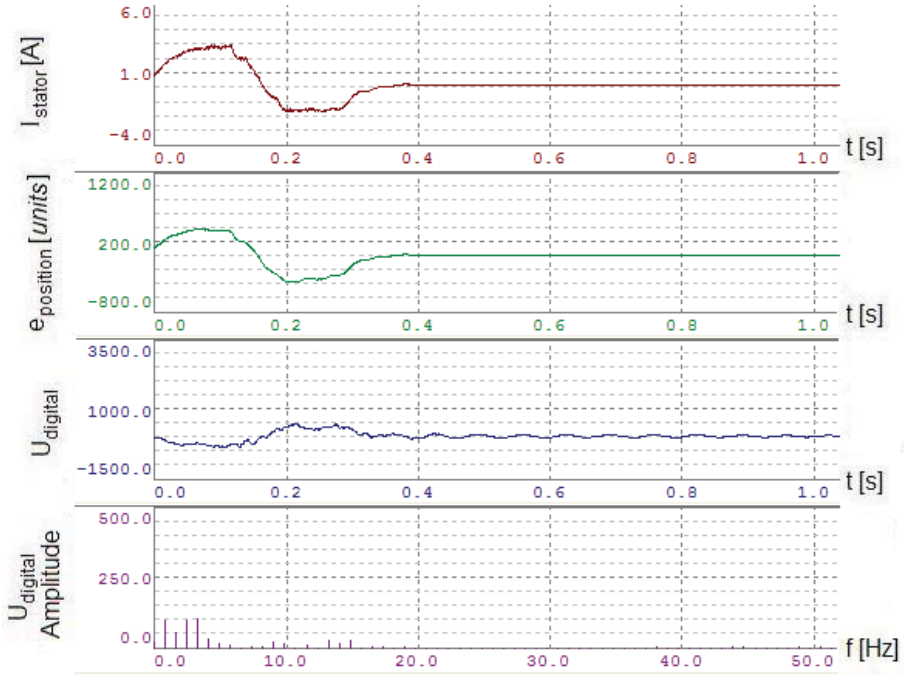

Figure 8.23. *Filtered results, double notch 14,15Hz and 16,15Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

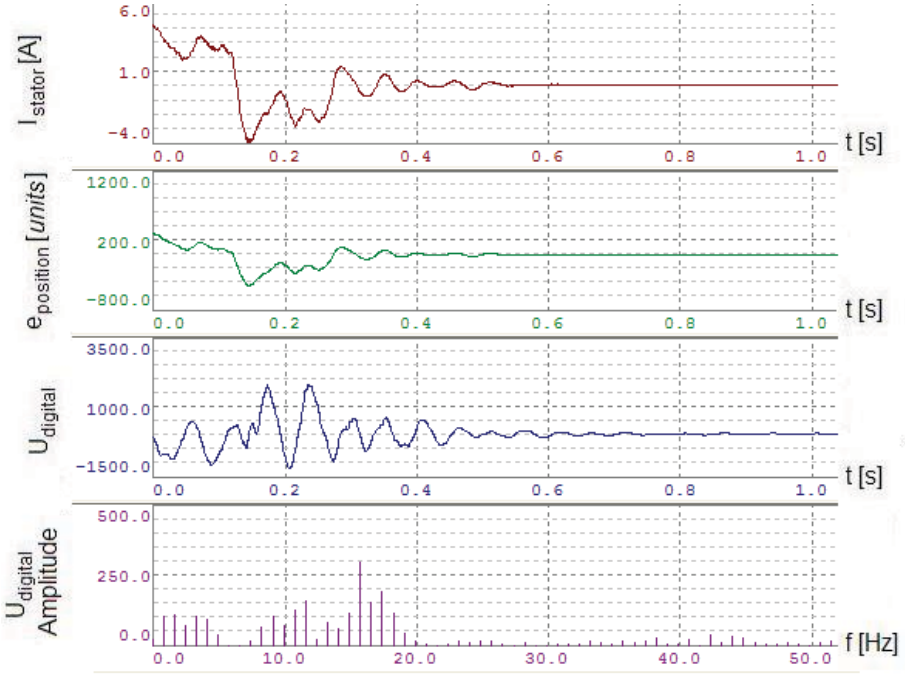

Figure 8.24. *Unfiltered results; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

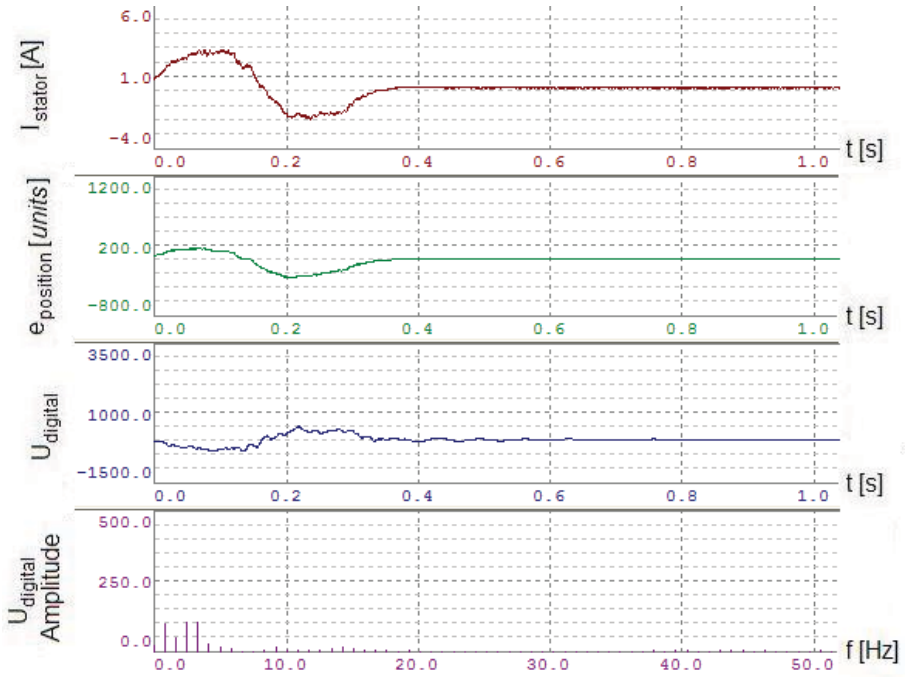

Figure 8.25. *Filtered results, double notch 14,15Hz and 16,15Hz; stator current, lag error, accelerometer data and FFT of the latter.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 880\ 1/s$.

As we see when data for filtered profiles are compared to data for unfiltered profiles, the results look quite satisfying. The question now is how command shaping notch filters perform compared to the jolt limitation method.

8.3 Notch filter and jolt limitation comparison

The jolt limitation method has proven to work quite well in many applications. It is a fact that it decreases servo induced vibrations in the rest of the process in comparison to untreated profiles. This has also been observed on the portal robot for which the method has been applied.

Results for a profile with $s = 144000\,units$ but with higher speed set points than the ones in Section 8.2 are presented in Figures 8.26 and 8.27. Figure 8.26 shows stator current, lag error and accelerometer data when jolt limitation was used while Figure 8.27 shows the same data but when the notch filter described in Section 8.2 was used. The parameter t_jolt was set according to the rule of thumb described in Chapter 6, Subsection 6.3.1, which means that t_jolt = 1/14,15 *Hz* = 0,0707 *s*. The results show that there are in fact less transient vibrations in the process when notch filters are used. If we look at the frequency content of the accelerometer signals by means of the Fast Fourier Transform this becomes evident; Figure 8.28 shows a comparison of the two FFTs.

The FFTs tell us more than this. If we consider frequencies lower than about 6,5*Hz* we see that there is less energy in the signal for which the notch filter has been used. The low frequency content is not due to vibrations, it is due to the movement of the whole X-axis. Consequently, the profile which the notch filter has reshaped is slower than the profile for which jolt limitation has been used. This means that the question becomes which profile is the fastest when both this fact and the transient of the vibrations are taken into account. Figure 8.29 shows a

comparison of both the reference profiles and accelerometer data where the time differences are especially considered.

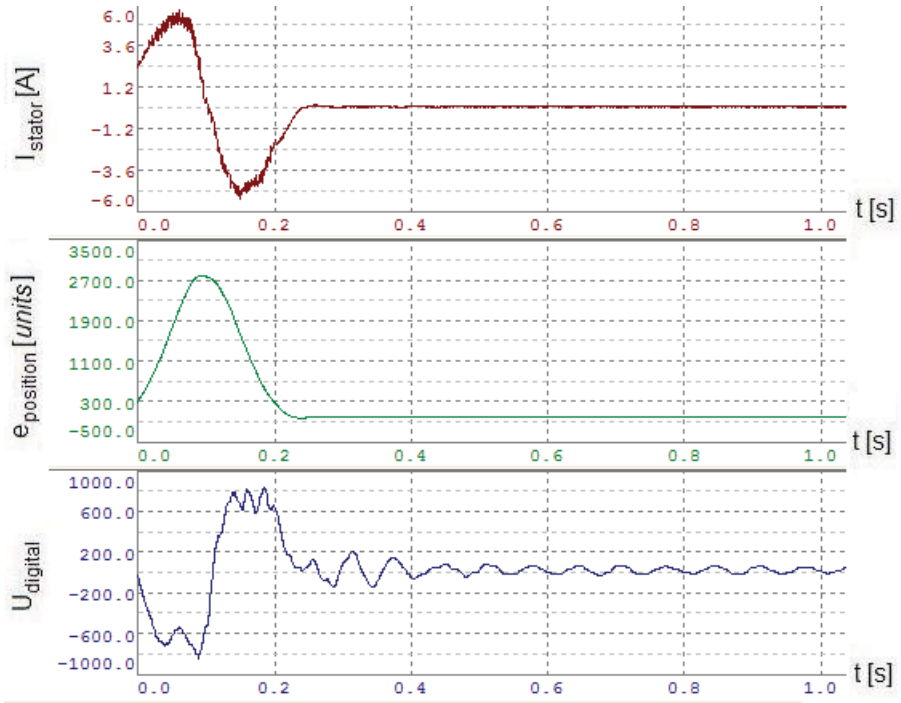

Figure 8.26. *Results when **jolt limitation** is used; stator current, lag error and accelerometer data.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

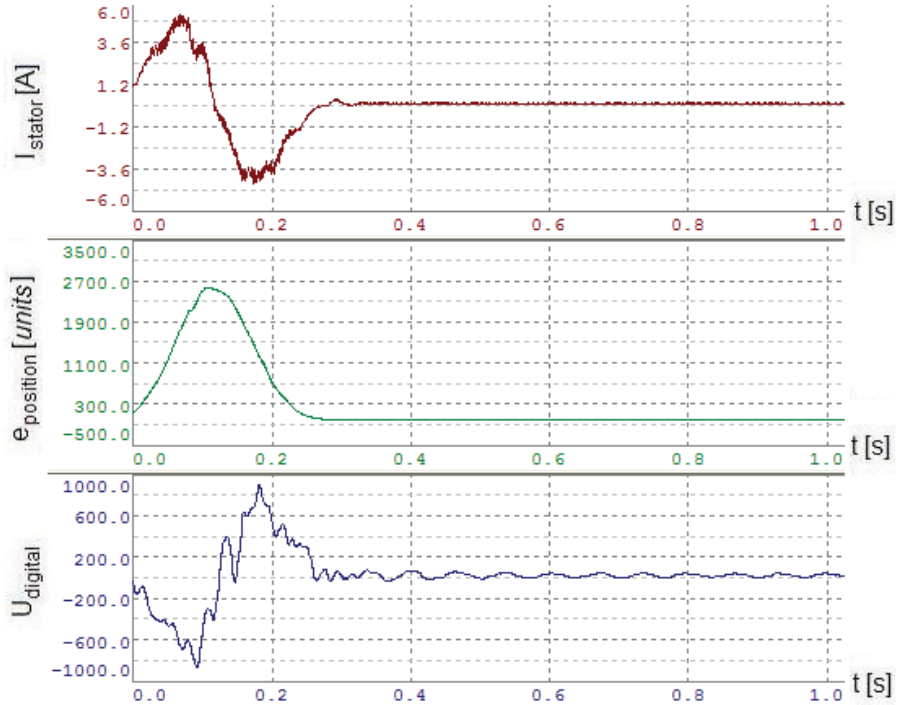

Figure 8.27. *Results when the **notch filter** is used; stator current, lag error and accelerometer data.* $s = 144000\,units$ *and position controller proportional amplification* $K_v = 440\ 1/s$.

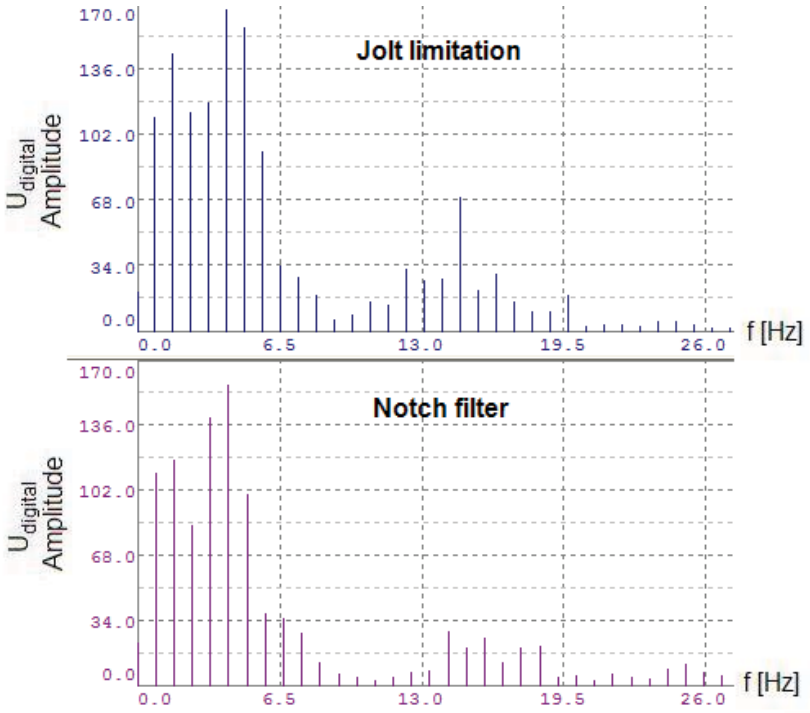

Figure 8.28. *Fast Fourier Transforms of the accelerometer signals in Figures 8.26 and 8.27. We see that there are less vibrations in the process when command shaping notch filters are used.*

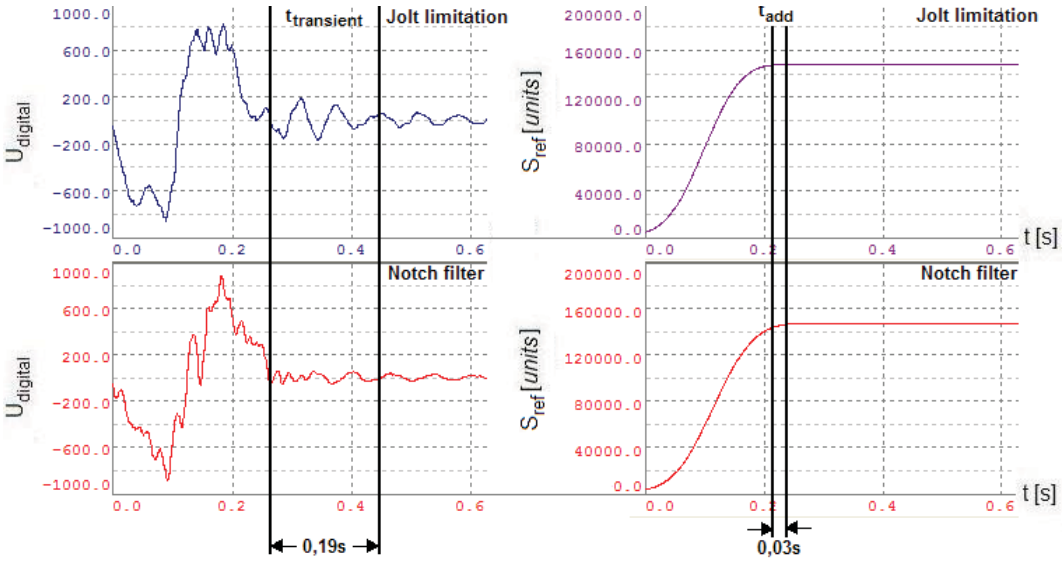

Figure 8.29. *A comparison of accelerometer data when jolt limitation is used and when notch filtering is conducted (left). Furthermore a comparison between the two different profiles is shown (right). The time it takes for the critical transient to fade is indicated as* $t_{transient}$ *and the additional time it takes for the notched profile to reach the final value is approximately indicated as* t_{add}.

As we see in Figure 8.29 there are no critical vibrations when the notch filter is used whereas it takes $t_{transient} = 0{,}19\,s$ for critical vibrations to fade when the jolt limitation method is used. We also see that the notched profile only is

approximately $t_{add}=0{,}03\,s$ slower than the jolt limited profile. Consequently, this means that the gain of using notch filters for this profile expressed in time becomes

$$t_{gain}=t_{transient}-t_{add}=0{,}19-0{,}03\,s=0{,}16\,s.$$

Other profiles have also been investigated. There has always been a gain of using notch filters instead of jolt limitation. The time gain can even be larger than the one calculated above. Without making the same detailed investigation as for the profile discussed above, accelerometer results for a profile with $s=72000\,units$ are shown in Figure 8.30.

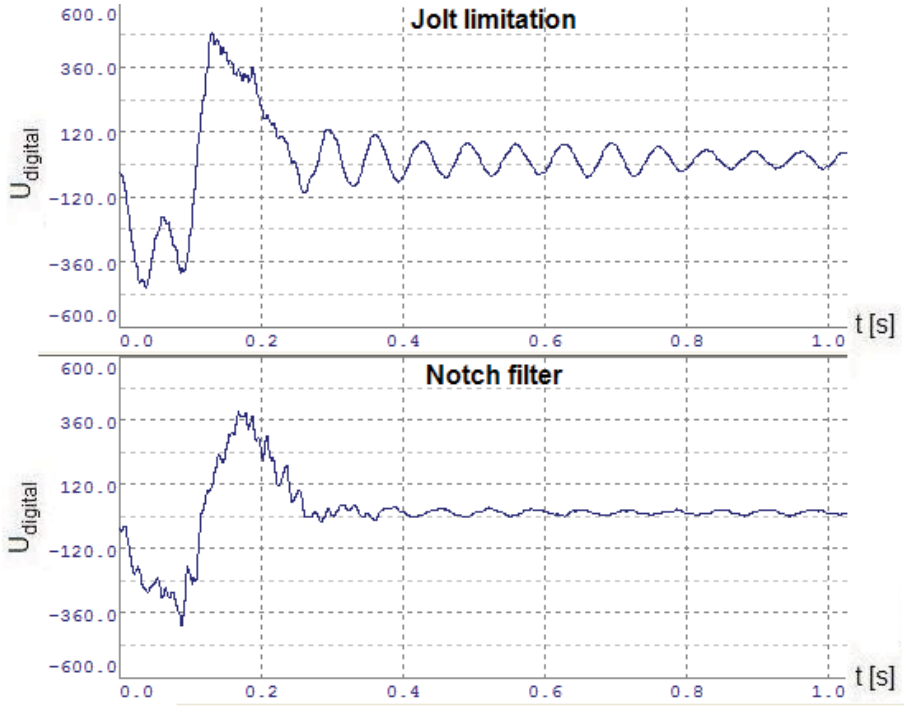

Figure 8.30. *Accelerometer results when **jolt limitation** and **notch filters** are used.* $s=72000\,units$ *and position controller proportional amplification* $K_v=440\ 1/s$.

Another important reason for why command shaping notch filters are preferable compared to the jolt limitation method, arises in applications where more than one undesired frequency component exist. The rule of thumb for jolt limitation takes only vibrations with one frequency component into account, whereas cascaded notch filters can prevent vibrations with several frequency components.

8.4 Reflections and aspects of reshaping the reference

Results show that command shaping notch filters are very effective at preventing critical process vibrations from occurring on both the experimental setup and the portal robot. The question is if the method comes with any drawbacks.
An identified drawback which applies for the notch filter method is not due to the

method as such, the drawback is general for all “softer” profiles. Even though it is not very clear in the diagrams above, it has been observed that the *transient* lag error is a bit larger when notch filters are used compared to the transient lag error when untreated profiles are used, i.e., when the vibrations have faded completely. This is also observed when the jolt limitation method is used or when a more gentle untreated profile is used. That means that the problem is not directly coupled with the notch filter strategy, instead it is a general problem that already exists. The problem arises when there is high friction in the process, in this case mainly on the tracks for the translation units. Depending on the application it is questionable if the larger lag error is a problem at all. With the controller parameters used here the larger lag errors has been in the range of $100 units$ or $0{,}178 mm$ on the Z-axis of the portal robot. If it is necessary to reduce the remaining lag error then well lubricated mechanics could be used, the integral action of the position PI could be increased as well as the proportional amplification.

As we have seen already, the notch filter method has a lot in common with the jolt limitation method: the small lag error drawback is the same, both can only prevent vibrations from occurring and both increase the time it takes for a reference profile to reach its final value. However, what we can conclude from experiments is that the notch filter strategy has performed better than the jolt limitation method. Considering the first discussed profile in Section 8.3 the time gain of using notch filters instead of jolt limitation is about $t_{gain} = 0{,}16\, s$ even though the profile is slower. A perhaps very simplified and approximative example can be expressed by a machine performing this motion effectively 6 hours a day having to wait out vibrations. Since the profile operates for about $0{,}25 s$ this means, every day the movement is performed, about

$$\frac{6 \cdot 60 \cdot 60}{0{,}25} times = 86400\, times.$$

Every day

$$86400\, t_{gain} = 86400 \cdot 0{,}16\, s = 13824\, s$$

would be saved. Expressed in hours this means about 3,8 hours! The example should of course be considered from the perspective that the worst case scenario has been assumed, i.e., that the machine is totally unable to make further operations if there are any vibrations present. This assumption is of course not entirely true but an indication that there may be time savings to collect using notch filters. Another aspect is wear of mechanics, which is also reduced by both notch filters and jolt limitation. However, notch filters seem to perform better based on experiments conducted during this project. The only case obvious to the author where notch filters should not be used is when it is important not to distort the reference profile obtained by the Path Generator.

9. Conclusions and future work

The initial objective for this project was to find a control solution to actively damp vibrations in robot applications. The main restriction was that no additional sensors should be involved for control. This meant that the control strategy could only rely on e.g. encoder data and/or current measurements.

That called for some type of model based feed back solution. Another important aspect that arose during the initial stage of the work was that the control method should be implementable on different types of robots at different operating states. This, however, called for a somewhat more general solution. To change the whole control strategy, i.e., to replace the whole cascaded PI structure, was out of the question since this solution works well and has been established in industry for a long time. Therefore the existing control strategy was studied in order to understand how it would be possible to modify it so that these demands could be met.

A simple model based state feedback solution was created. The purpose was to test the feedback idea. Position estimates based on current inputs were fed back to the position loop. This solution would meet the objective of active damping. However, the solution would not be general. A good model would at least have to be created for every single robot type or other application were the solution should be implemented. Besides the fact that a lot of work would have to be done, there was the fact that no "real" vibration measurements would be available to compensate for model errors or to validate the model against. This would make the work very cumbersome. The question if the control strategy really had to be "active" in the sense of feedback control arose. If there would be disturbances involved in the considered applications, active damping would be needed. The main vibrations however are not induced by disturbances, they are induced by the movements of the servo motors themselves.

This fact led to the idea of command shaping notch filters, an idea quite similar to jolt limitation. Both methods are based on reshaping of the reference path, both methods need the frequency of the oscillations as a priori knowledge and both methods are general in terms of different applications. If the vibrations are induced by the servo motors and if it is possible to observe the vibrations in available signals, then it is possible to reduce the vibrations. Due to this fact, the main restriction that no additional sensors should be involved was left non-violated. The fact that the solution is not based on an additional sensor opens up for the following observation: it does not matter were in the process the vibrations occur. If they are induced by the servo and if they are observed in e.g. the encoder, they can be prevented.

Experiments with notch filters show that they work very well, even better than the existing jolt limitation method. Moreover, notch filters can easily be cascaded. This makes the strategy suitable for applications where several resonances are involved. The jolt limitation method would probably fail for such applications if the frequencies are not close to each other.

In order to identify the frequencies the controller should not be too stiffly tuned, i.e., the vibrations should be allowed to appear on the servo side. This of course applies for both methods. An algorithm identifying the main vibrations automatically, e.g. maybe by FFT or by estimating the parameter ω_n^2 in a transfer function similar to Equation 6.1 (it does not matter if the other parameters are wrong), would be a nice feature. This would perhaps make the installation

procedure easier. It should also be recalled that the results are based on experiments where only one servo has been active. The theory described in Chapter 5, Section 5.2, that the distortions can be divided into vector components oscillating at two different frequencies would have to be verified by experiments. If the theory is right it is possible to measure the vibrations in each servo and then notch the undesired frequency “servo by servo”. Since the frequency of the vibrations may be different at different operating states of the application (e.g. if it carries a load), a look up table containing the right parameters for each operating state could perhaps be created.

Another important aspect is that the notch filters were implemented in the PLC. A solution of provisional character was created to get in between the Path Generator and the position loop. This would probably not suffice for implementation in industry. If notch filters should be implemented, it is recommended that this is done in the ACOPOS. This would eliminate the Powerlink delay and the notch filters would run at the same cycle time as the position loop. The computational load on the PLC due to the algorithm is probably not very high, but having a task running at 0,8*ms* in parallel to other tasks may not be desired if the other tasks are many and/or demanding.

What the work with this project has shown is that under the experimental conditions notch filters have performed well and even better than the jolt limitation method. A feedback solution would probably not be recommended by the author if no additional sensors are available and if the method should have a general “easy to implement character”.

Hopefully, this thesis may be able to support further work on a good vibration control strategy - even if the problem should be solved differently.

References

In order of appearance:

[1] *The Basics of Motion Control*, B&R Training manual 400, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

[2] Gustaf Olsson, Christian Rosen
Industrial Automation, Applications, Structures and Systems, Department of Industrial Electrical Engineering and Automation, Lund University, Lund, 2005.

[3] *B&R Products 2011*, Operator panels and Industrial PCs, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

[4] *B&R Products 2011*, I/Os, Fieldbuses and Controllers, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

[5] *B&R Products 2011*, Automation Software and Service, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

[6] *Automation Studio Diagnostics*, B&R Training manual 223, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

[7] Gustaf Olsson, Bo Peterson, Olof Samuelsson, Mats Alaküla, Jari Valis, Tore Svensson
Elmaskinsystem, Department of Industrial Electrical Engineering and Automation, Lund University, KF-Sigma, Lund, 1994. (In Swedish).

[8] Tore Hägglund
Reglerteknik AK, Föreläsningar, Department of Automatic Control, Lund University, Lund, 2007. (In Swedish).

[9] *ACOPOS Control Concept and Adjustment*, B&R Training manual 450, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

[10] Steve L Dickerson, Wayne J. Book
Innovations in Motion Control: Part I-Learned Effort, Control, Motion Fundamentals, 2002, CAMotion, Inc.

[11] Björn Wittenmark, Karl Johan Åström, Karl-Erik Årzén
Computer Control: An Overview, Educational Version 2009, IFAC Professional Brief, Department of Automatic Control, Lund University, Lund, 2009.