Speed Acquisition Methods for High-Bandwidth Servo Drives

(Drehzahlerfassungsmethoden für Servoantriebe hoher Bandbreite)

Vom Fachbereich 18
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Alexander Bähr
geboren am 28.8.1975 in Karlsruhe

Referent:	Prof. Dr.-Ing. Peter Mutschler
Korreferent:	Prof. Dr.-Ing. Eberhard Abele
Tag der Einreichung:	7. Juli 2004
Tag der mündlichen Prüfung:	2. Dezember 2004

D17
Darmstädter Dissertation

Preface

This thesis is the result of a 4-years project at the Institute of Power Electronics and Control of Drives, Darmstadt University of Technology.

I thank Prof. Mutschler for his interest in my work, his support with both technical and presentation issues, and critical discussion of the results.

To Prof. Abele, I thank for his interest and for acting as co-advisor.

I thank the DFG Deutsche Forschungsgemeinschaft for financially supporting my projects MU 1109/6-1 and -2.

I would like to thank all my colleagues at the institute for their support and comments, a good working atmosphere, and many useful discussions. Special thanks to J. Fassnacht for his support with getting started.

Many non-scientific issues are important for an experimental project. I appreciate the work and advice of the institute's technical staff, draftswoman, and administrative staff.

I thank the students who did their diploma theses in my topic and whose results have been used in this thesis. Even those who only found out that and why their issue did not work helped the project considerably.

Last but not least, I thank my wife and parents for encouragement, support and interest during my study and PhD time.

Wetzlar, 6.7.2004

Abstract

A servo control needs the actual values of speed and position. Usually, the latter is computed from the signals of a position encoder; its 1st derivative is smoothed by a low-pass filter and used as actual speed signal. A number of enhanced and alternative methods is experimentally investigated in this thesis. Based on an equal steady-state behavior, the controlled servo's dynamic stiffness is used as the performance measure. The used setup has a special feature: because of its rather high resonant frequencies (870 and 1280Hz), the encoder's oscillation against the drive can no longer be neglected.

The mechanical resonance can be met by using notch filters to damp the resonant frequencies out of the controller spectrum, leading to major improvements. By identifying and modeling the mechanical setup at different levels of precision, observers were designed to provide an alternative actual speed signal, leading to a further improvement; however, active damping was not possible due to the configuration of the resonant system. The use of a state controller allowed active damping, but at the expense of reducing control gain and thus dynamic stiffness.

The signals of an optical position encoder show characteristic errors. Using measures to correct those errors, it was tried to improve steady-state speed quality and allow a higher control gain. Two table-based and one on-line adaptive method were investigated. As stated in previous works, the correction of signal records resulted in a considerable error reduction with all methods. However, the improvement due to correction used in the control loop is small, because the loop gain is quite low at the error signals' high frequencies.

The use of an acceleration sensor for speed acquisition has the advantage that the signal is integrated instead of derived, reducing noise instead of amplifying it. The improvement in the experiments was only low, because oscillation and not noise is the problem limiting control gain. Another advantage of the acceleration sensor is a much easier fixing compared to the position encoder. By mounting the acceleration sensor at an optimal location concerning oscillation, it is possible to damp the oscillation considerably without any knowledge about the resonant frequencies.

The thesis is completed by theoretical investigations of speed quality and dy-

namic stiffness, investigations of drive-side and load-side behavior and necessary computation power for the investigated algorithms.

Zusammenfassung

(Abstract in German)

Zur Regelung eines Servoantriebs ist die Information über die Istwerte von Drehzahl und Lage notwendig. Üblicherweise wird hierzu aus den Signalen eines Winkelgebers die Lage berechnet; deren 1. Ableitung, geglättet durch ein Tiefpassfilter, wird als Drehzahlsignal verwendet. Eine Reihe von erweiterten bzw. alternativen Verfahren werden in dieser Arbeit experimentell untersucht. Als Vergleichsgröße wird bei gleicher erreichter Drehzahlqualität die dynamische Steifigkeit der Regelung herangezogen. Der verwendete Versuchsstand weist eine wesentliche Besonderheit auf: durch die hohen mechanischen Eigenfrequenzen (870 und 1280Hz) kann die Schwingungsfähigkeit des Gebers gegenüber dem Antrieb nicht mehr vernachlässigt werden.

Den mechanischen Eigenschwingungen des Versuchsstandes kann begegnet werden, indem mit Hilfe von Notch-Filtern die fraglichen Frequenzen aus dem Spektrum des Drehzahlsignals ausgeblendet werden. Dadurch ließen sich deutliche Verbesserungen erzielen. Weiter kann die schwingungsfähige Mechanik identifiziert und modelliert werden. Auf Basis unterschiedlich genauer Modelle wurden Beobachter ausgelegt, die eine Alternative zur Drehzahlermittlung darstellen. So wurde eine weitere Verbesserung erreicht; eine aktive Schwingungsdämpfung war jedoch aufgrund der ungünstigen Konfiguration des Mehrmassensystems nicht möglich. Diese kann erst mit einer Zustandsregelung erzielt werden, geht jedoch -bei gleicher erreichter Drehzahlqualität im Gleichlauf- auf Kosten der Regelverstärkung und somit der dynamischen Steifigkeit.

Die Signale eines optischen Winkelgebers weisen charakteristische Fehler auf. Mit Hilfe von Verfahren zur Korrektur dieser Fehler wurde versucht, die Drehzahlqualität im Gleichlauf zu verbessern, um eine höhere Regelverstärkung zu erlauben. Zwei tabellenbasierte Verfahren und eines, das die Korrekturdaten in Echtzeit generiert, wurden untersucht. Es konnte zwar bei der Korrektur von aufgenommenen Zeitverläufen -wie in Vorarbeiten beschrieben- mit allen Verfahren eine wesentliche Verringerung des Drehzahlfehlers erreicht werden. Die Verbesserungswirkung bei Einsatz im Regelkreis blieb aber gering, da dieser die

im hohen Frequenzbereich angesiedelten Fehlersignale ohnehin weitgehend ignoriert.

Bei Verwendung eines Beschleunigungssensors wird dessen Signal zur Berechnung der Drehzahl integriert und nicht differenziert; dadurch verringert sich das enthaltene Rauschen wesentlich. Die Verbesserung im Experiment blieb jedoch gering, da die Schwingungsproblematik des Versuchsstands und nicht das Signalrauschen die Regelverstärkung begrenzt. Ein weiterer Vorteil des Beschleunigungsgebers ist jedoch die einfache Anbringungsmöglichkeit mit geringen Genauigkeitsanforderungen - dadurch ist eine Montage auch an anderen Stellen als der B-Seite des Antriebs möglich. Mit einem an der Kupplung zwischen Antrieb und Lastmaschine angebrachten Beschleunigungsgeber konnten deutliche Verbesserungen erzielt werden, ohne dass dazu eine Kenntnis der Resonanzfrequenzen notwendig ist.

Die Arbeit wird abgerundet durch theoretische Überlegungen zu Drehzahlqualität und Steifigkeit, Untersuchungen des antriebsseitigen und lastseitigen Verhaltens sowie zur nötigen Rechenleistung für die untersuchten Verfahren.

Contents

List of symbols

All quantities are measured in SI units, unless otherwise stated.

Small bold letters denote vectors, and bold capitals are matrices. $\mathbf{I}$ is the identity matrix. $\mathbf{X}^T$ denotes the transposed matrix, $\mathbf{X}^{-1}$ is the inverse matrix.

The superscript $\hat{x}$ denotes estimated values (in connection with observing). The superscript x^* denotes a reference value.

$\mathbf{A}$	system matrix of a state-space system
a_i	denominator coefficients of a (continuous-time or discrete-time) transfer function
A_x, A_y	offset errors of encoder track signals
$\mathbf{B}$	input matrix of a state-space system
b_i	numerator coefficients of a (continuous-time or discrete-time) transfer function
$\mathbf{C}$	output matrix of a state-space system
c_{01}, c_{12}	spring constants in the resonant system models
c_i	polynomial coefficients
C_{dyn}	dynamic stiffness
C_{static}	static stiffness
count	state of the encoder line counter, equivalent to the coarse position
d_{01}, d_{12}	damping coefficients in the resonant system models
f	frequency in Hz
$F(s)$	transfer function in Laplace domain
$F(j\omega)$	transfer function in Fourier domain
$F(z)$	transfer function in discrete-time Laplace domain
$F_O(s)$	open-loop transfer function
f_i	feedback coefficients of HN-IIR predictive filter
i_q	current in quadrature axis
j	imaginary unit
J	overall plant inertia
J_0, J_1, J_2	inertias in the resonant system models
k	discrete time, k = 0,1,2,3,...
k_i	observer feedback constants

K_R	speed controller gain
K_S	plant gain, $K_S = k_T/J$
k_T	torque constant, torque $= k_T * i_q$
K_V	position controller gain
K_α	acceleration feedback gain
M	interpolation polynomial order (with predictive filters)
N	filter order, encoder line count, or generally an integer number
NG	noise gain
O_x, O_y	offset errors of encoder track signals
p	prediction step, $1\hat{=}$sampling time
$\mathbf{Q}$	weighting matrix for system states (linear-quadratic controller and static Kalman filter)
$\mathbf{R}$	weighting matrix for input or sensor outputs (linear-quadratic controller and static Kalman filter, respectively)
s	Laplace-transform operator
S/H	sample-and-hold lock
T	control system sampling time
T_{01}, T_{12}	spring torques in the resonant system models
T_I	integrator time constant of the PI speed controller
T_S	time constant representing the plant's phase lag, caused by digital control and stator time constant
$T_{control}$	time constant representing the digital control system's delay
T_{filter}	time constant representing the speed filter's delay
T_{load}	load torque
T_α	acceleration sensor delay, or complete delay in the acceleration feedback path
W	weighting factor
x	cosine track signal of the encoder, or generally a signal
x_c	encoder cosine signal, corrected
y	sine track signal of the encoder
y_c	encoder sine signal, corrected
z	discrete-time Laplace transform operator
α	acceleration
α_{sens}	acceleration as measured by acceleration sensor
δ	loss angle
Δ	phase error between encoder tracks
Δ	successive difference operator
$\Delta\theta$	maximum position deviation

$\Delta\theta_{rat}$	maximum position deviation due to a load torque change by the nominal torque
θ	rotational position
σ	r.m.s deviation
$\sigma\Omega$	r.m.s speed error
$\sigma\theta$	r.m.s position error
Ω	rotation speed
Ω_0, Ω_1, Ω_2	rotation speed of certain masses in the resonant system models
ω	frequency in rad/s
ω_2	a speed signal computed without regarding the line counter

1. Introduction

1.1. Focus of work

The PI-speed-control using a position encoder for feedback is the standard configuration in servo control. For highly precise and dynamic systems, an encoder with sinusoidal signals with 1000..5000 periods per turn is necessary. In control system design, one design goal is a fast rejection of load changes. To achieve this, it is necessary to minimize the filtering delay, in order to make high controller gains possible. On the other hand, smooth turning at constant reference speed is desired, requiring a filter with good smoothing effect and only moderate controller gains.

Originally, this work was intended to investigate only the tradeoff between low delay and good suppression of high-frequency noise. The experimental setup was built up using small motors and stiff couplings, so that mechanical oscillations would not be a problem. However, it turned out that mechanical oscillations are in fact the key problem limiting controller gain, despite their high resonance frequencies of about 900 and 1300 Hz. Thus, observers using a model of the resonant system were also used (see chapter 3), and yielded best results.

Papers and theses dealing with mechanical resonance [22, 33, 36] often regard plants where the controlled servo motor oscillates against the load inertia(s). However, in the case regarded here, the controlled servo and the shaft extension on which the encoder is fixed oscillate against each other, meaning that sensor and actuator are different parts of the resonant system. This makes an important difference: Proportional speed control, even with zero delay, does not stabilize the system at all the resonance frequencies. This is a problem when using methods intended to produce a smooth speed signal with low phase lag, e. g. the speed observer using acceleration signal from sensor I (chapter 6).

The thesis starts with the industry-standard method of using the filtered derivative of the position signal as actual speed (chapter 2). Speed observers with different kinds of plant model are considered in chapter 3.

The signals of a sinusoidal encoder have several systematic errors. These errors can be detected and corrected by either table-based or on-line methods (chapter 5). This correction has the potential to improve the quality of the actual speed signal without contributing to plant delay.

Chapter 6 investigates several methods to use an acceleration signal, acquired either from a dedicated acceleration sensor or by double derivation of the position signal.

The discussion of theoretical background and a large part of the comparison of methods has been published in [3]

1.2. Experimental setups

1.2.1. Setup I

A Schematic diagram of setup I is shown in fig. 1.1. A PC working under realtime OS RT-Target [21] does the supervision and security functions, computes position and current control, and provides real-time data recording and saving. The current command is sent as analogue data to the current controller, which is an analogue bang-bang controller working in the stationary reference frame [19]. The circuit send the pulse pattern to a proprietary inverter powering the drive servo. The load servo is powered by an industrial inverter in current control mode. It has the only function of creating disturbances to test the drive's control.

The mechanical part, as shown in fig. 1.2, consists of two coupled 2.2 kW permanent magnet servo machines. The control methods are implemented for the MOOG G465-204 servo on the left side ("drive"). This motor was chosen mainly because of its strong B-side shaft extension, which made it possible to mount an extension for the encoders. It also features no magnetic latching and a low inertia of $4.6 kgcm^2$. The servo's rated torque is 4.3 Nm, the torque constant is $k_T = 0.533 Nm/A$ in the sense of fig. 3.1 and 3.4, the overall inertia of the setup is $J = 20.05 kgcm^2$. The AMK DS 5-5-6 ("load servo") on he right is controlled by a commercial servo drive from AMK, which is used in current control mode to simulate the changing load of a work machine. Its current control's response to load changes is approximately a ramp function, reaching the new setpoint after $500 \mu s$.

At the controlled servo, at the opposite site to the coupling, an incremental position encoder with sinusoidal signal output and an acceleration sensor are fixed. The fixing required a short extension sleeve, as shown in fig. 1.2(b). This extension is one reason why the encoder resonance plays an important role in this setup (see chapter 3). The acceleration sensor's brass disk is fixed using a ring with four grub screws to fix itself on the shaft extension and four to press on the brass disk. Under the grub screws, some material of the ring had to be taken away, otherwise the ring can get stuck because of deformation of the shaft material.

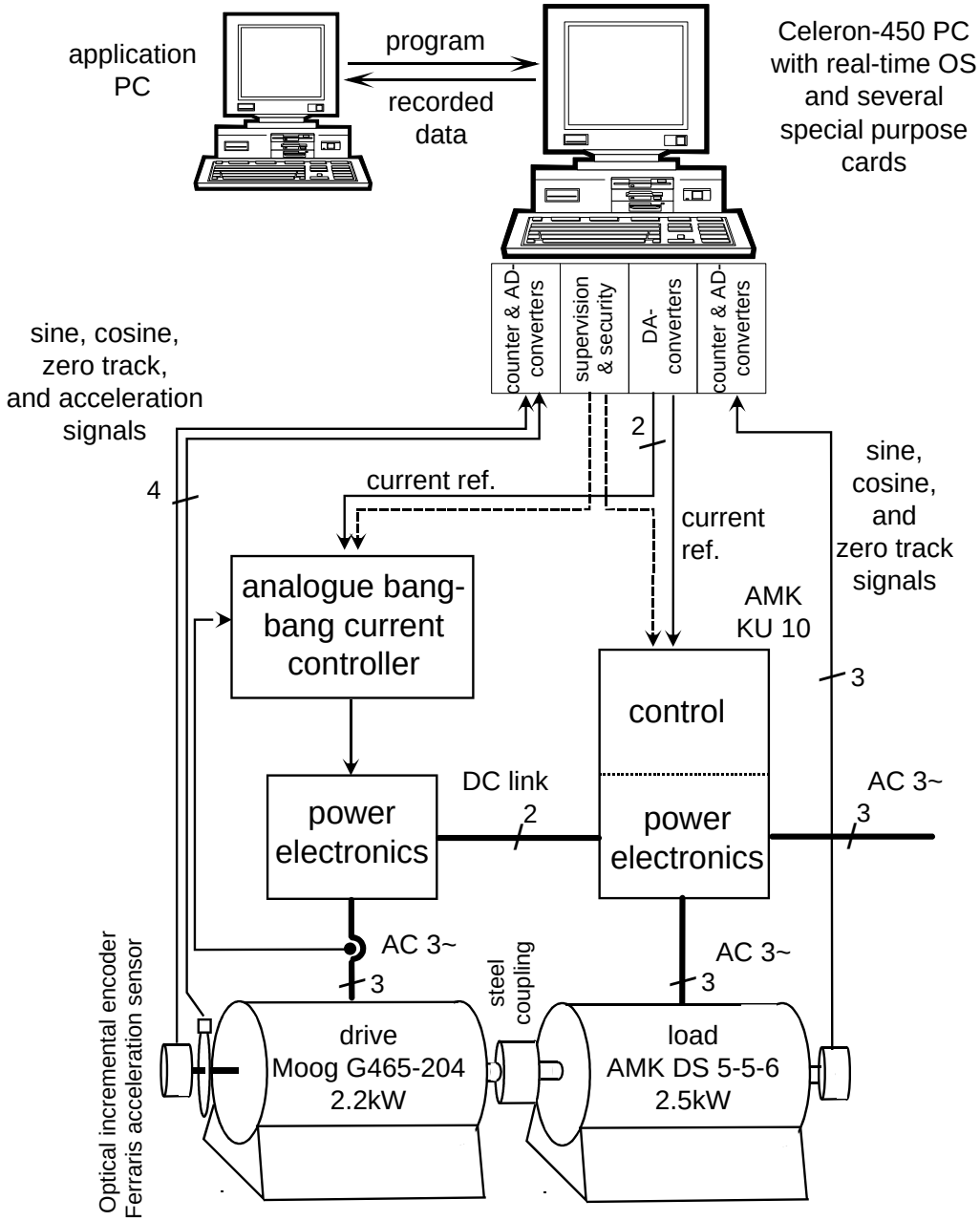


Figure 1.1.: Schematic diagram of experimental setup I

(a) (b)

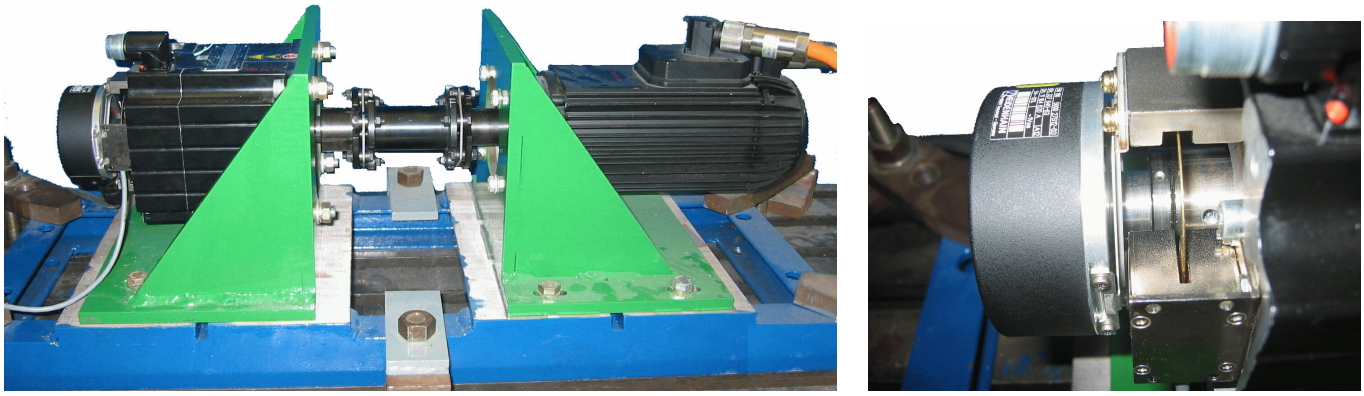


Figure 1.2.: Experimental setup I

Two position encoders were tested on this setup: A Huebner HOGS80 with 2048 lines (bought in year 2000), and a 5000-line Heidenhain ERN180 from 2002. Both provide an hollow shaft with 25mm diameter; fixing to the servo case is done by a steel plate. Both provide an equal precision compared to their line count; however, the 2048-line encoder's signals are mainly subject to noise, while the 5000-line encoder has low noise but considerable systematic errors (see chapter 5).

The sensor signals are digitalized by 12-bit AD-converters. The usual arctangent method is used to compute the position. Lines are counted on an PLD that is fed by AD converters' most significant bits, as it has been proposed in connection with oversampling [48]. This method reduces the hardware effort by dispensing with the hysteresis comparators and the pre-amplifiers necessary to decouple them from the converters. However, the converters must be triggered at least twice during one encoder period. They are triggered several times during one control cycle, disregarding the analog value. In other words, there is a velocity limit of

$$\Omega \leq \frac{2\pi/N/2}{T_{ADC}} \tag{1.1}$$

(Ω = rotational speed in rad/s, N = encoder line count, T_{ADC} = sampling time of the analog-digital-converters). With $T_{ADC} = 2.4\mu s$, the velocity limit is 2500/min using a 5000-line encoder.

The acceleration limit is due to the fact demand that a sequence of samples must be interpreted correctly as forward or backward movement. It is usually beyond practical accelerations.

The position is computed in the software by evaluating the line counter for coarse position and analogue samples for interpolation with the usual arctangent

equation [48]

$$\theta = K_1 * count + K_2 * atan2\left(\frac{y}{x}\right) + K_3(sgn(x), sgn(y)) \tag{1.2}$$

where θ is the position angle, atan2 represents a four-quadrant arctangent, y the sine and x the cosine track signal, and K_i correction factors to keep θ continuous and in appropriate domain. The speed is computed using various methods - see the following chapters.

The standard deviation of speed at setup II, computed by the difference quotient at $21\mu s$ sampling rate, is 0.005 rad/s and 0.283 rad/s for the 5000-line and 2048-line encoder, respectively, with the servo not moving. These values represent the noise in the sensor signals. When moving, the 5000-line encoder's standard deviation increases to 0.132 rad/s, while the 2048-line encoder's deviation does not change measurably (see chapter 5). This means that the 5000-line encoder has significant systematic errors.

The setup has a Ferraris acceleration sensor mounted together with the encoder. The working principle of this sensor is shown in fig. 6.3: It consists of a metal disc that rotates with the servo shaft, and one or two stationary sensor units. A sensor unit is a permanent magnet with a coil around it. The metal disc "sees" the magnet's field as a traveling one, thus eddy currents are created proportionally to the servo speed. As soon as the disc is accelerated, the eddy currents change and induce a voltage back in the coil. This voltage is used as the acceleration signal.

The acceleration sensor used for the final measurements is a Huebner ACC94 type. With a brass disk of 85mm diameter, it has a a signal amplitude of $2.8\frac{\mu V}{rad/s^2}$ and an intrinsic phase lag of about $60...80\mu s$.

Position and speed control are computed in a standard 450MHz Celeron PC. It works under realtime OS RTTarget; interrupt latency times of about $5\mu s$ were found to be the worst case if no program is running in parallel. Control timing is done by a commercial PCI-bus timer card, while two ISA cards decode the sensor signals from controlled servo and load servo side.

1.2.2. Setup II

Setup II is shown in fig. 1.3. It consists of a 9 kW asynchronous servo motor, with the encoder fixed on the shaft end that is normally used to couple the load. It is driven by a BBC servo controller in speed control mode, the speed reference is set by hand.

This motor was chosen for run-down experiments required in chapter 5 because of its large inertia ensuring a long run-down time.

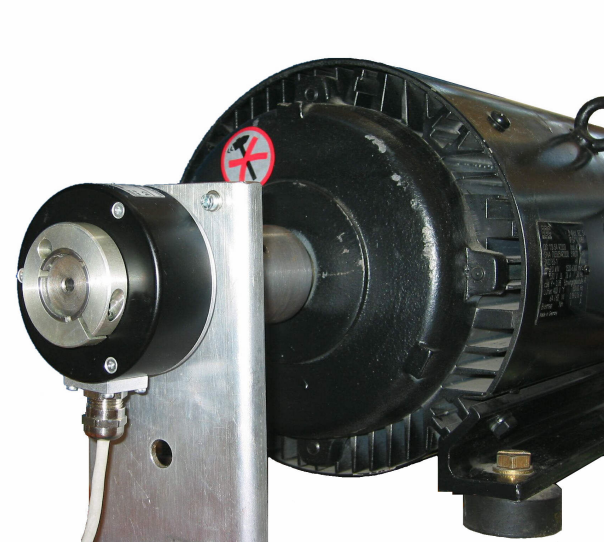


Figure 1.3.: Experimental setup II

1.3. Control loop

For all measurements, the usual cascade of position, speed, and current controllers is used. The addition of an acceleration control loop is discussed in chapter 6, and tests concerning state control are explained in chapter 4.

Current control for the controlled servo is done by a circuit working with hysteresis comparators, as proposed by Kazmierkowsky et al. [19]. This controller gets reference values in stator-fixed coordinates from the PC and switches voltage vectors that will lead the current space vector directly to that location, with respect to hysteresis tolerance and minimum turn-on times.

The reference current is computed by a P-PI-cascade control implemented on the PC using different speed acquisition schemes. Direct axis current is set to zero. The speed controller was designed according to the well-known symmetrical optimum [7], based on the load acquisition filter's or observer's delay time constant. The current control works with maximum slope of 69 A/ms. Supposing a current command step change from zero to the max. current of 10 A (alpha-beta), this is approximated as a 1st order delay of $144\mu s$. The deadtime of the digital speed control loop is

$$T_{control} = \frac{1}{2} * T + T_C \tag{1.3}$$

where T is the sampling time and T_C the time reserved between A/D and D/A conversions. With the standard timing of $100\mu s$ sampling time and $40\mu s$ calculation time, the summed time constant of the speed loop is $T_S = 234\mu s$ plus filter

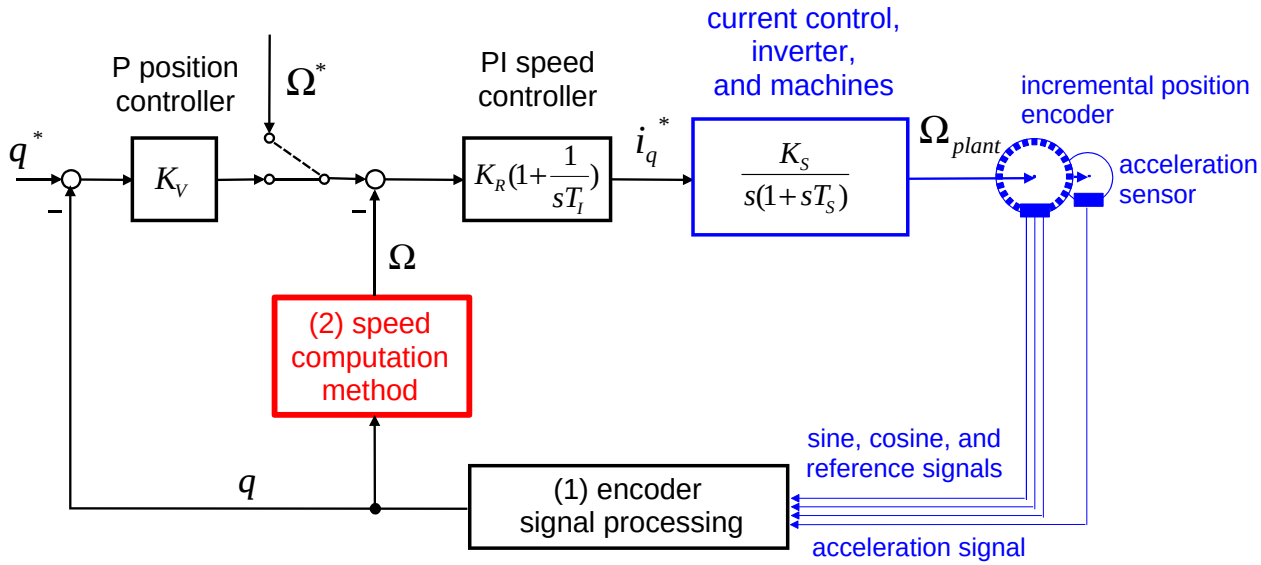


Figure 1.4.: Block diagram of the control system

delay. The tuning rules of the Symmetrical Optimum are then

$$T_S = T_{control} + 144\mu s \tag{1.4}$$

$$K_R = \frac{1}{2K_S(T_S + T_{filter})} \tag{1.5}$$

$$T_I = 4(T_S + T_{filter}) \tag{1.6}$$

$$K_V = \frac{1}{2T_I} \tag{1.7}$$

When using observers, there are two possibilities to design the speed control. First, the observer's characteristic polynomial can be approximated by its linear part in order to get an approximate time constant, which is then used for speed controller design.

Second, the method from state control can be used. In state control, the controller feedback is usually designed independently from the observer, i. e.

assuming that all state variables were measurable without delay. Care must only be taken that the observer poles are "far" left from the control system's poles. This method is based on the separation theorem, which states that the eigenvalues of the closed controller loop with observer will be those of the control loop plus the observer's. For the proof of the separation theorem [7], no assumptions about the feedback vector need to be made, so it extends to the proportional controller feeding back only one state variable. The integral part is added in cascade as well as state control to provide steady-state precision.

If the PI speed controller is designed according to the symmetrical optimum assuming a delay-free observation, the gain is too high to produce working results. Thus, an extension of the symmetrical optimum was used where the proportional gain was chosen according to the steady-state speed error limit (see section 1.6), and the integral gain is computed such that in the bode plot, the maximum phase occurs where the logarithmic absolute value graph crosses zero. For an open speed loop transfer function

$$F_O(s) = \overbrace{K_R * \left(1 + \frac{1}{sT_I}\right)}^{\text{PI speed controller}} * \overbrace{\frac{K_S}{s\left(1 + sT_S\right)}}^{\text{controlled system}} \tag{1.8}$$

the rule is to set the integration time constant to

$$T_I = \frac{1}{\left(K_R K_S\right)^2 * T_S} \tag{1.9}$$

no longer regarding the observer's delay. The position controller gain was chosen using (1.7). This method was used for the two and three mass observers. Because of their high order and thus high delay time constant, the standard symmetrical optimum would have led to inacceptably low gains.

However, this method is only applicable for high-gain controller designs. With certain controls, moderate gains also have to be chosen by hand. In those cases, the integration time constant was chosen according to the damping optimum. This tuning rule states that an optimum is achieved if, in the denominator of the closed loop's transfer function, all "double ratios" of the quotients are equal to 2 [9]. If both K_R and T_I are chosen by this rule, the result is the Symmetrical Optimum. If K_R has been fixed, only one condition can be satisfied by selecting T_I, leading to the tuning rule

$$T_I = \frac{2}{K_R K_S} \tag{1.10}$$

1.4. Theory of servo control performance

1.4.1. Dynamic stiffness vs. controller gain

The static stiffness C_{static} is a value that characterizes the stiffness of a torsional spring, where the application of a torque T to its ends will cause a proportional angular deviation θ:

$$C_{static} = \frac{T}{\theta} \tag{1.11}$$

In a usual position-controlled servo system, a load torque will not cause a permanent position deviation because it is compensated by the speed controller's integral part. However, a change in the load torque ΔT will cause a temporary position deviation; the maximum position deviation $\Delta\theta$ is again proportional to the torque change and is often used to characterize the control system performance. Using both values, the dynamic stiffness can be defined analogously to the static stiffness as

$$C_{dyn} = \frac{\Delta T}{\Delta \theta} \tag{1.12}$$

Usually, the dynamic stiffness refers to a stepwise load torque change.

In fig. 1.5(c), simulated results for dynamic stiffness are shown. They were simulated using P position control and PI speed control with different gains and speed acquisition using a three-inertia system resonant observer (see fig. 1.4 and section 3.4). The simulation results correspond very well to the experimental results. To illustrate the dynamic stiffness data, a second ordinate axes shows the maximum position deviation that will result from a load torque change by 100% of the drive's nominal torque.

A theoretical discussion of the relationship between controller parameters and dynamic stiffness was published by Weck and others [20]. It discusses the cascade control structure shown in fig. 1.4, assuming a rigid-body setup with no mechanical resonance and neglecting the delays of current control loop and speed acquisition. A first approach, also neglecting the speed controller's integral part, yields the equation

$$C_{dyn,1} = K_V K_R k_T \tag{1.13}$$

This equation gives the coarse idea that a high dynamic stiffness can be achieved by raising the controller gains in general, while the relationship between K_V and K_R is not important; thus, special optimization of the position controller does not promise better results. However, the values from (1.13) are far away from the experimental and simulation results; they are shown in fig. 1.5(a).

In another approach in [20], the speed controller's integral part is regarded,

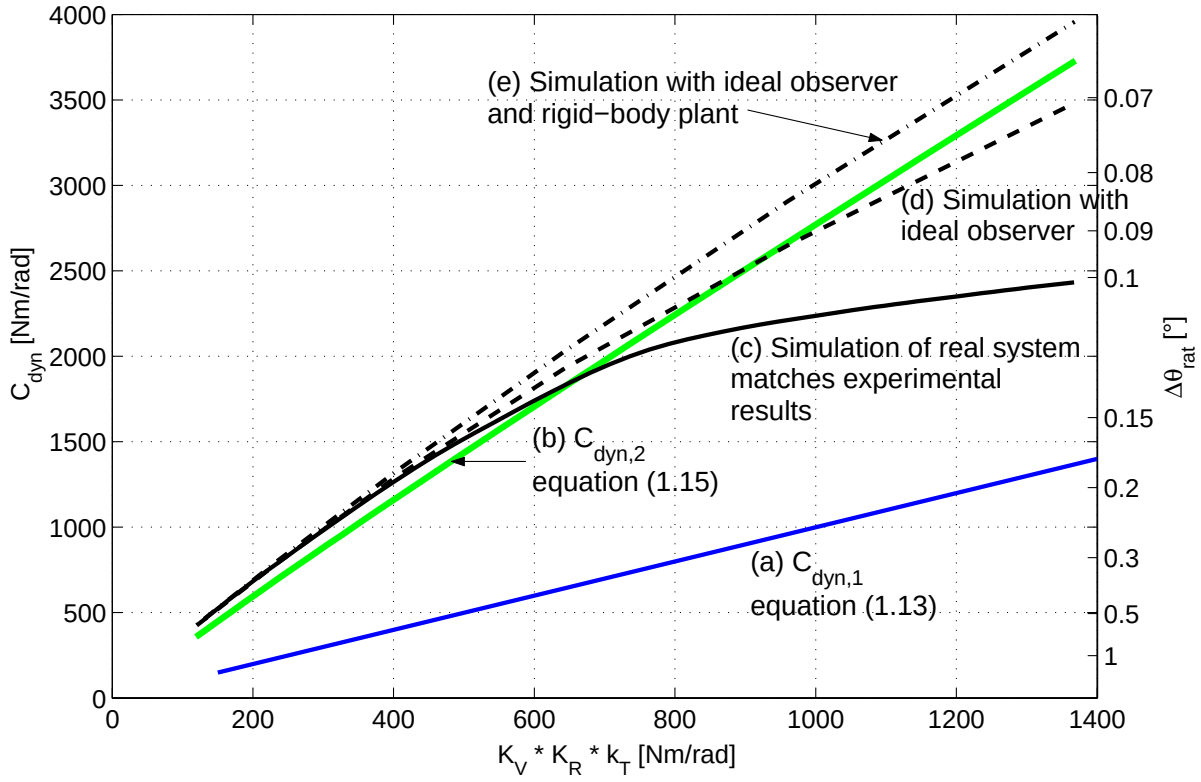


Figure 1.5.: Simulation results for dynamic stiffness vs. controller gain

but other assumptions are made [1]. The dynamic stiffness equation becomes much more complicated:

$$\begin{aligned} C_{dyn,2} &= K\left(1+\frac{e^{-D\frac{\pi}{\sqrt{1-D^2}}}}{\sqrt{1-D^2}}\right)^{-1} \\ \text{where } K &= \frac{K_R k_T\,(1+K_V T_I)}{T_I} \\ D &= \frac{1}{2}\sqrt{\frac{K_R k_T T_I}{J\,(1+K_V T_I)}} \end{aligned} \tag{1.15}$$

The results from (1.15) are shown in fig. 1.5(b). They match the experimental results quite well for low-gain controls, but overestimate the dynamic stiffness achievable with high gains.

Using high order observers, it seems necessary to regard the observer's considerable delay in detecting load changes. Unfortunately, it is no longer possible to solve the problem symbolically. Instead, simulations were carried out using different models. Fig. 1.5(e) shows results for a rigid-body plant and ideal observing (i.e. the plant values are fed directly to the controller), graph (d) shows results with a resonant plant and (c) the final simulations with resonant plant and Luenberger observer, which correspond very well to experimental results with the same configuration.

The most significant result is a "saturation" of dynamic stiffness for high control gains, which is caused by both the resonant system and the observer (thus, graph (d) shows values between (c) and (e)). The observer shrinks performance because it needs time to detect that the load has changed; the controller's reaction is also delayed by this time. The resonant system itself is limiting performance because after the load mass is accelerated, time is needed before the springs are stretched and the drive mass is affected; this time also delays the reaction. The solution to the latter problem would be a sensor fixed on the load mass, as discussed in sec. 7.3.

This "saturation" is the reason why the strongly increased controller gains using observers did not result in an appropriate rise of dynamic stiffness (fig. 3.9).

[1] The disturbance transfer function, neglecting current control and speed acquisition delay, is

$$\frac{\Delta\theta(s)}{T_{load}(s)} = \frac{s}{s^3 J + s^2 K_R k_T + s K_R K_T (\frac{1}{T_I} + K_V) + \frac{K_R k_T K_V}{Ti}} \tag{1.14}$$

For drives with a bandwidth much larger than 50 Hz, the last numerator term can be neglected.

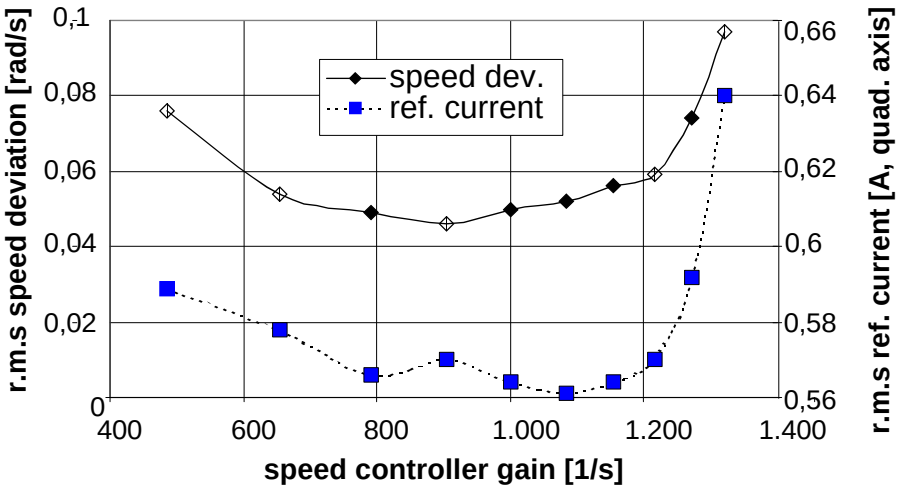


Figure 1.6.: Experimental results for steady-state speed error vs. controller gain

1.4.2. Steady-state smoothness vs. controller gain

Speed errors at steady state are caused by errors in the sensor signals, which are amplified by the control and applied to the drive as real current, causing energy consumption and real deviations. In addition, the weakly damped resonant poles of the system move towards the instable region when the controller gain is raised.

Fig. 1.6 shows the relationship between controller gain, speed quality and motor current. The data were measured on setup I using different PI speed controls with 1st order filters to acquire the actual speed. There is a optimum around controller gain 1000 where the speed deviation is minimal. For higher gains, both deviation and current are raised dramatically, because the plant's resonance is excited; this is also clearly audible. With lower gains, the deviation also rises, this time because of disturbance forces caused by slot latching; the controller cannot properly compensate them because of its large phase lag.

Mechanical resonance limits both dynamic stiffness and speed quality. Stiffness is limited because the oscillation can produce a large overshoot of the position deviation, reducing the stiffness considerably.

1.5. Simulation model

This thesis relies on two different kinds of measurements: speed quality at steady state and dynamic stiffness against load changes. The speed quality is very hard to simulate, because the exact kind of noise in all system parts, model uncertainties, damping coefficients etc. need to be known very exactly. It was not possible to

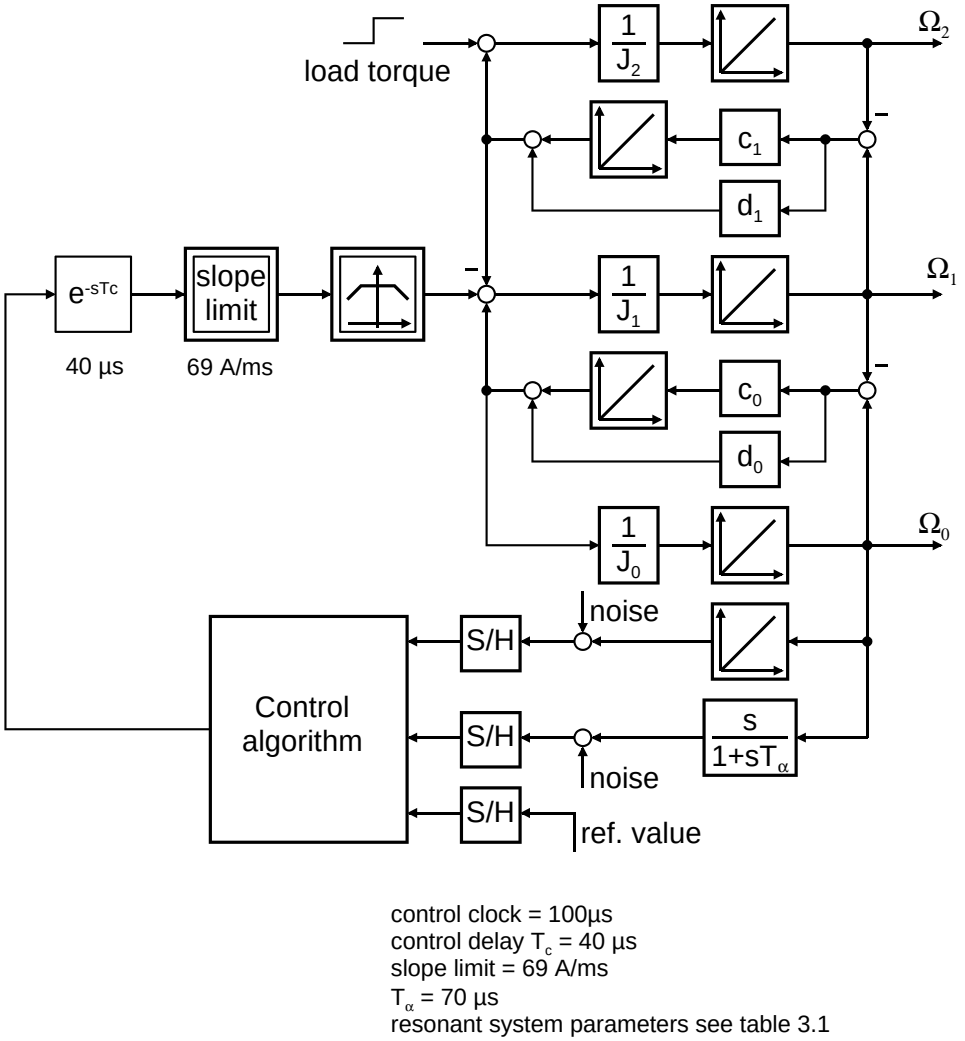


Figure 1.7.: Simulation model

find a model that simulates the measured data correctly. The reaction to a load change, however, does mainly depend on the system and control parameters and is therefore possible to simulate.

The simulation model is shown in fig. 1.7. Its main feature is a three-inertia mechanically resonant system. The system model is shown in fig. 3.3(b), identification and modeling are explained in sec. 3.4.1.

The coupling between both servos has a limited static stiffness, which reduces the load-side dynamic stiffness considerably (see sec. 7.3). This static stiffness should equal the model parameter c_{12}; however, there are large errors because the model was identified to mirror only the frequency-domain behavior. The best way to simulate the load-side dynamic stiffness correctly was to use a modified model having small errors in the transfer function, but with $c_{12} = 12107$ Nm/rad. The model parameters are shown in the last row of table 3.1

The current control used is an analog bang-bang controller, that keeps the current space vector in a specified area around the reference using hysteresis comparators [19]. Any reference value change will be followed as fast as the DC link voltage and stator inductance allow. This is approximated by a slope limit with the value of DC voltage / stator inductance.

The current-torque relationship is not exactly linear because of saturation. It has been implemented in the simulation model as a static characteristic according to the motor's datasheet.

The used simulation model neglects some features of the setup. For instance, the current control might also be modeled more precisely using ideal comparators and a lookup table; the motor can be represented using the linear model in field-oriented reference frame. This will reduce the simulated stiffness of high-gain controls a little, strongly depending on how the minimum turn-on time is implemented. As the model with the approximated current control works considerably faster and the difference in results is neglectable, the simplified model was chosen.

The load servo has strong slot latching, that is visible in the Fourier transform of steady-state signals and certainly affects the speed quality at steady state. However, when investigating load changes with position control, the position range is only some milli-rad. Thus the slot latching torque, which has a major period of 18 per turn, is constant inside this region and is compensated by the speed controller's integral part. Thus, slot-latching was not simulated.

Using the final model, the measured stiffness values could be reproduced within a tolerance of $\pm 100 Nm/rad$. The stiffness of observer controls is simulated too low, while it is simulated too high for controls using a filter.

1.6. Investigation Method

The different speed acquisition filters were investigated experimentally for their closed loop performance. The comparison is based on equal steady state behavior, i. e. in all cases the control was tuned to produce an equal r.m.s. speed deviation at steady state. The 2048-line encoder was tested a a reference speed of 2π rad/s (60 1/min), with a maximum speed error of 0.1 rad/s. The 5000-line encoder was tested at π rad/s (30 1/min), the speed error limit was 0.07 rad/s. Both error limits represent highly dynamic controls, the actual speed quality is comparable. In every case, the speed was computed by the difference quotient

$$\Omega = \frac{\theta(k) - \theta(k-1)}{T} \tag{1.16}$$

with no additional filter to keep the measurements comparable.

While the relationship concerning steady-state quality between different controls is reproducible, the level of speed quality varies by about $\pm 0.01 rad/s$, mainly depending on the ambient temperature and a warm-up of the setup. This problem was resolved by using the control with 1st order filter for speed acquisition as a reference: Care was taken that the speed quality of a control under test was comparable to the 1st order filter control once chosen. In experiments within a short time, the results can be reproduced within a tolerance of $0.005 rad/s$.

In a subsequent experiment, the dynamic stiffness against stepwise load changes between -50% and +50% of the controlled servo's nominal torque was measured. The dynamic stiffness was measured using the maximum deviation during through about 20 load changes, with equation 1.12. Because of the elastic coupling, the plant has only a limited static stiffness, which was measured as 12107 Nm/rad. Because the load side encoder is less precise than the encoders used on the controlled servo side, it was considered best to measure the position deviation on the controlled servo side and add the static stiffness as

$$C_{dyn,load} = \frac{C_{dyn,servo} * C_{static}}{C_{dyn,servo} + C_{static}} \tag{1.17}$$

(where $C_{dyn,servo}$ is the dynamic stiffness computed from servo-side measurement).

A third bar in most graphs shows the simulated dynamic stiffness. It is potentially more precise than the measured one in two ways:

The used load servo controller can create only slow torque changes taking a time of about $500 \mu s$ t, independently of the step height. This fact might falsify the results, because an important issue of this thesis is the active or passive damping of the resonant frequencies, with the lowest resonance around 1000 Hz. This resonance is excited stronger by a step function than by a rather slow ramp.

The load-side dynamic stiffness is available in the simulation with a good precision; the approximation using (1.17) is not necessary.

2. Speed computation using filters

2.1. Introduction

Chapters 2 and 3 deal with the speed computation method, which is block (2) in fig. 1.4.

Speed computation by deriving and filtering the position signal is the standard method in servo control. A position signal is normally available, because position control is also needed. The comparison of filter and observer speed acquisition methods has been published in [1].

The investigated setup has its lowest mechanical resonant frequency at 970 Hz, so the filter's task is to passively damp this frequency in order to allow high controller gains.

Usually, low order low-pass filters are used in the speed acquisition path. After design as continuous-time filters, the coefficients must be transformed to discrete-time domain. This is done using the bilinear transformation [10], resulting in an IIR type discrete-time filter.

As an alternative, FIR low pass filters can be designed using different methods. They have the advantages of needing only low computational precision and immanent stability. On the other hand, IIR filters can realize a much sharper cut-off behavior with much fewer coefficients and -the most important issue in this case- lower delay.

Gees [23] and Brahms [24] compare IIR-low pass filters of first and second order to other numerical speed computation methods such as the unfiltered discrete derivative, the derivative of a spline polynomial, or averaging two discrete differences. Gees makes a theoretical analysis and states that the IIR low-pass filter achieves the lowest output deviation by far. Brahms states that in closed loop, the IIR low pass filter achieves the smoothest speed signal. Polynomial speed estimators and predictors are HN-FIR filters of low order, with p = 0 for estimators. These methods are discussed in section 2.4.1.

Notch filters can be used to additionally damp the resonant frequencies. They are always used together with a low pass filter, which is necessary for a deriving filter since the numerator order must not exceed the denominator order. It was found that the notch part contributes only little to the delay.

For the control of two-inertia resonant systems, bi-quad filters have been proposed [39, 42]. These are filters with a pair of low-damped poles and a pair of low damped zeros. They represent the inverse transfer function of the two-inertia system (see sec.s 3.4 and A.1). However, since the most important resonance of setup I occurs between encoder mount and drive, the two-inertia system is different from the usual case, and its transfer function has no zero that could form the bi-quad filter's pole. The most similar solution are notch filters.

Predictive filters are based on the assumption that the sampled data can be approximated as low-order polynomials. This assumption allows a prediction of the polynomial sequence to the future by a given time. A good summary is Vaeliviita et al. [27]. Predictive filters will be discussed in sec. 2.4; the analysis and results are mainly taken from Maletschek [4].

The main characteristics of predictive filters can be seen in fig. 2.4. It shows the transfer functions of predictive filters, in comparison to that of an ideal differentiator. At lower frequencies, the filter's group delay is negative, which means that the filter is predicting instead of delaying ("prediction band"). With increasing frequencies, there is a range where the filter gain is significantly above that of an ideal differentiator; the maximum is called the "gain peak" and is typical for all predictive filters. The delay in this region is positive and large, which may cause problems in the control loop. The range above is the "stop band", showing the filter's low-pass characteristic. At certain frequencies, the gain is very low; if the filter is properly designed, these gain minima can serve to damp the resonant frequencies.

2.2. Low-pass filters

Filters of 1st to 3rd order according to the Butterworth [16, butter] and Chebyscheff [16, cheby1] optimizations were tried (results see fig. 2.7, no. 1-5).

A low-pass filter's transfer function (in continuous-time domain) has 1 in its numerator and a Nth order polynomial in the denominator. The gain is 1 for low frequencies and shrinks continuously for frequencies beyond the cutoff frequency ω_c, which is the most important design parameter.

Low-pass filters of 1st order are uniquely specified by the desired cutoff frequency. For higher orders, additional degrees of freedom arise that can be used for filter optimization. A Butterworth filter is optimized to have a monotonously descending gain vs. frequency that is near to one below the cutoff frequency, but descends as fast as possible beyond it. The poles of a Butterworth filter lie on a half circle in the left half of the complex plane

$$p_{i,Butterw} = -\omega_c e^{j\frac{(2i+1)\pi}{2N}+\frac{\pi}{2}}, i = 0, 1, 2...N-1 \tag{2.1}$$

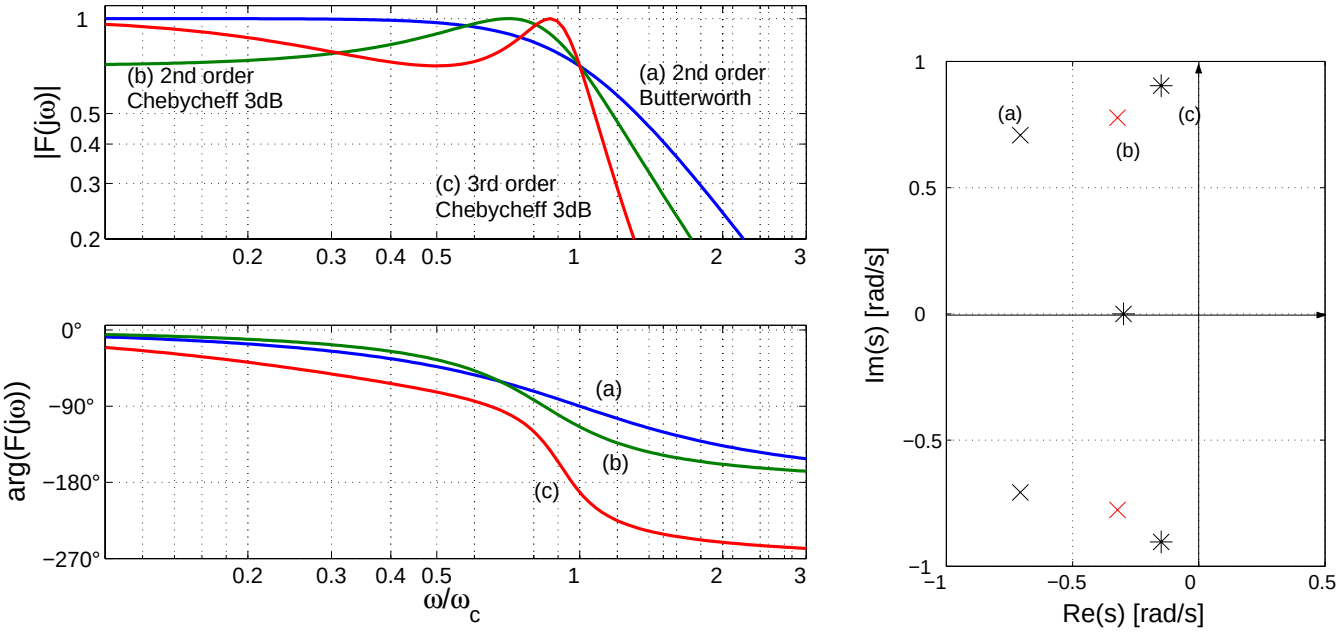


Figure 2.1.: Transfer function and pole distribution of low-pass filters

The transfer function magnitude and pole distribution of a 2nd order Butterworth filter with normalized cutoff frequency is shown in fig. 2.1(a).

Chebycheff filters allow the transfer function magnitude to fluctuate in the passband by a certain amount. At this expense, they achieve a steeper falling edge rate than Butterworth filters. The poles lie on a half ellipse depending on the allowable gain fluctuation R (in dB)

$$\begin{aligned} p_{i,Cheby} &= sinh(\mu)Re(p_{i,Butterw}) + jcosh(\mu)Im(p_{i,Butterw}) \\ \text{where } \mu &= \frac{1}{N}asinh(\frac{1}{\sqrt{10^{R/10}-1}}) \end{aligned} \tag{2.2}$$

Chebycheff filters with a gain fluctuation of 3 dB were tested. Their characteristics are shown in fig. 2.1(b) and (c).

For each filter, a delay time constant was calculated through approximation of the denominator polynomial by its constant and linear parts. The numerator was multiplied by s to create differentiating filters.

For implementation in a control algorithm, a time-discrete filter is needed. The conversion can be done using the bilinear transformation equation

$$s \approx \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{2.3}$$

If s is replaced in this way, the result is the filter transfer function in discrete-time

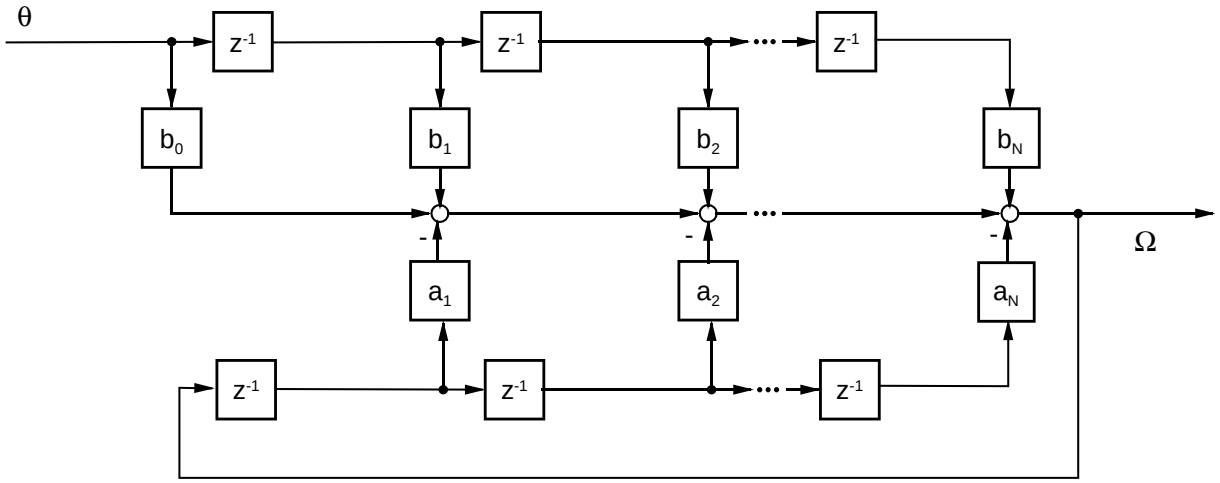


Figure 2.2.: Block diagram of a IIR filter

domain

$$F(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_N z^{-N}} \tag{2.4}$$

The denominator will be of the same order as the continuous-time domain transfer function; the numerator will be of the same order even if its order was lower in continuous-time domain. This filter can now be realized as shown in fig. 2.2; z^{-1} is a delay by one sampling period.

The experimental and simulation results with low-pass filters are shown in fig. 2.7 no. 1-5 and fig.2.8 no. 1.

2.3. Notch filters

Notch filters are designed in continuous-time domain by applying the lowpass-to-bandstop transformation to a 1st order lowpass filter. For design, the center frequency ω_c and the width of the stop band ω_d have to be specified. The resulting transfer function is

$$F(s) = \frac{s^2 + \omega_{c1}\omega_{c2}}{s^2 + s(\omega_{c2} - \omega_{c1}) + \omega_{c1}\omega_{c2}} \tag{2.5}$$

This filter transfer function is cascaded with a low-pass filter by multiplying the transfer functions. This allows computation of the delay time constant, and using the design process as described in sec. 2.2.

Fig. 2.3 shows the transfer function and pole/zero distribution of a pure notch filter

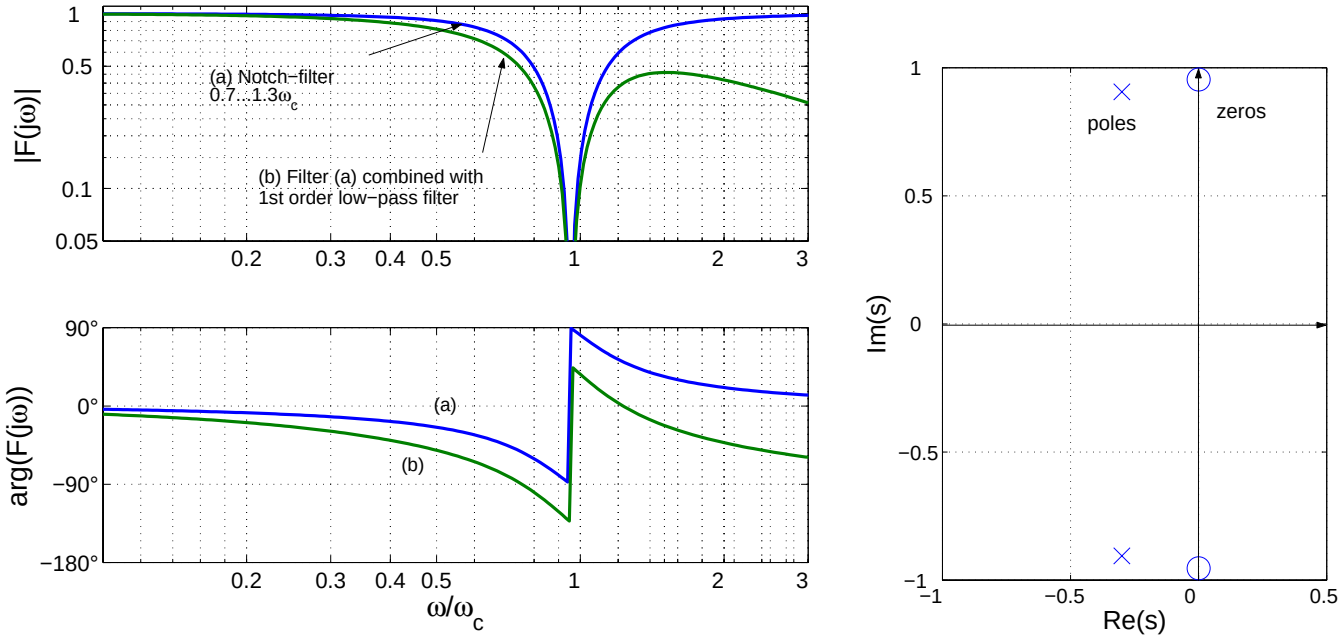


Figure 2.3.: Transfer function and pole distribution of notch filters

The time constant of the filter was chosen by varying the cutoff frequency of the low pass filter. The notch filter turned out to contribute only a little delay.

As an alternative, a simple FIR bandstop filter is proposed in [22]. It is located at the output of the speed controller, but can be moved to the feedback path with no changes to the load behavior which is investigated here. This filter superposes its input with a copy delayed by a number of time steps k. For the frequency ω_S fulfilling

$$\omega_S * k * T = \pi$$

(with sample time T), this means a phase shift of 180°, so it is damped out. This filter's transfer function is very similar to a notch filter except that the values of the transfer function continuously repeat after sampling frequency times filter length. However, its delay is much larger than that of a notch filter.

The best filter design in z-domain to damp out a 970 Hz oscillation at 10 kHz sampling rate is then

$$F(z) = \frac{1 + z^{-5}}{2} \tag{2.6}$$

with a delay time of $2.5 * T = 250\mu s$. This filter has minimal gain frequencies at 1000, 3000, ... Hz. It was chained with a 1st order low pass filter.

The experimental and simulation results with notch filters are shown in fig. 2.7 no. 6-8 and fig.2.8 no. 2-3.

2.4. Predictive filters

2.4.1. Heinonen-Neuvo FIR Predictors (HN-FIR)

Heinonen-Neuvo FIR predictor (HN-FIR) are the most basic version of predictive filters. They are derived assuming that the input signal is a Mth order polynomial in time with unknown coefficients. The design parameters are polynomial order M, filter order N, and prediction step p.

FIR filters only consist of a feedforward path. They are computed as shown in fig. 2.2, except that all a_i coefficients are zero. The equation for the b_i-coefficients of a non-differentiating HN-FIR filter is

$$\underbrace{\sum_{j=0}^{M} c_j p^j}_{\substack{\text{polynomial value}\\ \text{to be predicted}}} = \sum_{i=0}^{N} \left(b_i * \underbrace{\left(\sum_{j=0}^{M} c_j (-i)^j \right)}_{\substack{\text{polynomial value}\\ \text{i steps ago}\\ (0 = \text{recent})}} \right) \tag{2.7}$$

where the time unit is T, and 0 is the recent sample. c_j are the (unknown) polynomial coefficients. For a differentiating filter, the output should be the derivative at point p

$$\underbrace{\frac{1}{T} \sum_{j=1}^{M} j c_j p^{j-1}}_{\substack{\text{derivative value}\\ \text{to be predicted}}} = \sum_{i=0}^{N} \left(b_i * \underbrace{\left(\sum_{j=0}^{M} c_j (-i)^j \right)}_{} \right) \tag{2.8}$$

Since this should be true for every possible Mth order polynomial, the equation must be separated for the different powers

$$\frac{1}{T} j p^{j-1} = \sum_{i=0}^{N} \left(b_i * (-i)^j \right), j = 0, 1, 2, ...M \tag{2.9}$$

yielding M+1 equations for the N filter coefficients. This equation system is underdetermined if $N > M$.

The remaining degrees of freedom are used to minimize the noise gain NG, i. e. the gain for a white noise signal. The noise gain of a FIR filter is

$$NG = \sum_{i=0}^{N} (b_i)^2 \rightarrow \min \tag{2.10}$$

Table 2.1.: HN-Filter filter coefficients up to polynomial order 2

M = 0	$b_i = 0$
M = 1	$b_i = \frac{1}{T}\frac{6(N-2i)}{N(N+1)(N+2)}$
M = 2,p = 1	$b_i = \frac{1}{T}\frac{6(N^2-32iN^2+7N^2-64iN-13N+30i^2N+6i+60i^2}{(N-1)N(N+1)(N+2)(N+3)}$

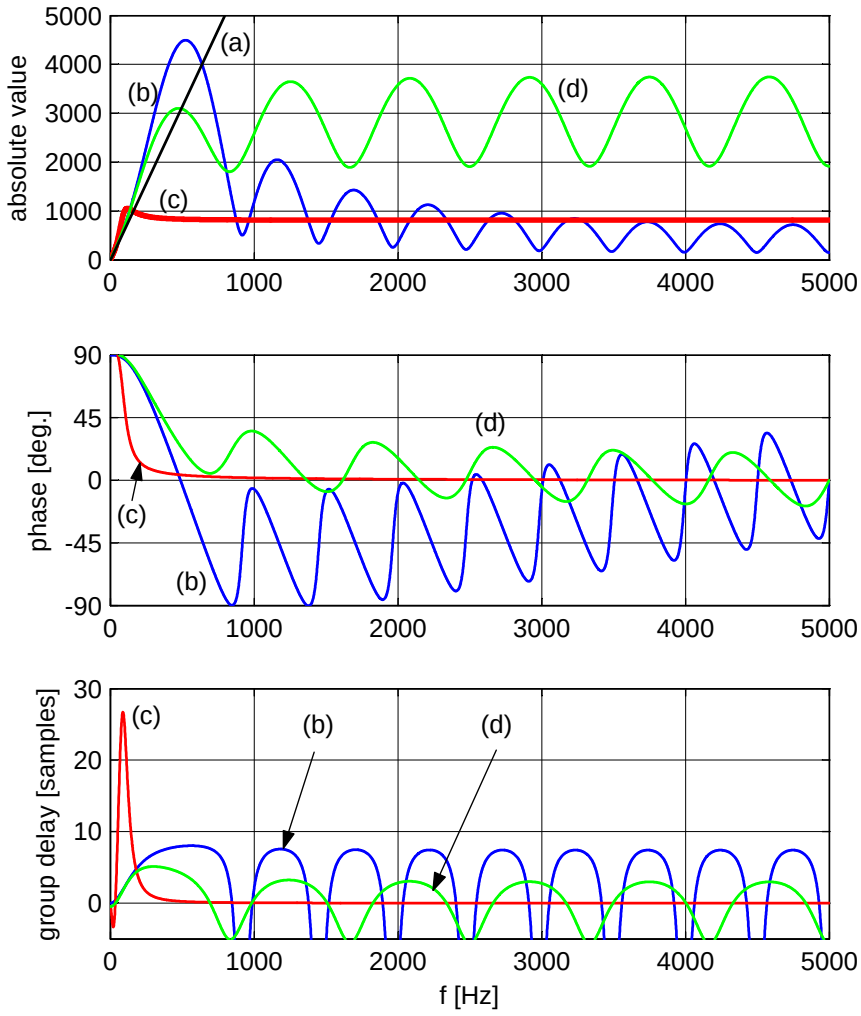
The optimization can be done using the method of Lagrange multipliers [26]. This is an algebraic method to compute the extreme values of a function of several variables subject to certain conditions. In this case, the target function is quadratic, and the conditions are M+1 linear equations for the b_i. Thus, the result is a unique minimum. The coefficient equations for one-step-ahead prediction differentiating filters are shown in table 2.1. A polynomial of order $M = 0$ is a constant, thus the derivative is zero. For $M = 1$, the derivative is constant; thus the filter is independent from p. M = 2 is the lowest order where the differentiating HN-FIR filter is in fact predictive.

The Bode plot of a HN-FIR filter that has been used in the setup (see fig. 2.9 no. 2)is shown in fig. 2.4. The most remarkable aspects are regular "notch" frequencies where the filter gain is minimal. Below the first notch frequency, the filter gain has a large maximum; this is called the "gain peak" and is typical for all predictive filters. The HN-FIR filter shown here is designed to be delay-free, but not predictive (p = 0!); thus the group delay for low frequencies is zero. With a predictive filter, there would be a small band after zero where the filter's group delay is negative; this is called the "prediction band".

In addition to the HN-FIR predictor's natural notch frequencies, [32] showed that it is possible to minimize the gain at arbitrary frequencies, e. g. to suppress resonant frequencies of the plant. This is convenient if the plant's resonant frequencies are very low compared to the sampling frequency, however, for the setup regarded here this was not necessary.

The design parameters of a HN-FIR filter are polynomial order M, filter order N, and prediction step p. When the polynomial order M is raised, the gain peak strongly increases. Raising N will lower the gain peak and the frequency where it appears and improve the damping in the stop band.

From a control designer's point of view, the main design criterion is the first minimum in the absolute gain vs. frequency graph, because it might be used like a notch filter to damp the first resonant frequency. The frequency where it occurs is the lower, the higher N is chosen; however, a closed expression for this "notch"



(a) ideal differentiator
(b) HN-FIR predictor M=2, N=19, p=0
(c) HN-IIR predictor M=2, N=2, p=0.5
(d) RLSN predictor M=1, N=12, p=0.5, a=0.175

Figure 2.4.: Transfer functions of the used predictive filters

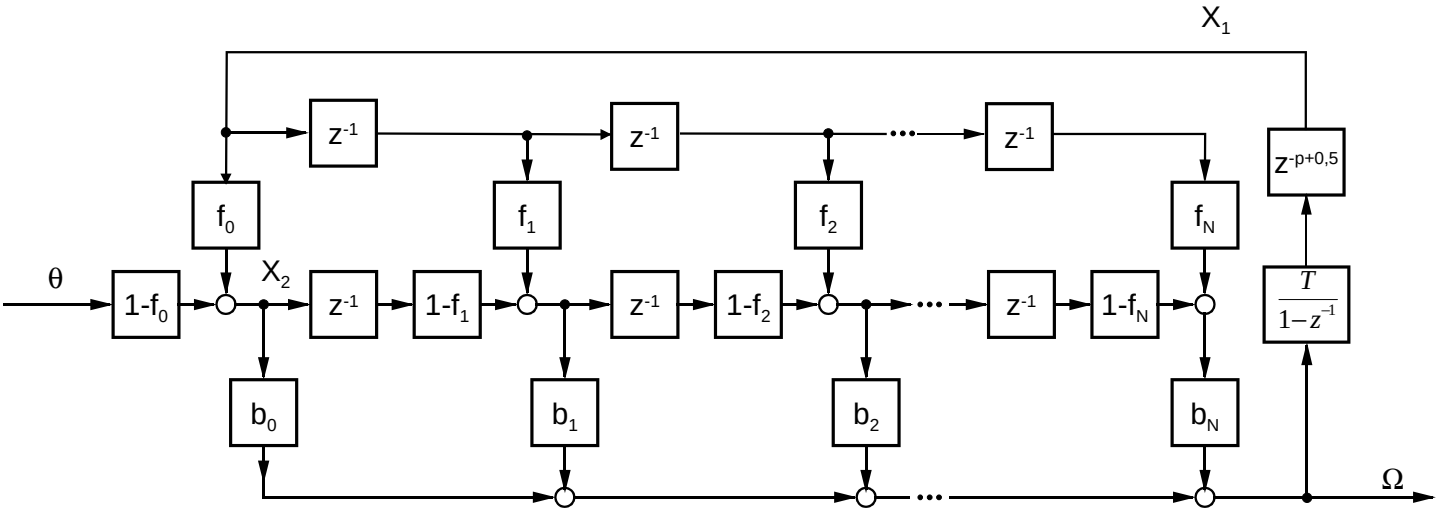


Figure 2.5.: Block diagram of the HN-IIR predictor

frequency cannot be given.

The results achieved with HN-FIR predictors are shown in fig. 2.9 no. 1-3.

2.4.2. Extended Heinonen-Neuvo IIR Predictors (HN-IIR)

A HN-FIR predictor has a large gain peak and only a moderate low-pass characteristic. To diminish these drawbacks, Ovaska et al. [28] proposed to augment it with a feedback path, forming an IIR filter.

The first step is always to design a HN-FIR filter as described in the last section. Then, an IIR path is added in a specific way using the coefficients f_i. A block diagram is shown in fig. 2.5; the specific structure ensures that the filter remains predictive. The signal X_1 is an output-based estimate of the current input signal; it is acquired by integrating and delaying the output[1]. The signal X_2 is the current input signal if $f_0 = 0$. When adding the IIR path, it becomes a weighted acverage of the current input signal and its estimate. The effect of the other feedback coefficients is analogous; this way, a feedback structure is achieved that keeps the predictive feature. The feedback coefficients are again optimized for three targets: a minimum noise gain, a minimum gain peak, and a penalty function if exceeding a pre-defined maximum pole radius. The latter is necessary to ensure stability. The cost function is then e. g.

$$cost = W * NG + (1 - W) * max(|F(j\omega)| - \omega) + max(|p_i| - R)^{50} \qquad (2.11)$$

with weighting factor W, and maximum pole radius R.

[1] The integrator $\frac{1}{1-z^{-1}}$ has a delay of 1/2 sampling time; thus, the additional delay is only $p - 0.5$. Since only whole powers of z can be realized, p must be an odd multiple of 0.5 for differentiating HN-IIR filters.

The optimization of HN-IIR filters is a complex task. Optimization using an iterative method such as the Simplex algorithm [16, fminsearch] is very problematic because the problem has many parameters and many local minima. Genetic algorithms should be used for optimization because they are always capable of finding the global minimum. This was not tested during this work for reasons of time; only the given optimal result in [28] was evaluated. The result is shown in fig. 2.9 no. 4.

All HN-IIR filters acquired during further optimization tests share the disadvantage of a huge maximum of group delay in the gain peak range. This is very problematic because it causes oscillations with that frequency in the closed loop. Thus, only a low control gain could be used. A possibility to solve this problem would be to account for the "delay peak" in the optimization target function.

2.4.3. Recursive least-squares Newton predictors (RLSN)

Recursive least-squares Newton predictors (RLSN) are derived from Newton's backward interpolation algorithm. This algorithm yields an interpolation polynomial of arbitrary order that is valid between the two last samples; the anterior samples are approximated, but not interpolated. This polynomial can also be used for prediction up to one step ahead. Because the frequency characteristic of a pure Newton predictor is very problematic, the RLSN predictor proposed by Ovaska and Vainio [30, 31] contains several improvements.

Newton predictors are based on Newton's backward interpolation algorithm. Newton's interpolation algorithm relies on the difference operator Δ, which is defined as

$$\begin{aligned} \Delta x(k) &:= x(k) - x(k-1) \quad \circ\!\!-\!\!\bullet \quad (1 - z^{-1})X \\ \Delta^2 x(k) &:= \Delta x(k) - \Delta x(k-1) \quad \circ\!\!-\!\!\bullet \quad (1 - z^{-1})^2 X \\ &\dots \end{aligned} \tag{2.12}$$

The equation for one-step-ahead prediction, in the case of equally spaced sampled data, is then

$$\hat{x}(k+1) = \sum_{i=0}^{M} \Delta^i x(k) \tag{2.13}$$

The polynomial used for prediction is always of the same order as the number of past samples used. In comparison to the interpolation polynomial, the Newton polynomial relies more on the recent sampling points than on the past ones. The block diagram of a Newton predictor is shown in fig. 2.6 as part (a) (including the summation at the left-hand side).

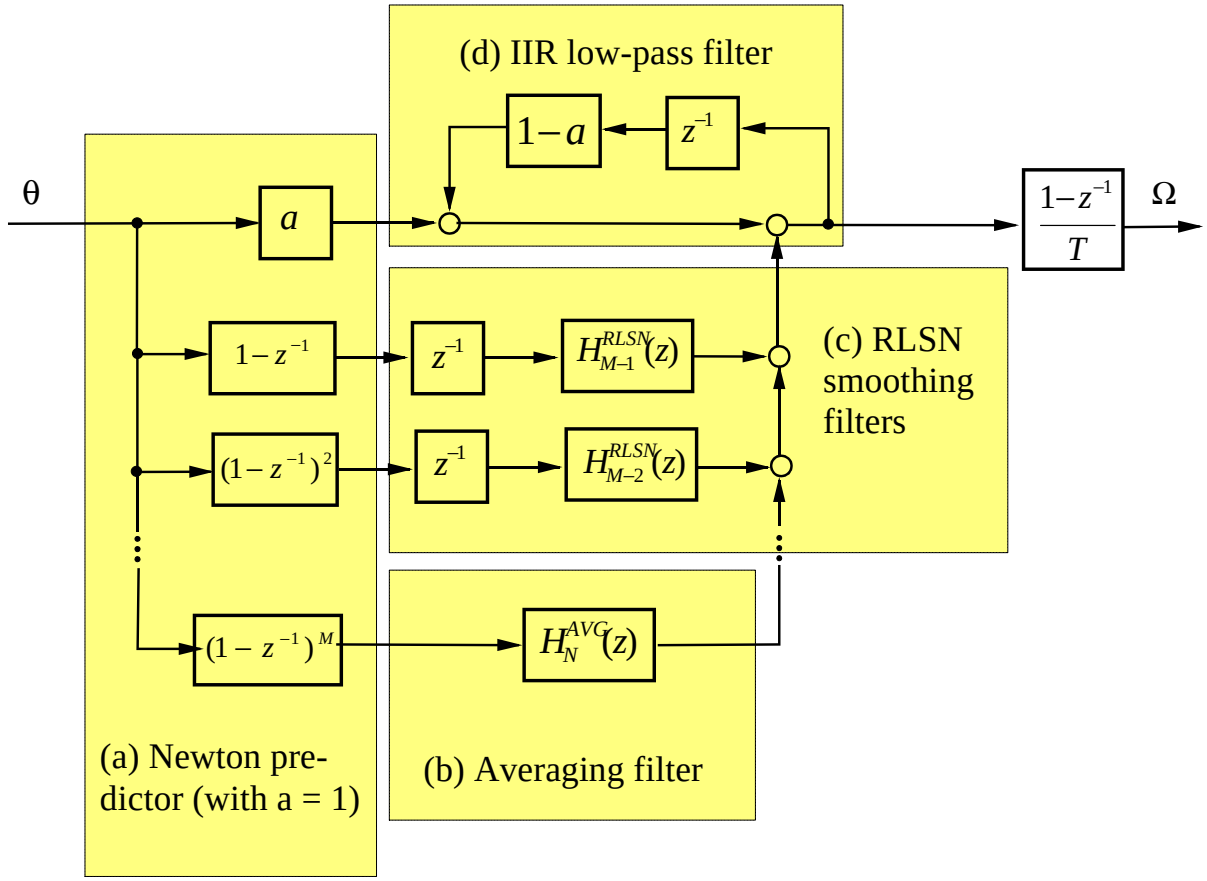


Figure 2.6.: Block diagram of the RLSN predictor

The Δ^n coefficients, written as $(1-z^{-1})^n$ in the block diagram, represent the nth derivative of the polynomial. Thus, with the assumption of a Nth order polynomial, $\Delta^N x$ is constant. A low-pass filter is inserted into this path to reduce noise amplification; a delay in this path is not problematic. A length N moving averager

$$H_N^{AVG}(z) = \frac{1}{N}\sum_{i=0}^{N-1} z^{-i} \tag{2.14}$$

is used (part (b) in fig. 2.6).

The lower derivatives should also be filtered, however, a delay is critical here. Thus, RLSN predictors $H_{M=?}^{RLSN}$ of lower order are used, together with a delay to compensate the prediction (fig. 2.6 part (c)). This combination features a low-pass characteristic and zero delay. The filter definition becomes recursive, however, it is still easy to handle up to $M = 2$. The direct path from input to output is augmented to a 1st order IIR low-pass filter with feedback coefficient a; the feedback filters the complete output signal (part (d)).

The design parameters of a RLSN predictor are polynomial order M, moving averager length N, and feedback coefficient a. The filter order is $N+M+2$, where all but $2M+4$ numerator coefficients are zero. In spite of the complex derivation, the equations of RLSN predictors are quite simple. A $M = 1$ predictor is

$$H_{M=1}^{RLSN}(z) = \frac{1-z^{-N}}{N\left(1-\frac{1}{z}\right)} \tag{2.15}$$

The amplitude versus frequency ω [rad/s], with removed constant terms and roots, is [31]

$$\frac{1-\cos(N\,\omega\,T)}{1-\cos(\omega\,T)} \tag{2.16}$$

This indicates that the "notch" frequencies are approximately at

$$\omega_{notch} = i * \frac{2\pi}{NT}, i = 1, 2, 3, \ldots \tag{2.17}$$

and the gain peak is approximately at

$$\omega_{peak} = \frac{\pi}{NT} \tag{2.18}$$

The predictor for $M = 2$ has the transfer function [31]

$$H_{M=2}^{RLSN}(z) = \frac{(1+a\,N)+(a^2\,N+a-1)z^{-1}-(a\,N+a)z^{-2}-z^{-N}+(1-a)z^{-(N+1)}+az^{-(N+2)}}{(1-(1-a)z^{-1})^2} \tag{2.19}$$

(2.17) and (2.17) can be derived in the same way if a is set to zero and the trigonometric functions of ωT are approximated as

$$\begin{aligned} cos(\omega T) &\approx 1 \\ sin(\omega T) &\approx \omega T \end{aligned}$$

Raising a does not move the frequencies significantly.

To make the RLSN predictor a differentiating one, it is simply cascaded with a FIR discrete differentiator (1.16), reducing the prediction step to p = 1/2.

The results achieved with RLSN predictors are shown in fig. 2.9 no. 5-7.

2.5. Experimental and Simulation Results

Fig. 2.7 shows the results for classic filters measured at setup I with the 2048-line encoder. All controls were tuned to achieve a r.m.s speed error of 0.1 rad/s at steady state. The notch filters' bandwidth was chosen so that its lower border equals the cutoff frequency of the low-pass filter.

The speed loop gain that resulted from tuning is shown in the first bar for each filter. The second bar shows the dynamic stiffness that was measured in a subsequent experiment using several step-wise load changes between +50% and -50% of the drive's rated torque (4.3 Nm). The position deviation $\Delta\theta_{rat}$ in this experiment can be seen relating the dynamic stiffness bar to the axes below the graph.

It can be seen that all low pass filters (no. 1-5) lead to similar results. Performance shrinks with increasing filter order, thus filters of higher order than 3 were not investigated. Obviously for the used setup, a steeper stop band behavior at the expense of a higher delay time constant does not pay, because the resonance frequencies allow a high control loop bandwidth even when lightly damped. However, the control loop without any filter was found to be unusable.

The two Chebyscheff filters' behavior is due to the imperfection of the time constant approximation used in this thesis. For a 2nd order Chebyscheff filter, the time constant is considerably lower than the Butterworth filter's, while for a 3rd order Chebyscheff filter it is considerably larger. It was not investigated here how exactly Chebyscheff filters with their low damping affect the closed loop stability, which is design goal of the symmetrical optimum. Please note also that both filters did not show the dynamic stiffness that would have been expected from the controller gain, if the ratio is compared to other filters. Additionally, in the load step experiments, both Chebyscheff filter controls show an overshoot of position, which is very problematic in most practical cases.

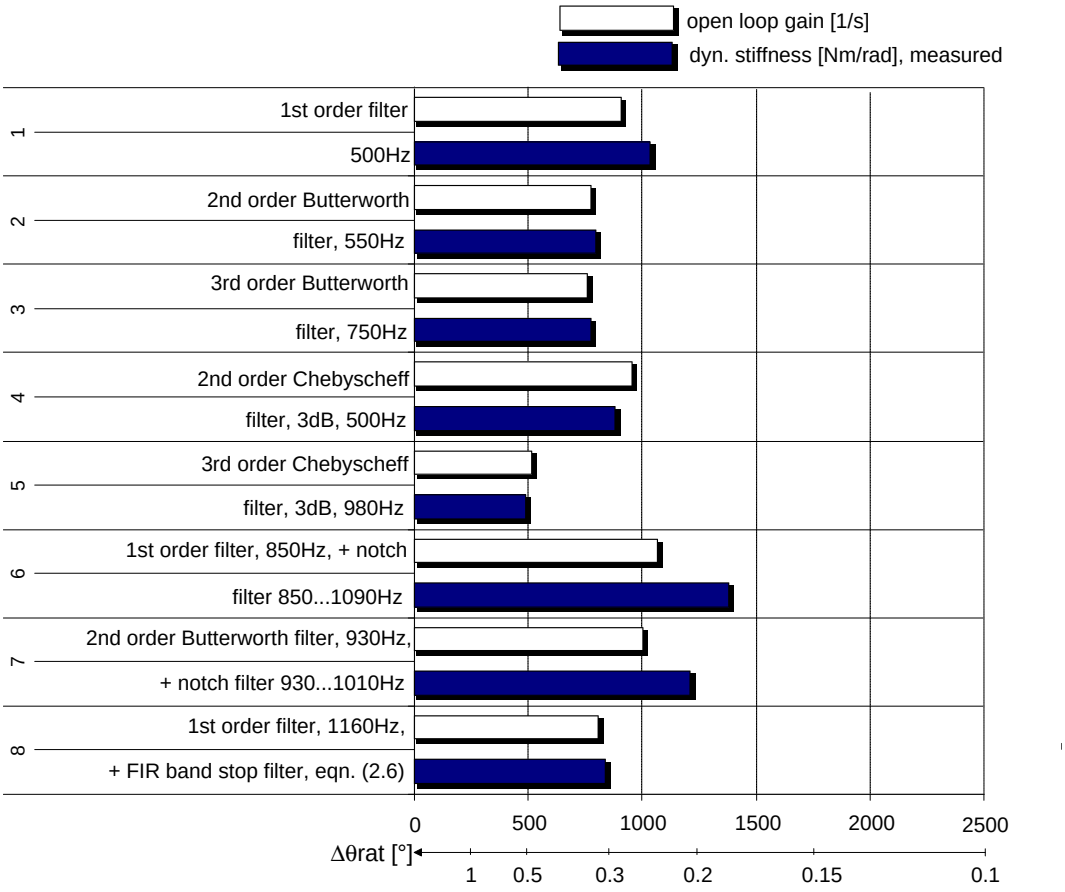


Figure 2.7.: Experimental results with 2048-line encoder

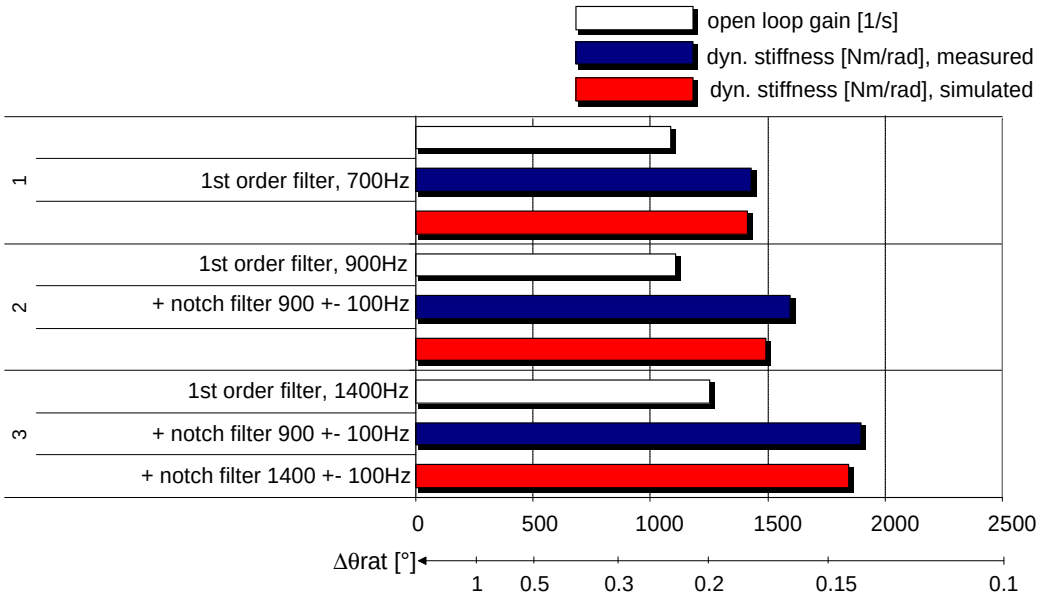


Figure 2.8.: Experimental and simulation results with 5000-line encoder

The low pass plus notch filters (no. 6,7) perform significantly better than simple low pass filters. The reason is that the notch filter achieves a much better suppression of the 970 Hz resonant frequency while contributing only few to the delay time constant.

The FIR band stop filter according to (2.6) (no. 8) shows a stop band behavior comparable to a IIR notch filter, however its delay time contribution is much higher. This explains the poor performance of this filter in the experiments.

Figure 2.8 shows measurements using some of the filters with a 5000-line encoder. The control loops were tuned to a steady state speed error of 0.07 rad/s. The additional bar shows the result of simulation of the load step experiment (with the same control structure and parameters); see section 1.6 for a discussion.

Again, the diagram shows that the performance using a 1st order filter -which is already quite good- can be improved using notch filters. Because of the higher control loop bandwidth, a double notch filter could improve performance, which was not the case with the 2048-line encoder. A comparison with fig. 2.7 indicates that the filter controls profit from the better position signal - though the speed error limit has been reduced. This is due to the fact that the resonant frequencies are excited by speed signal noise times filter gain at the respective frequency times controller gain. If the speed noise is reduced, the filter gain a the resonant frequency may be higher without exciting the plant above its passive damping capability (see section 1.4).

Fig. 2.9 shows the results achieved with predictive filters. Since the filters take into account the setup's 1st mechanically resonant frequency, they should be

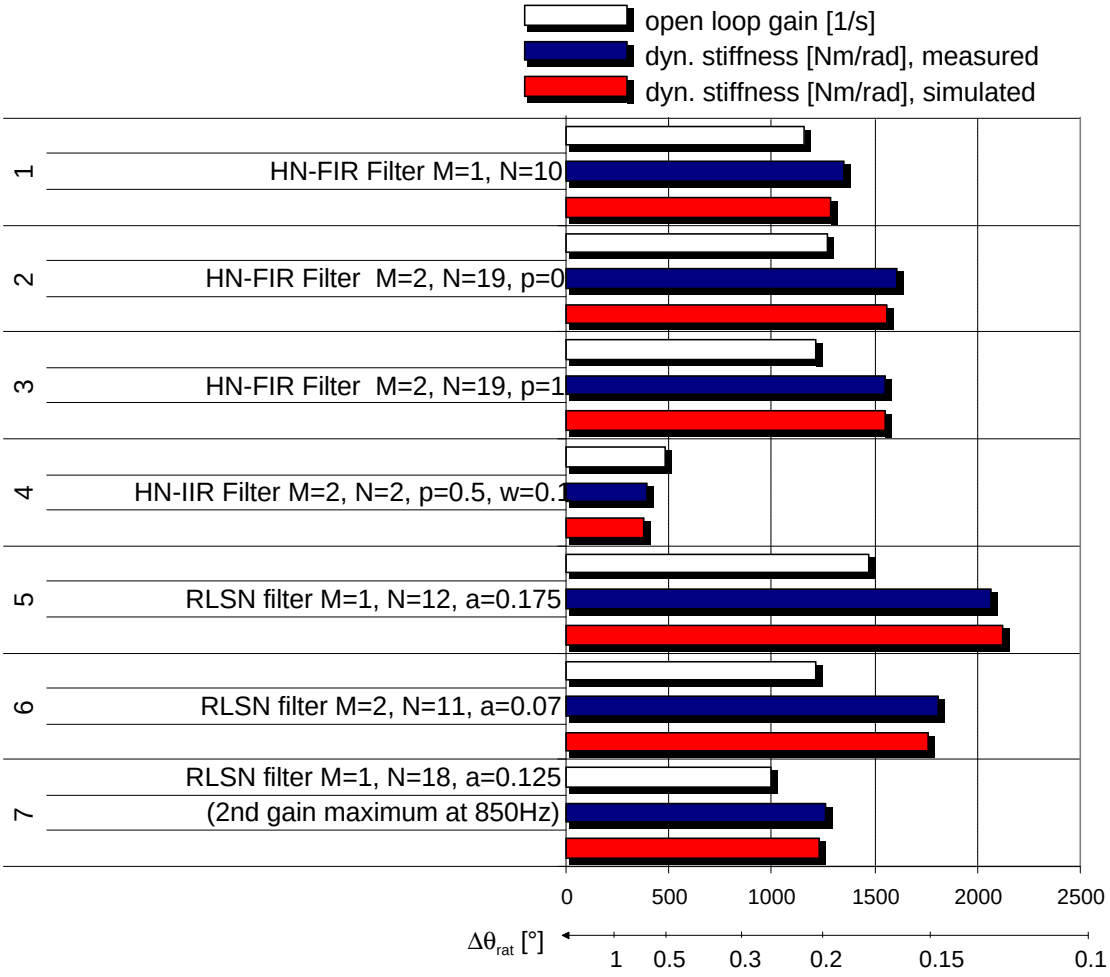


Figure 2.9.: Experimental and Simulation results for predictive filters with 5000-line encoder

compared to the results achieved with low-pass + single notch combination.

The HN-FIR predictors achieve results in an equal range. A 2nd oder polynomial should be used, however the filter performs better if it is not predictive (p = 0). This is still a great advantage compared to the standard low-pass filter with its large delay.

The HN-IIR filter allows only a weak control gain due to the "delay peak" problem discussed in sec. 2.4.2. It would be necessary to reduce the delay in the gain peak range to achieve better results. A possible approach would be to punish the maximum delay in the optimization target function.

The RLSN filter provides best performance with a 1st order polynomial. Its performance is better than the best conventional filter combination; this type of predictive filters can be recommended. If the resonant frequency is not exactly known, the filter may -by bad chance- be designed so that its second gain maximum appears at the resonant frequency. This case has been tested in fig. 2.9 no. 7. The performance is lower than with a low-pass filter, but still in a sensible range. Care must only be taken that the (first) gain peak stays below the critical frequency range; this is a design criterion similar to a low-pass filter. These facts suggest that RLSN predictive filters are also possible to handle in industrial applications.

3. Speed estimation using observers

3.1. Introduction

Chapters 2 and 3 deal with the speed computation method, which is block (2) in fig. 1.4.

A second possibility for speed acquisition is the use of observers (see section A.5). An observer is a mathematical model of the controlled system, which is calculated in real-time. To compensate modeling inaccuracies, it is a well synchronized with the real system through feedback of the difference between measured values and their modeled counterpart.

Observers with a rigid-body system model observers are tested in [24] and [25] in comparison to the filtered derivative. Both theses state that observers lead to a better uniform run. In both cases, experimental setups of linear motors without load machines are regarded, so that data for the disturbance rejection are missing and mechanical resonances should be negligible. Besides, [24] uses only an incremental position encoder (without interpolation from sinusoidal signals), causing significant quantization noise. The good performance of the rigid-body observer for incremental encoder signal processing is confirmed in other publications [40].

An observer requires two design steps: to find the model and to design the feedback matrix. Different possible models are discussed in the following sections.

Concerning the feedback, two main design procedures exist: Pole placing and the static Kalman filter design. Pole placing means to specify where the poles of the closed-loop observer system should be. This results in an unique feedback vector if the system is a single-output one, meaning that only one quantity is measured. If not, additional design criteria have to be specified. A usual pole placement strategy is to place all poles to one negative real location, so that the resulting system has no overshoot and a high damping. Another possibility is to place the poles like those of a Butterworth filter. This is intended to produce an optimal low pass behavior in frequency domain.

A more advanced method is an observer modeling the mechanical resonance as a two- or three-inertia system. Nearly all publications on this subject are limited to two-inertia models, pointing out that higher order systems can be approximated this way [33]. The frequencies to be actively damped are usually significantly lower than the lowest resonance of the setup investigated here, which is 970 Hz. As a

consequence, the encoder is assumed to be stiffly mounted, while its oscillating behavior is considered in this thesis.

Where observers for multi inertia systems are used, they are often designed by placing all poles to one location in the s-plane [33] or as a Kalman filter [34]. [35] uses the poles of the state control loop, left-shifted by a constant distance.

In [36, 37], a setup regarded as a three inertia system with eigenfrequencies of 400 and 860 Hz is investigated. An observer for speed acquisition is designed, and it is shown that good quality of the speed signal allows active damping even through the standard PI controller.

A comparison of filter and observer speed acquisition methods has been published in [1].

3.2. Observer based on rigid-body model

The simplest model for a controlled servo system is the rigid-body model, which is shown in fig. 3.1 (with plant inertia J, torque constant k_T, load torque t_L, reference current in the quadrature axis i_q^*, speed Ω, encoder angle θ, and feedback constants $k_{1,2,3}$). All system states and model parameters are estimated values and therefore marked with a hat. The reference current is used here instead of the actual value because actual current is not known to the control PC, and current control is considered ideal. To observe the load torque, an integrator is used which gets its input only from the feedback signal. Load and motor torque are integrated two times to get the estimated position angle, which is compared to the sensor angle to compute the observer feedback.

The transfer function of the rigid-body system observer is

$$\hat{\Omega} = \frac{\left(\left(\frac{k_T}{J}\left(K_1 s + s^2\right)\right) i_q^* + \left(K_3 s - J K_2 s^2\right)\theta\right.}{-\left(\frac{K_3}{J}\right) + K_2 s + K_1 s^2 + s^3} \tag{3.1}$$

Fig. 3.1 shows the continuous-time model for convenience. However, for implementation in the control PC, the corresponding time discrete observer has to be used. This makes it possible to use the predicted values of one step ahead (see appendix A.6), which improved the observer's performance a little in practice.

For observer design, two methods for pole placing and the Kalman filter were considered. As to pole placing, the desired poles were specified in s-domain, then converted to z-domain to calculate the feedback values. The Kalman filter design algorithm works directly in z-domain.

The first observer (fig. 3.6 no. 1) was designed by placing all three poles to one negative real location. This method is recommended in several publications, such as [24] and [25]. The delay time constant was chosen by changing this value.

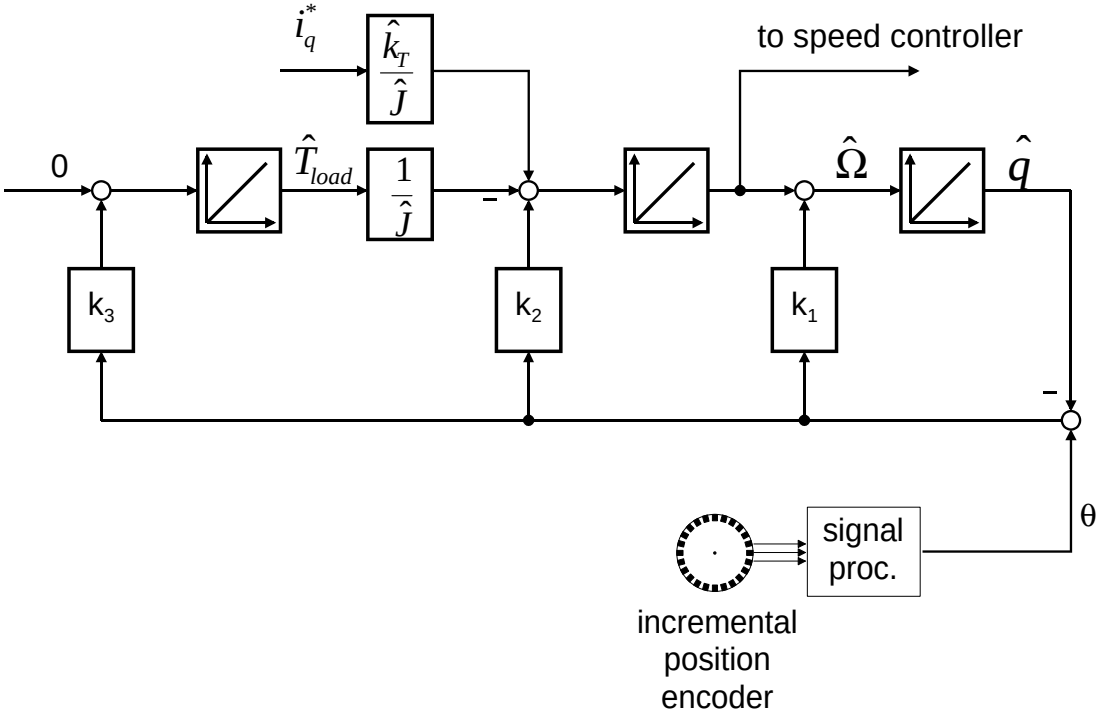


Figure 3.1.: Block diagram of the rigid-body observer

A second way is to place the observer's eigenvalues to those of a 3rd order Butterworth filter (fig. 3.6 no. 2). This is also called the damping optimum [9]. The goal of this placement is to achieve an optimum attenuation for disturbances beyond the filter's cutoff frequency. The time constant is altered by changing the cutoff frequency.

Finally, the static Kalman filter design method was used (fig. 3.6 no. 3). This method is based on the idea of a state space system that is disturbed by white noise added to the system equations and sensor outputs. The covariance matrices of system noise $\mathbf{Q}$ and sensor noise $\mathbf{R}$ are specified, and the feedback coefficients are chosen for a statistically optimal estimation. Usually, only the main diagonal of $\mathbf{Q}$ is set to values which stand for the amount of all kinds of disturbances the designer expects for the respective equation. $\mathbf{R}$ is a scalar for single-sensor systems. As a first approach, only the load torque was considered disturbed. If the state vector $\underline{x}$ is defined as

$$\underline{x}^T = (\theta \ , \ \Omega \ , \ T_{load})$$

this means that only $\mathbf{R}$ and the matrix element $\mathbf{q}_{33}$ are given a value different from zero. As the resulting filter does not change if the $\mathbf{Q}$ and $\mathbf{R}$ matrices are scaled equally, there is only one degree of freedom left. This was used to adjust the observer's time constant: The lower $\mathbf{R}$ and the larger $\mathbf{q}_{33}$, the faster the observer.

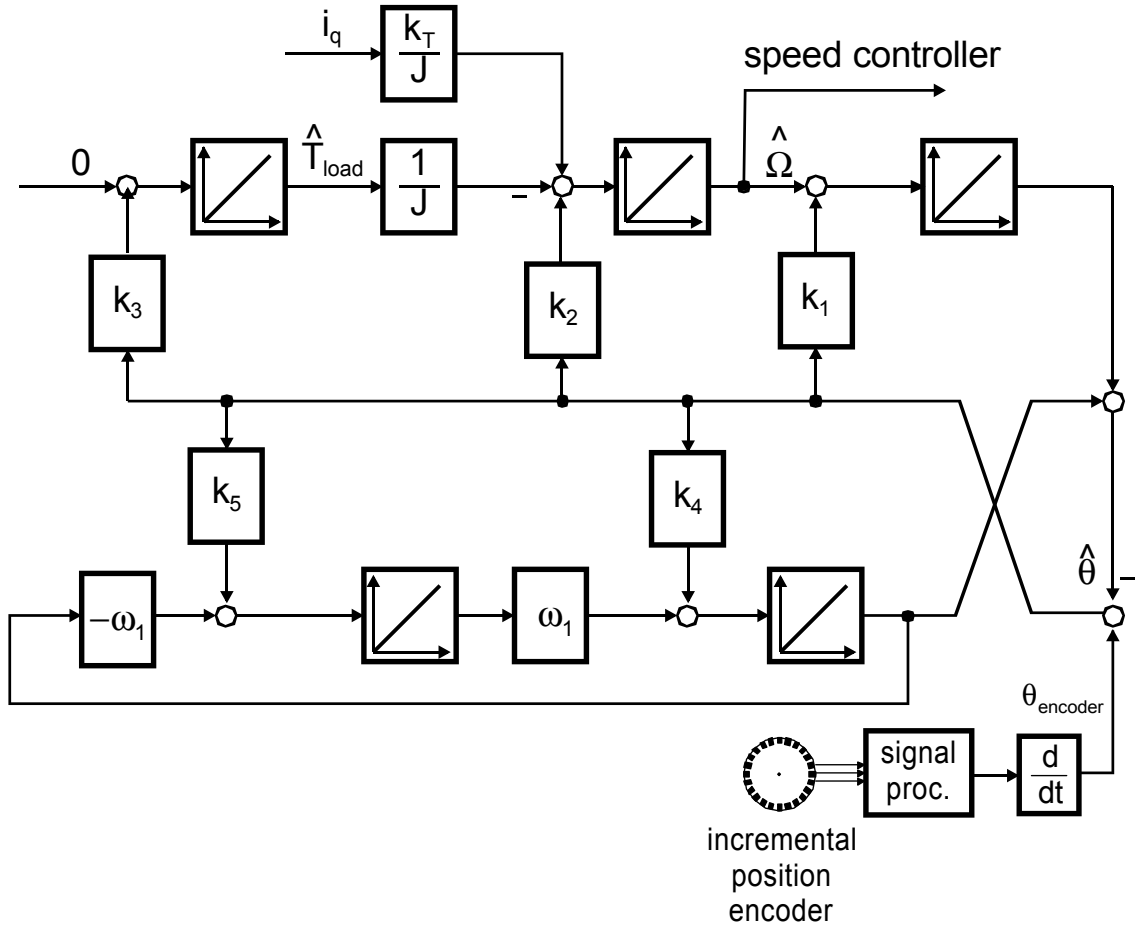


Figure 3.2.: Block diagram of the observer including an abstract oscillation model

The results achieved using rigid-body observers are shown in fig. 3.6 no. 1-3.

3.3. Observers including an abstract oscillation model

As a second step, the plant's mechanical resonance can be abstractly modeled without regarding its physical origin. This kind of observer is shown in fig. 3.2. Of course, again all model parameters are estimated values and should therefore be marked with a hat; this is omitted in the following to keep the diagrams and equations as simple as possible.

It is assumed that a source of oscillation exists independently from the other plant model. This oscillating signal is part of the estimated encoder angle, but is not fed to the controller. This kind of observer model has an effect much like a notch filter, as can be seen in fig. 3.7.

The transfer function of the rigid-body system observer with abstract oscillation model is

$$\hat{\Omega} = \frac{\left((K_1+K_4)\,s^3 + s^4 + K_1{\omega_1}^2 s + (K_5\omega_1 + {\omega_1}^2)\,s^2\right) k_T i_q^* + \left(-K_3 s^3 + JK_2 s^4 - K_3{\omega_1}^2 s + JK_2{\omega_1}^2 s^2\right)\theta}{(JK_1+JK_4)\,s^4 + Js^5 - K_3{\omega_1}^2 + JK_2 s{\omega_1}^2 + s^3\,(JK_2 + JK_5\omega_1 + J{\omega_1}^2) + s^2\,(-K_3 + JK_1{\omega_1}^2)} \tag{3.2}$$

For feedback design, the same choices exist as stated in section 3.2. The method of pole placement like a Butterworth filter led to good results. Pole placement to one negative real location works except that two out of five poles must be set to a weakly damped pole pair near the oscillation frequency. Otherwise, the strongly damping feedback will prevent the oscillator model from oscillating.

The results achieved using rigid-body observers are shown in fig. 3.6 no. 4-6.

Of course the other possibility also exists: the controller can be fed the plant model speed plus the oscillating one. This did not work at all with the setup regarded here. The reason is shown in section 3.4: a two-inertia model with motor and sensor mass oscillating against each other is a quite good model for the setup. If the speed controller output contains a signal at oscillation frequency with a phase phase opposite to the sensor speed, it will excite the oscillation instead of damping it.

3.4. Observers based on two- or three inertia resonant system models

3.4.1. Model Identification

The two inertia model is shown in fig. 3.3 (a). As opposed to most publications, the resonance of the encoder mount is not negligible. It is even more important than the resonance between servo and load (see fig. 3.3 (b)), since it causes a phase shift between sensor and actuator of the plant. Thus, as a first approach, this resonance is modeled.

The structure of the three inertia model is shown in fig. 3.4. The two inertia model is constructed by leaving out the paths containing c_{12} and J_2. It consists of the two inertias (J_0: encoder mount, J_1: machines), the elastic coupling with spring constant c_{01}, the integrator for modeling the encoder angle and the integrator estimating the load torque. The load torque was modeled as affecting both inertias equally. This is not physically correct, but decouples the load torque observer from system oscillations, leading to better observer performance. The proportional part of the PI speed controller was implied in the system model as if it was part of the system; this path is depicted as a dashed line in fig. 3.4. The

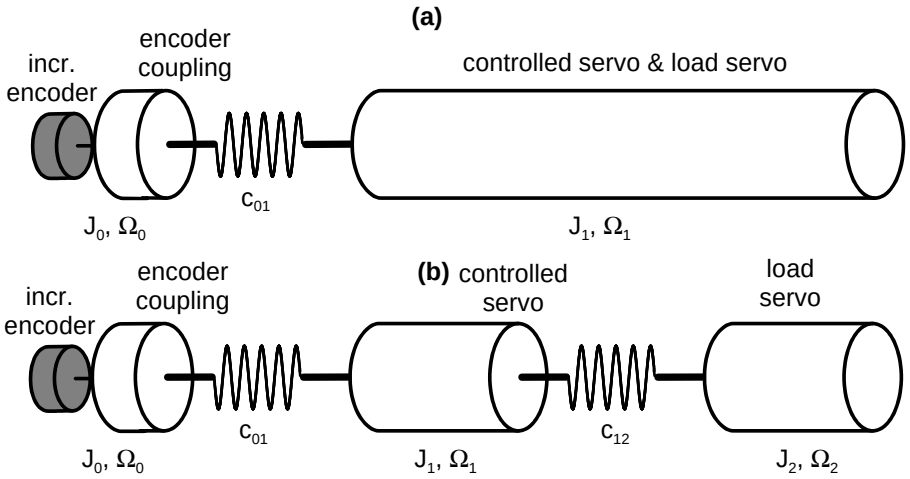


Figure 3.3.: (a) Two- and (b) three-inertia model for setup I

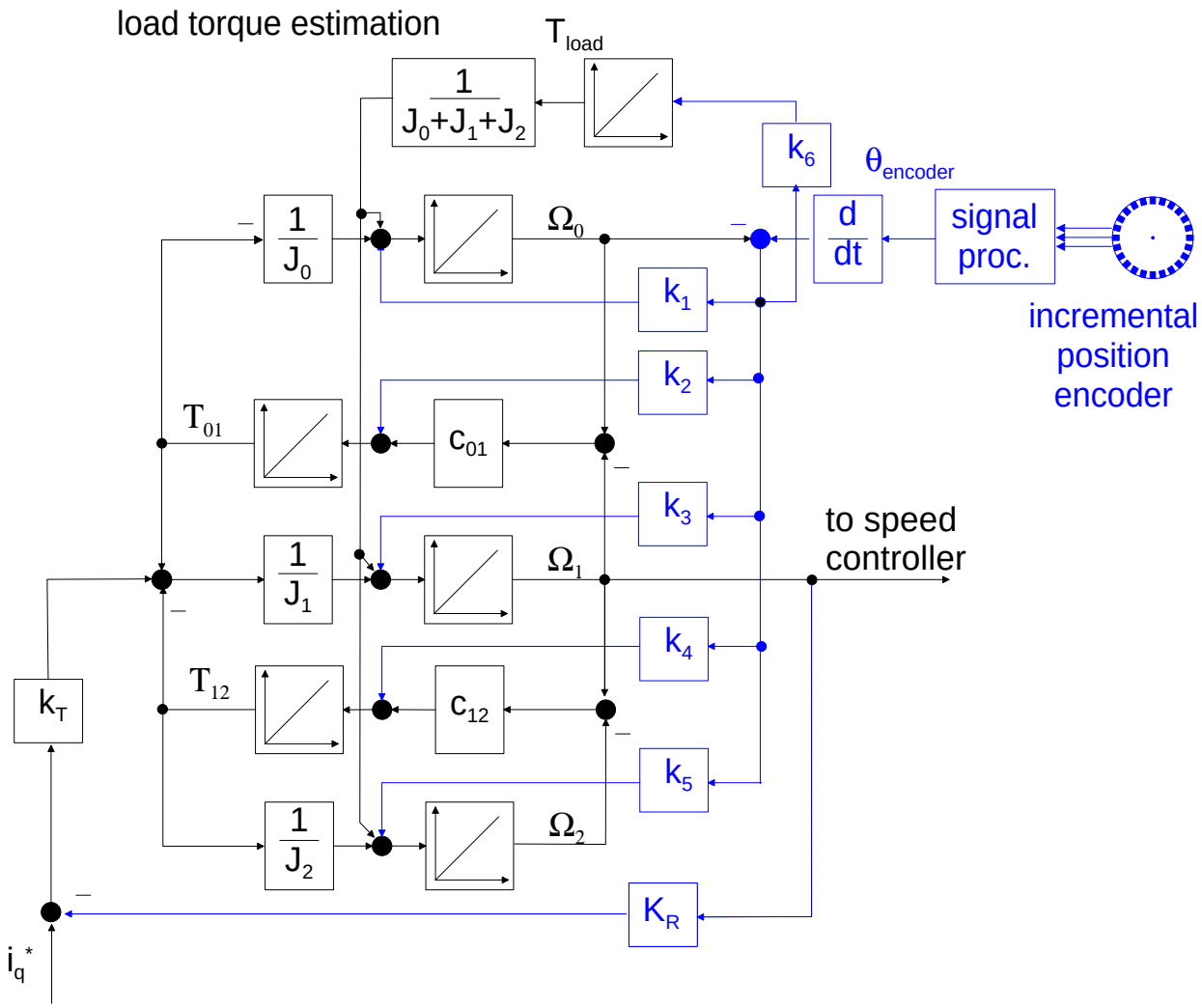


Figure 3.4.: Block diagram of the three-inertia system speed-only observer

parameters of the two inertia system are acquired by a procedure in three steps, which is described in [41]:

The plant is excited by a noise signal with uniform (or at least a known) frequency distribution. The best choice is a multi-sine signal [41]: In a data array, multiple sine functions with constant amplitude and random, all of them periodic with array length, are summed up. Then, array data are scaled so that the maximum absolute value is 1. Better results were achieved if the motor is turning at slow speed during identification, because this diminishes the effect of slot latching; thus, the speed control loop was used at the same time, with its bandwidth limited to 10 Hz.

The encoder position signal is recorded during excitation and derived once or twice by difference quotients to achieve speed or acceleration, respectively. Double differentiation by the difference quotient

$$\alpha(k) = \frac{\theta(k-1) + \theta(k+1) - 2 * \theta(k)}{T^2} \tag{3.3}$$

is recommended because it is possible without phase shift (off-line only).

The dominant poles and zeros are identified. There are two main possibilities to do this [41]: One method is to identify them graphically in a plot of the Fourier transform of the transfer function. Since both poles and zeros are only very lowly damped, they appear as clear peaks and minima of the transfer function vs. frequency. Another possibility is to use the ARMAX identification method [12]. It is important to choose the system order as high as the real system order. This was about 8..12 for setup I. Otherwise, the non-dominant oscillation modes will appear as systematic disturbances in the identification procedure, making the result unusable. The graphical method has the advantage of being easy, especially when done by hand; it is also possible to automate it [41].

The model parameters can be computed symbolically. If the state-space model is given, the poles and zeros of the transfer function can be computed using eqn. (A.5). This equation can be solved algebraically up to the three mass system using e. g. Mathematica [17]. Matching the equations for poles and zeros in terms of the model parameters with the identified numerical values yields a set of equations for the model parameters.

The total inertia could be determined from the plant amplification identified by the ARMAX method. However, it was found better to use another experiment, e.g. constant acceleration. This test is also necessary for correct design of the speed and position controllers.

A two-inertia resonant system with sensor and actuator acting on different loads has one (double) pole and no zeros, as opposed to the usual case where the load-to-motor resonance is regarded. Thus, there are only two equations (total

Table 3.1.: Identified parameters of setup I

	J_0 $kgcm^2$	J_1 $kgcm^2$	J_2 $kgcm^2$	c_0 Nm/rad	c_1 Nm/rad	pole(s) rad/s	zero(s) rad/s
2-inertia system, 2048-line encoder	2.7	17.35		8679		$\pm j2\pi970$	
3-inertia system, 2048-line encoder	2.09	8.7	9.26	10376	22249	$\pm j2\pi970$, $\pm j2\pi1370$	$\pm j2\pi780$
2-inertia system, 5000-line encoder	2.8	17.25		7214		870	
3-inertia system, 5000-line encoder	3.1	8.4e-4	8.55	11100	19600	$\pm j2\pi872$, $\pm j2\pi1286$	$\pm j2\pi762$
Simulation model (3-inertia system, 5000-line encoder)	2.07	12.71	5.27	11156 $d_{01} = 0.045$	12107 $d_{12} = 0.14$	$\pm j2\pi870$, $\pm j2\pi1285$ $\delta = 1.0°, 1.7°$	$\pm j2\pi762$ $\delta = 1.6°$

inertia and pole frequency) for determining three parameters. Computation is easiest when the encoder fixing inertia is computed from geometrical data first. This is no problem in practice since the encoder fixing's geometry is known at design time. The other parameters result from the equations.

The three-inertia system with actuator and sensor fixed on different inertias has two pole and one zero locations, so there is again one equation missing. Inserting total inertia, two (double) poles and one (double) zero into the symbolic equations, it is easiest to solve the parameters J_0, J_2, c_0, and c_1 for J_1. Then, in a graphical plot of J_0 and J_2 vs. J_1, the solution can be chosen which is closest to geometrical data.

Table 3.1 shows the pole and zero locations and physical parameters of the used plant models. Using the 2048-line encoder, the response signal shows an increased noise level between about the two pole frequencies. It is caused by the encoder itself, which is mounted to the servo housing only by a weak elastic steel plate. According to Huebner company, this fixing has been improved meanwhile. The acceleration sensor was originally using an aluminium disc, which has been replaced by brass to minimize the sensor's delay (see section 6.1). Because brass is heavier than aluminium, the resonance frequencies are lower with the 5000-line-encoder/ brass disc combination.

The observer models were designed neglecting the setup's natural damping. However, the plant model used in the simulation contained damping. The parameters were identified by matching the Bode magnitude plot to the bode plot of the plant's response to excitation. Starting from zero damping, the Simplex algorithm

[16, fminsearch] was used to minimize the quadratic error between theoretical and experimental data in the range of 50..2000 Hz. The identified values are shown in table 3.1, in the sense of fig. 1.7. δ means the angle between pole or zero location and the imaginary axis. An alternative would be to assume the natural damping of steel, which is given in [13, S. O12] as about $\delta = 0.057°$. This is much too low; the difference is due to bearing friction and magnetic losses which also contribute to the system damping.

3.4.2. Observer structure and feedback design

Based on this model, the first step is to design an observer similar to [36]: The encoder angle is derived and compared to the modeled encoder speed $\hat{\Omega}_0$, the difference is used for feedback; this structure will be called the speed-only observer (results see fig. 3.8 no. 1,2,5,6 and fig. 3.9). It is shown in fig. 3.4. Of course, again all model parameters are estimated values and should therefore be marked with a hat; this is omitted in the following to keep the diagrams and equations as simple as possible. The transfer functions of the resonant system observers are shown in appendix C.

For design, the method of eigenvalue placing was used. Though the eigenvalues of an observer can be placed arbitrarily in general, the usual pole placing strategies mentioned in section 3.2 led to poor results.

Better results were obtained when the poles were placed in a configuration not too far away from the system's natural pole locations. See table 3.2 for a summary of the pole configurations used.

The two mass speed-only observer is 4th order. The system's natural poles are two poles at $s = 0$ and one conjugate imaginary pole pair, representing the resonant frequency. The observer feedback's goal is to move the poles to the left, making the system stable. For the resonant poles, the absolute value was changed only little, in order to move it away from the encoder oscillation range at 1000 to 1300 Hz. The poles were turned around the origin to the left by either 20 or 45 degrees. Turning them further, in order to get a faster observer, resulted in too much noise in the estimated state variables. The two poles at the origin could be moved far to the left to $s = -3000$.

The corresponding three mass system is of 6th order, with one more imaginary pole pair representing the second resonant frequency. These poles' absolute value was enlarged a little, then they were turned to the left by 10 degrees. Further turning was not necessary.

As an alternative observer structure, the estimated encoder angle as shown in fig. 3.4 can be compared to the measured encoder angle, with the difference used for feedback. This structure is called the complete observer; it is shown in fig. 3.5.

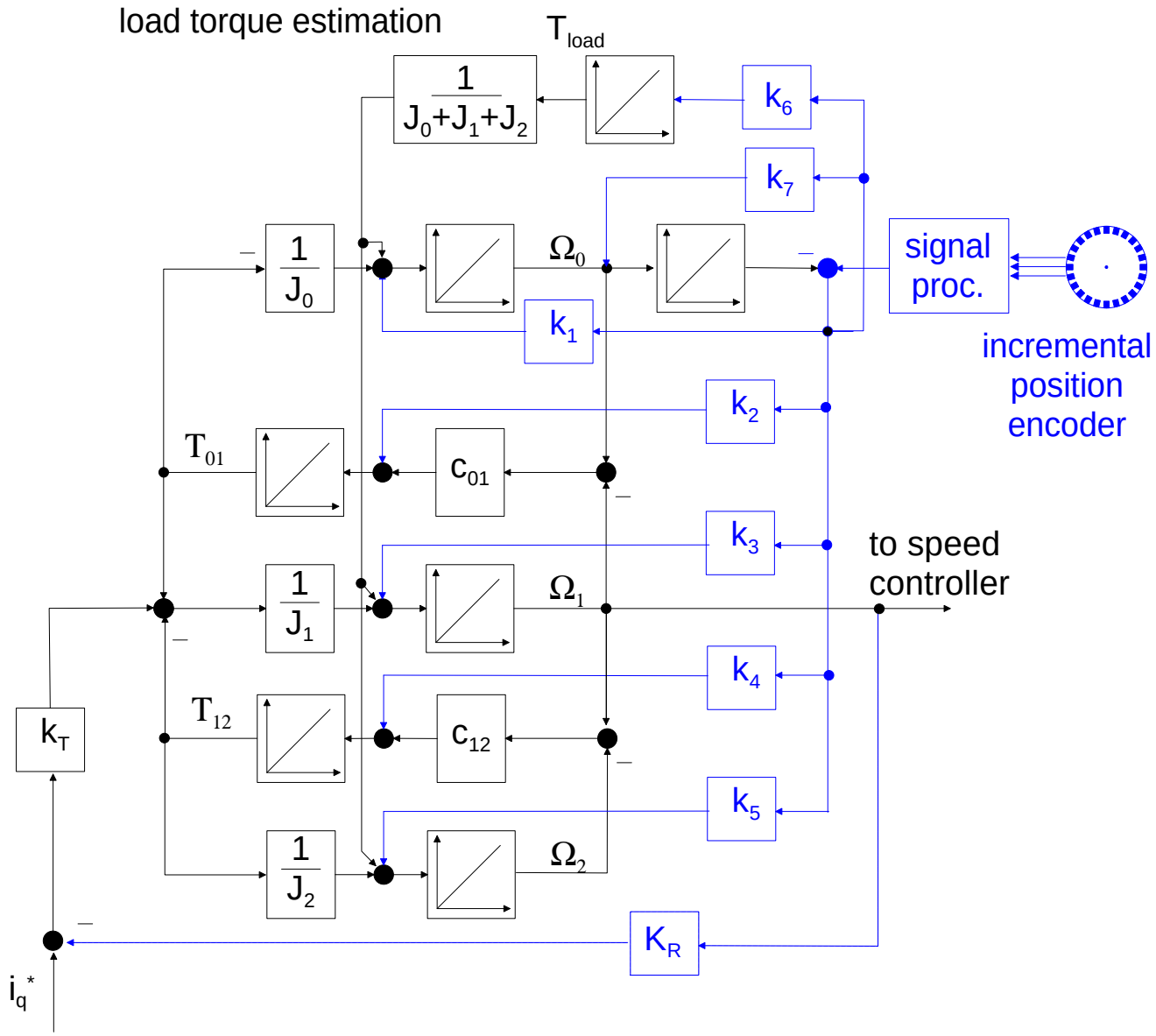


Figure 3.5.: Block diagram of the three-inertia system complete observer

The results are shown in fig. 3.8 no. 3,4,7,8. The order of the complete observer is one more than the speed-only observer. It is good to place the additional pole considerably faster than the others, as the observer will show low frequency oscillations otherwise. The block diagram of the complete observer is shown in fig. 3.5.

The third variant tested is the reduced observer [7] (fig.3.8 no. 9,10). This is an alternative method to reduce the order of the complete observer. The derivation is explained in appendix A.5.2; there is no comprehensive block diagram. Moreover, there is no longer the option of predicting the system states one step ahead.

Table 3.2.: Summary of pole configurations used with 2048-line encoder

observer poles [rad/s]	
Version 1: 20° damping	Version 2: 45° damping
-3000, -3000	
$-1881 \pm j5168$	$-3889 \pm j3889$
additional poles for three inertia observer: $1562 \pm j8863$	
additional pole for complete observer: -10000	

Table 3.3.: Summary of pole configurations used with 5000-line encoder

observer poles [rad/s]
two inertia system observer $-1710 \pm j4698$, -3000, -3000
three inertia system observer $-1710 \pm j4698$, $1562 \pm j8863$, -3000, -5000

3.5. Experimental and Simulation Results

Fig. 3.6 shows the results achieved with the 2048-line encoder and observers as described in sections 3.2 and 3.3. All observers were designed as stated in tables 3.2 and 3.3. In order to tune the control loop for the given speed quality of 0.1 rad/s, only the PI controller gain was changed (see section 1.3).

The results are very poor, compared to those achieved with simple low-pass and notch filters (fig. 2.7). The simple rigid-body observer performs weaker than an low-pass filter. The observers with abstract oscillation model do not reach the performance of a notch filter, while for both speed computation methods the resonant frequency needs to be known.

Fig. 3.7 compares the transfer functions of the two best-performing rigid-body observers to that of a 2nd order filter. It can be seen that both observers do not achieve an equal low-pass behavior, though they are of 3rd and 5th order, respectively. The rigid body observer neither implement knowledge about the resonance, nor suppress it sufficiently, which is an explanation for the poor performance. The addition of an abstract oscillation model results in a transfer function modification much like that of a notch filter, with a similar effect on performance. However, this cannot remove the basic performance lack of the rigid-body observer.

The observers for two and three inertia systems, whose results are shown in

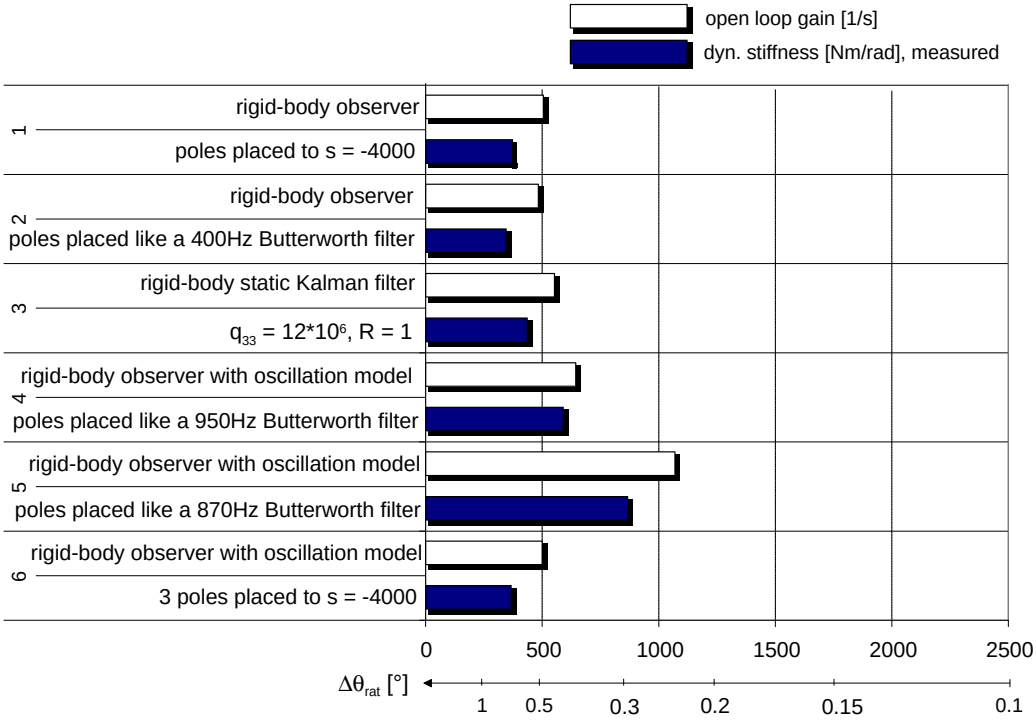


Figure 3.6.: Experimental results with 2048-line encoder, rigid-body observers

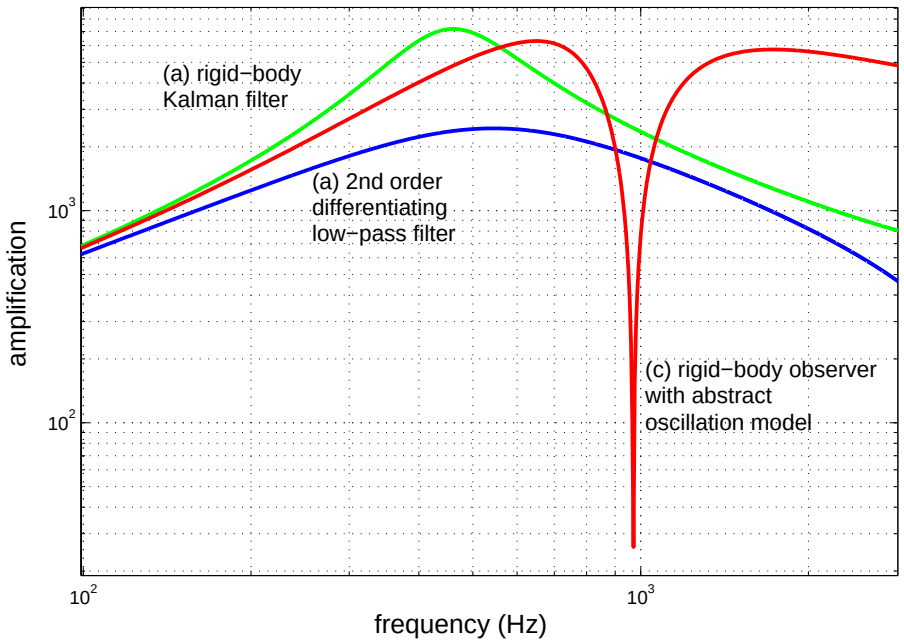


Figure 3.7.: Transfer functions of rigid-body observers compared to a differentiating filter

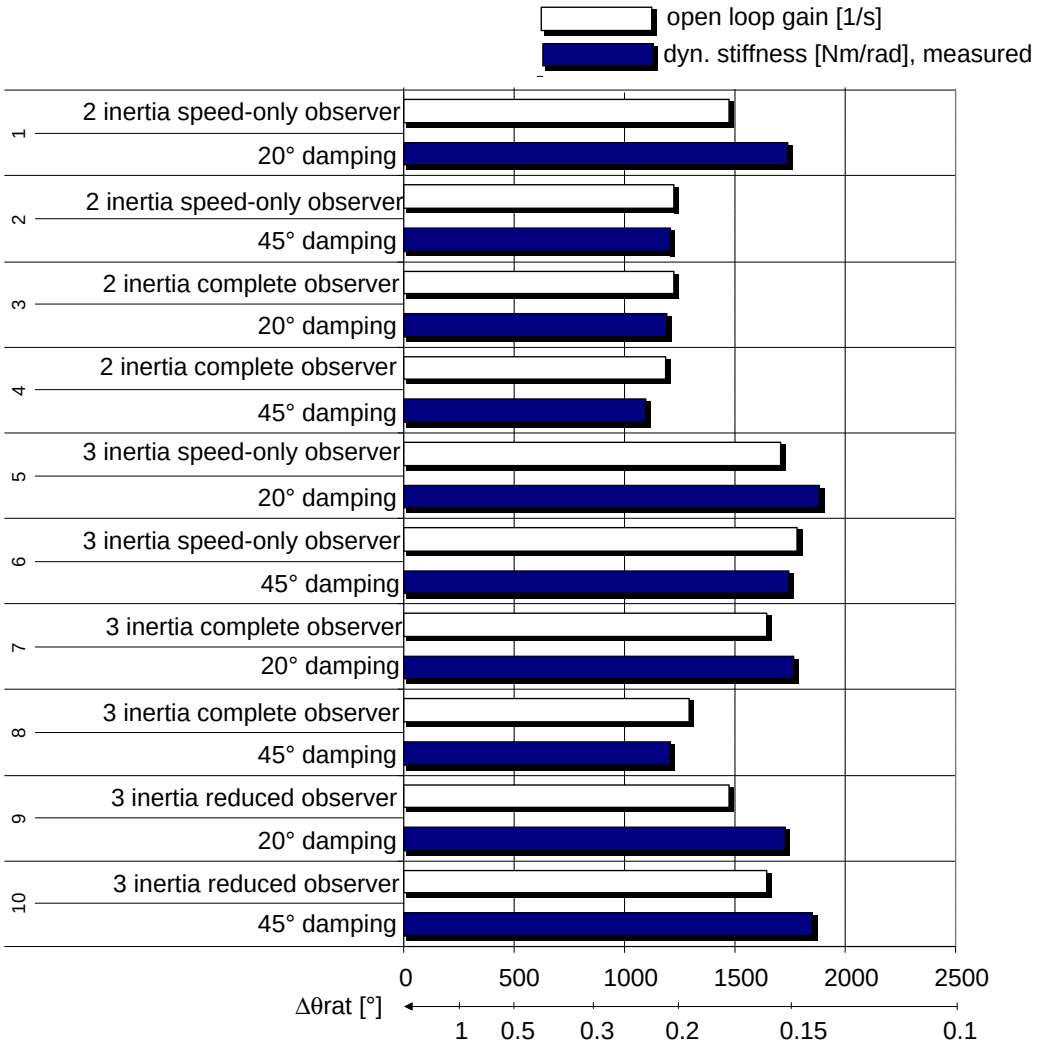


Figure 3.8.: Experimental results with 2048-line encoder, resonant system observers

fig. 3.8, do implement knowledge about the mechanical resonance in a physically quite correct way. As a result, their performance is significantly better than that of a filter.

Best performance is achieved using the speed-only observer for a three-inertia system (fig. 3.8 no. 5, fig. 3.4). Compared to the reduced observer, it has the advantage of realizing a prediction of the state variables. Since the reduced observer is also more difficult to design, it is not suggested here. The complete observer has a higher order and tends more to low frequency oscillations. This is most likely due to model imperfections, which tend to move the control system's eigenvalues toward the stability limit [37]. These oscillations were the limiting factor for the control loop gain of fig. 3.8 no. 7 and 8; their speed quality was better than demanded. Due to those problems, the complete observer also performs weaker than the speed-only structure.

The two-inertia speed-only system observer no. 1 shows a little inferior performance than its three-inertia counterpart, while other structures based on the two inertia model yield much inferior results. This suggests that the approximation of the system as a two mass system is sufficient if the observer is designed carefully, but it is shown here that the results are better and more reliable using the three mass system. However, identification and modeling of the two mass system are much easier. Concerning the pole placing, placing to stronger damped locations was not successful. It enlarged the noise in the feedback signal without achieving a better damping effect of the whole control. Placing the observer's poles to stronger damped locations is necessary if the model precision is lower.

Fig. 3.9 shows the results achieved with the 5000-line encoder and resonant system observers. They are not much better than those shown in 3.8; this indicates that an observer-based control does not depend as much on sensor quality as a filter-based control does. Again the three-inertia system observer performs better than its two-inertia counterpart, however the difference is small.

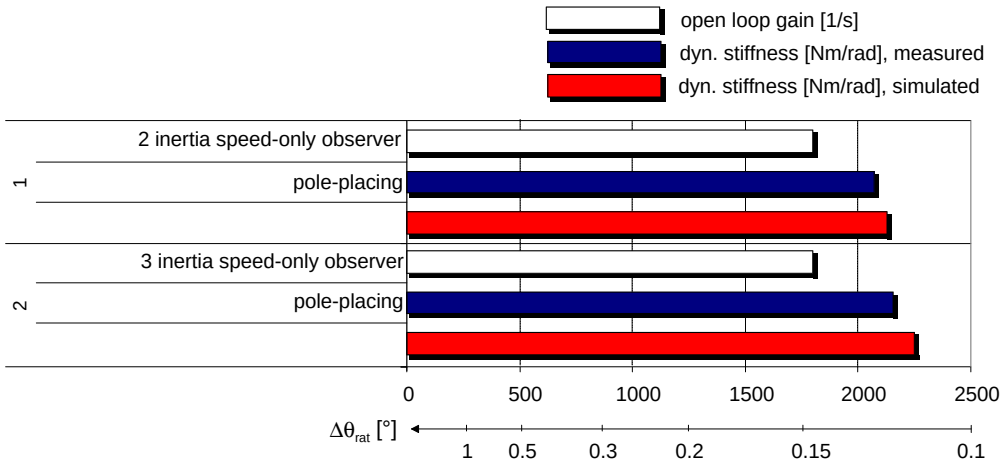


Figure 3.9.: Experimental and simulation results with 5000-line encoder, resonant system observers

4. State control

4.1. Introduction

Fassnacht [36, 37] reported that active damping of a three-inertia plant is possible using only a PI controller. This is also true for the setup regarded here, however the achievable damping is much lower; and it is necessary to use the motor speed rather than the sensor speed. As opposed to the standard case, sensor and actuator of the plant regarded here represent different inertias. At the oscillation frequency, these two masses will oscillate in phase opposition to each other. If speed is measured on one of them and fed back to the other, the oscillation will be excited instead of damped. Fig. 4.1 shows the root locus plot when the (ideally measured) motor speed Ω_1 is fed back to motor current by a proportional controller with a deadtime of $90\mu s$[1]. The achievable damping for the lower resonant frequency, which is dominant in practice, is very poor: the optimum pole location is $-630 \pm j5250\,rad/s$, meaning a 5% settling time of 4.7ms (see appendix A.4), at a gain which cannot be realized in practice due to noise, unmodeled resonant frequencies and additional delays.

A better active damping is only possible using a state controller. However, one fact indicates that it will be difficult to damp the 1st resonant frequency. It is visible in fig. 7.1, where the resonance modes corresponding to the two oscillation frequencies are shown. The 870Hz resonance hardly appears at the center mass, where it should be damped by the actuator. Active damping will require much torque at that frequency to take the energy out of the system.

4.2. State controller design

The state controller was designed as a PI-state controller for the speed (see A.4). The speed reference value is fed into the state control system by a proportional prefilter; the prefilter $\mathbf{v}$, which is a scalar for single-input-single-output systems, plays a similar role as the proportional gain K_R in the PI controller. Thus, the integration time constant was also chosen appropriately using the extended

[1] The 3rd order Pade approximation was used for the deadtime; however, the figure is zoomed on the mechanical system's poles

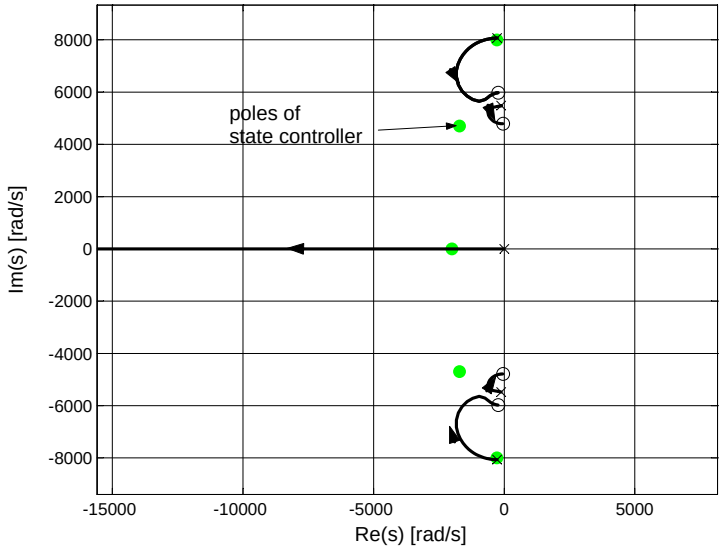


Figure 4.1.: Root locus graph for proportional feedback of Ω_1

Symmetrical Optimum (1.9). The PI-state controller may be cascaded with a position controller, allowing an easy comparison to the P/PI-control systems.

The feedback vector was designed by pole placing, with the goal of shifting the resonant poles directly to the left as it was proposed by Goslar [35]. The pole pair representing the lower resonance frequency was moved towards the left by 1710rad/s, the pole pair for the upper resonance was nearly left in place as this frequency does hardly appear in the measurements. The pole at $s = 0$ is moved to the left; its position determines how fast the controller will be. Fig. 4.2 shows the pole locations that were used for state controller and observer, the state controller's poles are also shown in fig. 4.1.

[37] proposes a linear-quadratic state controller design (see A.4) for a three-inertia resonant system. Only the main diagonal of $\mathbf{Q}$ is used, the inserted values are optimized iteratively. However, using this control method for the setup regarded here, it is not possible to design a state controller that damps the 870 Hz resonant frequency better than the PI controller; the design only shifts the pole at zero and the poles representing the 1370 Hz resonant frequency to the left. Since damping of the first resonant frequency was the design goal of the state controller, this method was not further investigated.

4.3. Observer design

For observer design, the same methods and rules were applied as mentioned in sec. 3.4.2. However, design is much more complicated because the state controller demands very good observation of all system states. Especially, a phase lag in the observation of the load-side system states Ω_2 and c_1 is very likely, since those quantities are physically located far away from the sensor, and very critical because they are fed back with high gains.

Thus, observer feedback was first optimized in a simulation. The simulation model shown in fig. 1.7 was used with the plant model parameters exactly matching those used for the observer. The sequence of a reference change and a load change was simulated, and a weighted sum of the deviations between plant quantities and their observed counterparts was used as the target function. In addition, the pole radius (in continuous domain) had to be limited and the observer's low-pass behavior was taken into account to prevent the control from exciting the unmodeled resonant frequencies.

Leaving the controller parameters constant, observer feedback was optimized using the Simplex algorithm [16, fminsearch].

The resulting pole pattern is shown in fig. 4.2. It is remarkable that the resonant poles are not much faster than the state controller's; the pole pair representing the lower resonance frequency is slower, i. e. further to the right. Pole patterns with faster observer poles would cause oscillations even at steady state, or require a controller design with less damping. The used pattern means that concerning the oscillations, the observer relies on its model rather than sensor feedback. Only the overall speed and load torque are observed with much faster time constants than the state controller's.

4.4. Experimental and simulation results

The used state controller has an open-speed loop gain of $K_R K_S = 1230/s$ and achieved a dynamic stiffness of $1352\frac{Nm}{rad}$. Gain cannot be raised further because of the steady-state performance required. This is the price for active damping: it does not contribute to dynamic stiffness, but creates inquietude in the control system, thus reducing the allowable controller gain.

The effect of active damping is seen best regarding a stepwise setpoint change of the speed control [36]. The step excites the plant's resonant frequencies, and the current limit prevents damping by the controller. As soon as the current limit is left, damping starts. Fig. 4.3 shows measured graphs of reference current i_q^* and encoder speed Ω_0 using different controls. Fig. 4.3(c) shows that the state

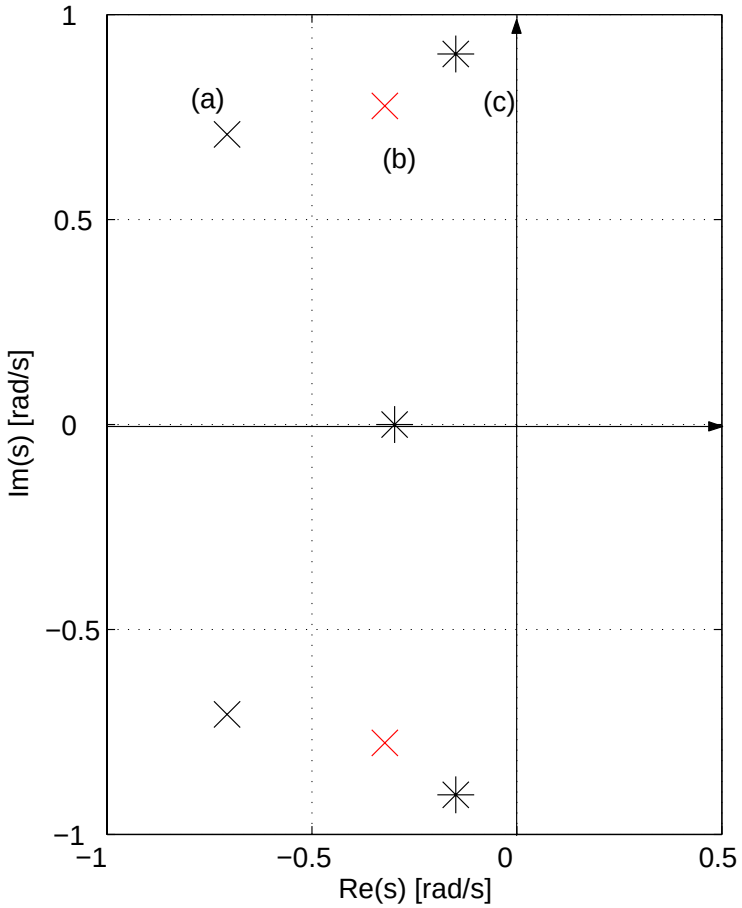


Figure 4.2.: Pole locations of state controller and observer

controller is able to damp out the oscillation within two periods. The PI controller using an observer (fig. 4.3(b) provides a weak active damping, while the damping of a PI control using a filter is not visible in the time range regarded here (fig. 4.3(a)).

4.5. Variable structure control

In the previous section, it was shown that the PI-state controller can achieve a good active damping. However, its stiffness against load torque changes is very low, compared to a PI control with feedback from a resonant system observer. This suggests the idea of a variable structure control that combines the advantages of both controllers.

The control described here uses a weighted sum of the outputs of two different controls, instead of switching between them, and has no sliding mode. This is different from most literature cases.

In this thesis, controls were used where the position and speed controllers are designed according to fixed rules in order to keep them comparable. Moreover, the resonant system observer's system models imply the proportional part of the speed control to yield a low phase lag; this is especially necessary for oscillation damping. The observer feedback design has to be different for the PI and PI-state controller. Those facts make it difficult to use any part of the control structure in common. Thus, a control was designed where two independent controls are computed in parallel, and the output i_q^* is combined from the two system outputs.

The structure is shown in fig. 4.4. The actual speed, computed as the discrete-time derivative of position θ, is compared to the reference speed, which is assumed zero for position control. Of the result, a weighting function λ is computed

$$\lambda(\Omega - \Omega^*) = \left\{ \begin{array}{ll} 0 & |\Omega - \Omega^*| \leq \omega_1 \\ \frac{|\Omega - \Omega^*| - \omega_1}{\omega_2 - omega_1} & \omega_1 < |\Omega - \Omega^*| < \omega_2 \\ 1 & \omega_2 \leq |\Omega - \Omega^*| \end{array} \right\} \tag{4.1}$$

The output of the P/PI-control $i_{q,I}^*$ is weighted by λ, the output of the P/PI-state control $i_{q,II}^*$ is weighted by $1 - \lambda$. In the same way, the two estimates for the new system state $\hat{\mathbf{x}}_I(k+1)$ and $\hat{\mathbf{x}}_{II}(k+1)$ are joined.

Best results were achieved for $\omega_1 = 0.5$ rad/s and $\omega_2 = 1$ rad/s. The performance of this control was simulated. The simulated dynamic stiffness is 1674 Nm/rad, which is in between the stiffness values of the PI and PI-state control. The oscillation damping performance is shown in fig. 4.5. Active damping is visible, but less than with the pure PI-state control.

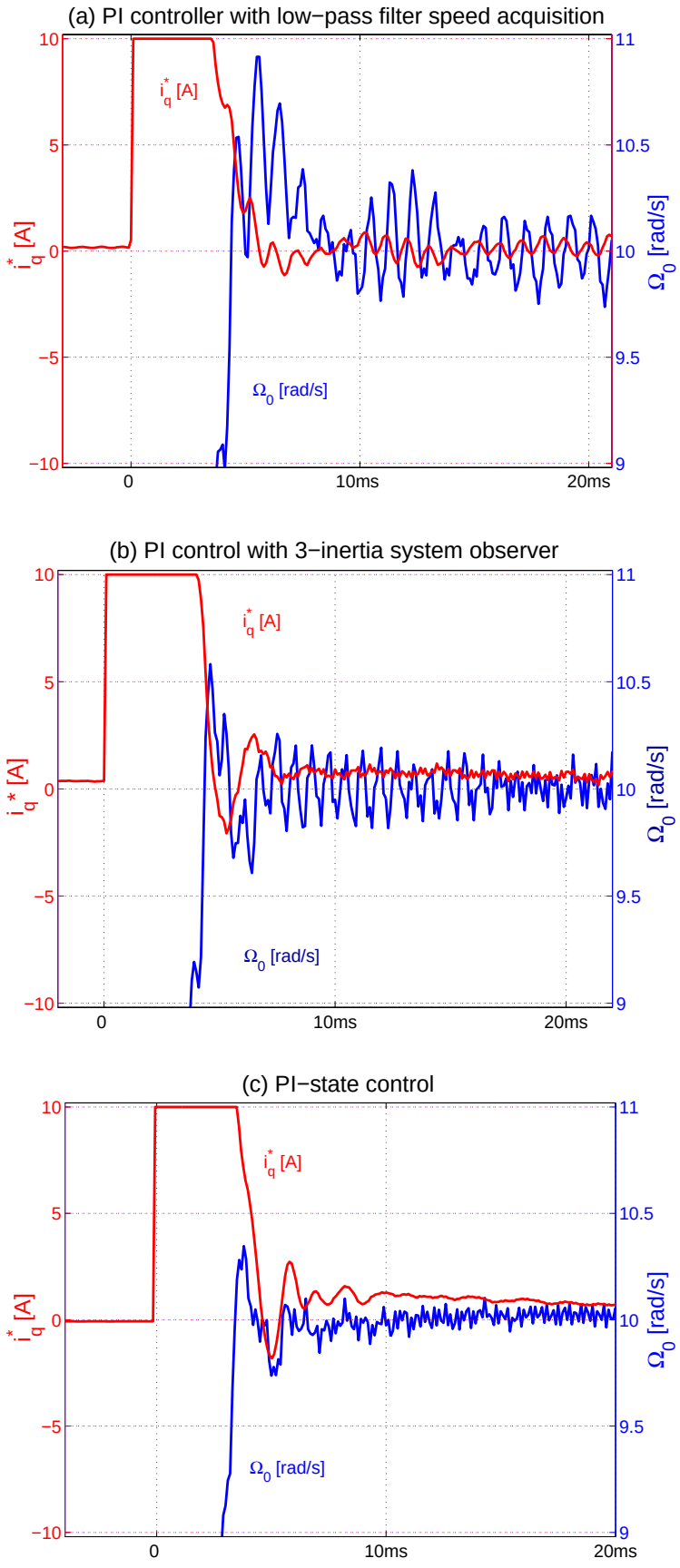


Figure 4.3.: Setpoint change during speed control; experimental results

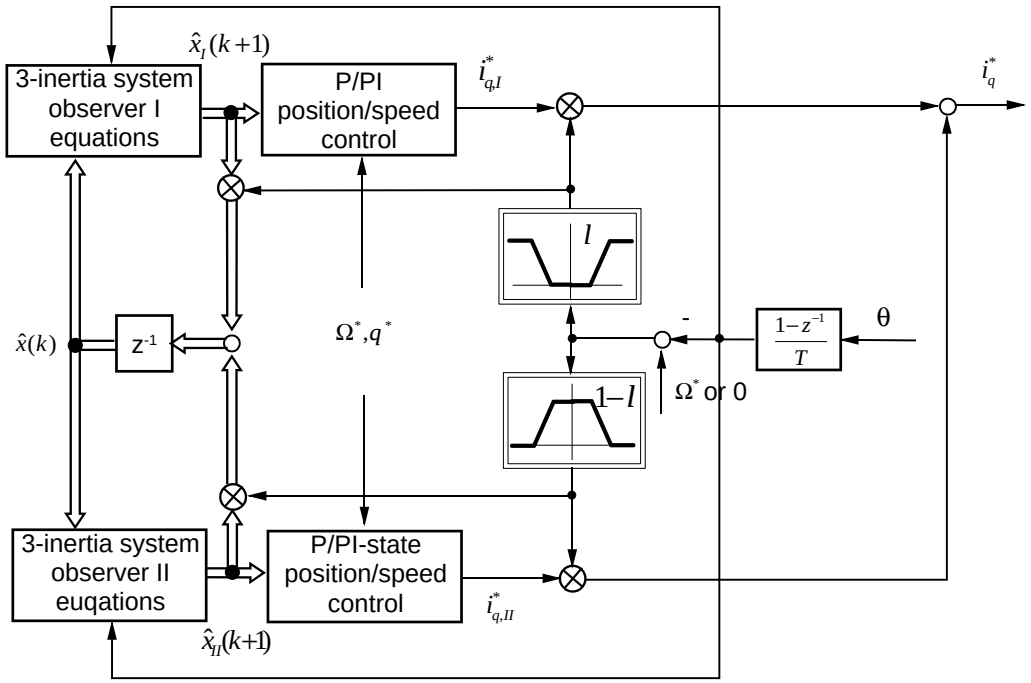


Figure 4.4.: Variable structure control

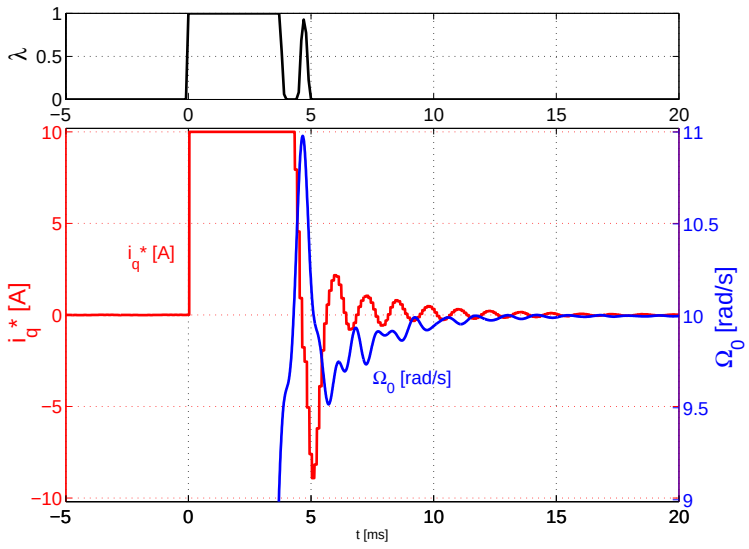


Figure 4.5.: Oscillation damping of structure-variable control (simulated)

A different choice of the switching levels ω_1 and ω_2 yields a different performance, nearer to either one of the basic controls. However, it can be concluded that is is possible to achieve controls with performance rating between the basic controls, but not to combine the advantages of both.

5. Correction of systematic errors in sinusoidal encoder signals

5.1. Introduction

This chapter deals with processing the encoder signals to get the position information, which is block (1) in fig. 1.4. Most of the contents has been published in [2].

Sensor signals have two categories of errors: systematic errors and noise. While hardly anything can be done about noise -except filtering, with the disadvantage of introducing a phase lag-, systematic errors may be identified and compensated, because they appear every time in a specific way.

The errors of sinusoidal encoder signals can be described with the equations

$$\begin{aligned} x(\theta) &= O_x + A_x * cos(\theta + \Delta) \\ y(\theta) &= O_y + A_y * sin(\theta) \end{aligned} \tag{5.1}$$

with encoder signals x and y, and position angle θ. O_x and O_y represent offset errors, that are caused by the encoder's electronics. Though the long-term offset was compensated in the signal processing electronics, offset errors varying from encoder period to encoder period still disturbed the signals. Algebraic calculations reveal that offset errors cause a position error of sinusoidal form with a frequency equal to the encoder signal frequency, if the signals are processed the normal way (see section 1.2.1). A_x and A_y represent the amplitudes of the encoder signals. While the amplitude itself does not matter because of the division in (1.2), an amplitude difference will cause a position error, again sinusoidal with a frequency of twice the sine frequency. Δ stands for a phase error, caused mainly by imprecise placement of the encoder's two light sensors. Δ can be considered much smaller than 1; it causes an error frequency of twice the sine frequency. An error source not mirrored in (5.1) are harmonic errors, i. e. deviations from the ideally sinusoidal shape. For example, many encoders yield signals tending to a triangular rather than sinusoidal shape. Position errors caused by harmonics may contain all multiples of the encoder sine signal frequency. All errors can vary along an encoder turn, and may change when the encoder is re-mounted.

Probst [46] reports that a considerable part of the systematic encoder errors is dependent on rotation speed; making those errors impossible to correct without additional effort. This was not found true for the setup regarded here; anyway, precision positioning and smooth rotation is most interesting at low speed in practice.

The resulting position inaccuracy is discussed in section 5.7. However, the most problematic errors are errors in the speed determined from the position signal. Fig. 5.1 shows the r.m.s speed error at different speed, simulated with error data from the respective encoder's parametric correction table (sec. 5.3), neglecting noise and harmonic errors. Since the errors are sine functions of the angle, the error magnitude in the derivative of position are proportional to the speed - as long as the error frequencies do not exceed the Nyquist frequency, and neglecting the low-pass characteristic of the difference quotient. Because the main errors of this encoder are errors in amplitude and phase, the significant error frequency is twice the encoder frequency. In the lower range, the speed error amplitude for different encoders behaves equal, because the greater precision of an encoder with more lines is countervailed by the higher error frequency.

The comparative measurements reported in sec. 5.6 were done at π rad/s, at the maximum of speed error for the 5000-line encoder. In fig. 5.1, it is also visible that the speed error is much smaller if a low-pass filter is used - in a control, this is anyway necessary because of the resonant frequencies. The maximum possible use of error correction will be to compensate a speed error of $8 * 10^{-3} rad/s$, thus allowing a faster control that would have a speed deviation larger by $8 * 10^{-3} rad/s$ without correction. This is about 1/10 of the total speed error limit of $0.07 rad/s$ for the 5000-line encoder (see sec. 1.6).

Another source of imprecision is measurement noise. Noise is most easy to investigate from a measurement at standstill. The resulting imprecisions are shown in tab. 5.1. The covariance between sine and cosine track noise as well as between subsequent position samples is approximately zero, and the frequency distribution is equal. Thus, measurement noise can be seen as ideal white noise, meaning that it is equally distributed in the frequency spectrum and that, in time domain, independent random values are added to each sample.

This thesis resumes three correction methods for systematic errors of optical incremental encoder signals. It describes the procedure of acquiring identification data and applying the error correction.

Measurements were carried out using two setups: Setup I for control performance tests and setup II for run-down experiments. Setup II was suitable for this purpose because of its large inertia. As a switching inverter can be disturbing in error identification, run-down data promise better results. Run-down was recorded at $21 \mu s$ sampling time.

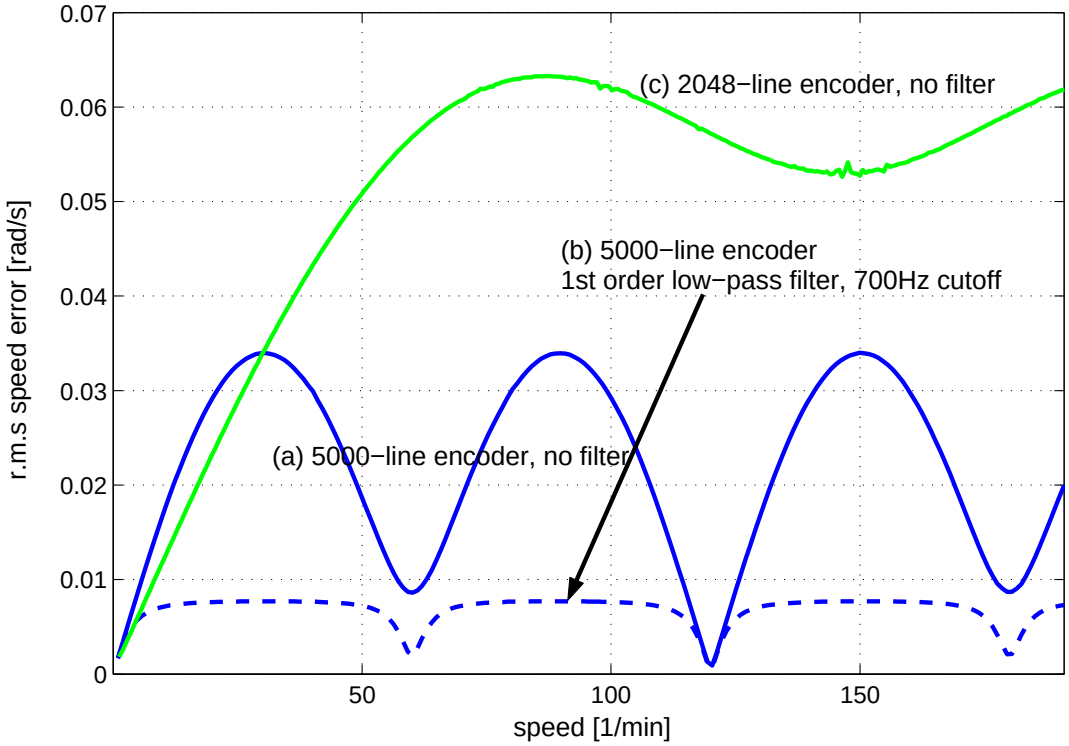


Figure 5.1.: Simulated speed error, $100\mu s$ sampling time

Oversampling means sampling an input signal several times per controller sampling period and using the average value. This was not realized, but is discussed theoretically in sec. 5.8. This method is proposed among others by Kirchberger and Hiller [48] as a means to improve the signal quality. They use a controller sampling time of $100\mu s$ and 1MHz analog-digital-converters, thus the signal is oversampled 100 times. Gees [23] proposes a variant called "burst oversampling" where the samples are taken only in the last part of the sampling period, not during the whole time; this reduces delay, but also reduces the effect if the same AD-converter sampling frequency is used. Since only the principal effect shall be shown here, full-period oversampling will be regarded.

5.2. Position error table

Errors can be corrected by recording a position error lookup table, which contains the identified position error at equidistant points along a whole encoder turn. For each measured position, the angular error is determined from the table, and subtracted from the measured value. This method can theoretically correct any position error.

For data acquisition, a record of a run-down at high sampling rate is needed,

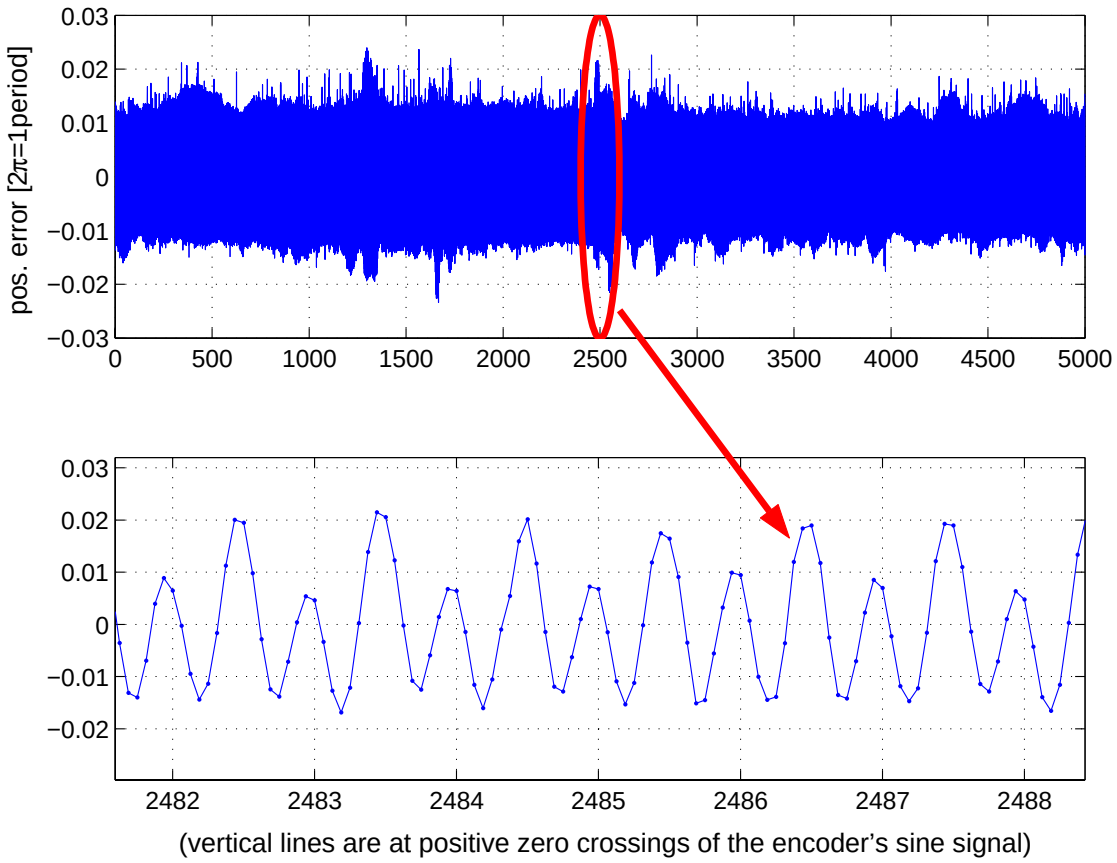


Figure 5.2.: Correction table for 5000-line encoder

so that many points per encoder period are available. For each period, a linear regression of the sampled data to a 2nd order polynomial reveals "true" position values. Linear regression was used rather than interpolation [43] because it reduces noise. A higher polynomial order did not improve performance. The data pairs of measured and "true" value are re-sampled at 16 equidistant points per period using spline interpolation; the necessary correction at those points is stored in the correction table. For each measured position value, the correction is computed using interpolation; linear interpolation is used here for performance reasons.

Figure 5.2 shows the identified correction table for the 5000-line encoder. The errors consist mainly of a 2nd harmonic of the encoder cosine signal, meaning that phase errors are the most important ones. They are nearly periodic, because the errors change only very slowly from one period to another.

In order to remove noise from the correction table, it is possible to filter the correction table data. Due to the periodic nature of the data, it is suitable to collect correction values from all encoder periods that are at the same position inside their respective period. Each of these sequences can be smoothed by applying a digital low-pass filter along it. Forward and backward filtering [16, filtfilt] should

be used to remove the delay. A 2nd order discrete-time filter with two poles at $z = 0.95$ yielded good results.

5.3. Parametric table

A second possibility is recording a parametric table as proposed by Hoescheler [43, 44] and Probst [46]. It considers gain, offset, and phase errors, identifies them for each encoder period, and uses this table for correcting the encoder signals. Thus, it is not capable of detecting harmonics in the sine signals.

Gain, phase, and offset errors can be described using (5.1). Using trigonometric function theorems, this leads to the ellipse relation

$$\begin{aligned} F(x,y) = \; & -Ax^2\, {A_y}^2\, cos(\Delta) + \left({A_y}^2\, (O_x - x)^2 + {A_x}^2\, (O_y - y)^2 \right)\, sec(\Delta) \\ & +2\; A_x\; A_y\; (O_x - x)\; (O_y - y)\; tan(\Delta) \\ = \; & 0 \end{aligned} \tag{5.2}$$

Correction data are computed by a nonlinear regression [16, lsqnonlin] of the sampled data to this relation. After extracting the data concerning a certain period from the sample, it is suitable to initialize the offset parameters to the respective average, and the amplitude parameters to the peak values. 6 iterations of nonlinear regression are sufficient to reach the final correction values; iteration was aborted when the maximum requested parameter step fell below 10^{-10}. Computing the parametric table takes about 20 minutes on a Celeron-500MHz PC running MATLAB.

Since the absolute signal amplitude is not important for position calculation, correction can be done using only four error parameters $\frac{A_y}{A_x}$, O_x, O_y, Δ:

$$x^* = (x - O_x) * \frac{A_y}{A_x} + (y - O_y) * \Delta \tag{5.3}$$

$$y^* = (y - O_y) \tag{5.4}$$

$$\theta = \ldots atan2\left(\frac{y^*}{x^*}\right)\ldots \tag{5.5}$$

(the complete equation for θ is (1.2))

The advantage of this approach is that for error identification, a high sampling rate is not required. The time relationship of the data points is not important, and data points from several runs through the same encoder period can be combined. Even speed fluctuations in the identification data are tolerable.

Hoescheler proposes the use of an additional correction table to correct the harmonic errors in the encoder signals, which are not comprehended by (5.1) and

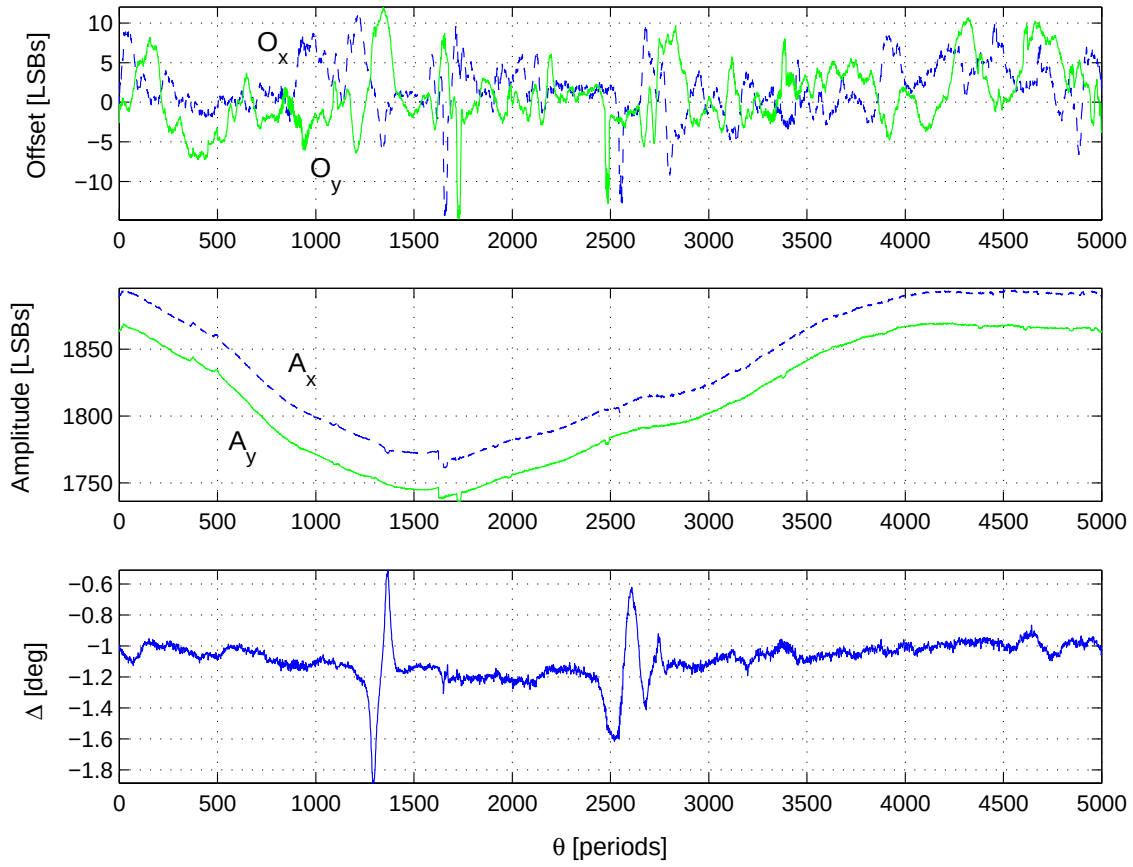


Figure 5.3.: Error parameters for 5000-line encoder

therefore cannot be corrected using the parametric table. As opposed to section 5.2, only a single data set of 16 points is computed, and applied to every period after parametric correction. The idea is that the sine shape imperfections do not change from one period to another. The data for computation of the correction data set are stochastically selected from all encoder periods.

Fig. 5.3 shows the error correction parameters of the 5000-line encoder used in this thesis. The first two graphs are scaled in LSBs of a 12-bit AD-converter. It can be seen that track x has a constantly larger gain that track y, but the signal amplitude also changes along an encoder turn. The phase error is nearly constant, with two large variations around lines 1400 and 2600. Regarding the offset errors, no explainable pattern was found.

5.4. On-line correction method

The third possibility is an on-line identification and correction method which does not require a table. It is again based on (5.1), and was proposed by Bünte and Beineke [45].

The approach of this method is to compute the first few Fourier series coefficients of the signal

$$s(\theta) = x_c^2 + y_c^2 - 1 \tag{5.6}$$

where x_c and y_c are corrected versions of the encoder signals. Using the assumptions

$$\begin{aligned} \Delta &\ll 1 \\ |\,O_x\,|, |\,O_y\,| &\ll A_x, A_y \end{aligned}$$

straight-forward calculations reveal that

$$\int_0^{2\pi} s\, d\theta \approx \pi \left({A_x}^2 + {A_y}^2 - 2 \right) \tag{5.7}$$

$$\int_0^{2\pi} s * \cos(\theta)\, d\theta \approx 2\pi\ A_x\ O_x \tag{5.8}$$

$$\int_0^{2\pi} s * \sin(\theta)\, d\theta \approx 2\pi\ A_y\ O_y \tag{5.9}$$

$$\int_0^{2\pi} s * \cos(2\theta)\, d\theta \approx \frac{\pi}{2} \left({A_x}^2 - {A_y}^2 \right) \tag{5.10}$$

$$\int_0^{2\pi} s * \sin(2\theta)\, d\theta \approx -\pi\ {A_x}^2\ \Delta \tag{5.11}$$

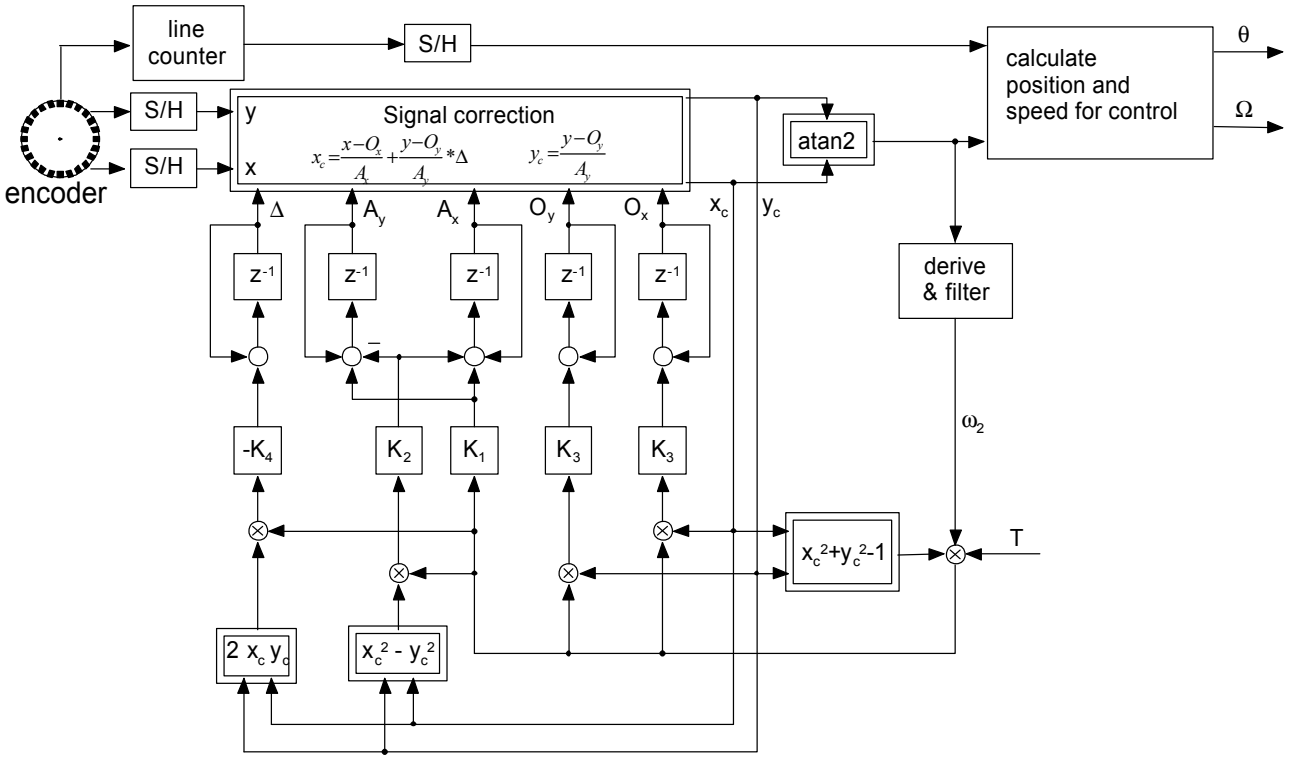


Figure 5.4.: Online error correction algorithm

(5.7) is used to determine the average amplitude, (5.8) and (5.9) for the offsets, (5.10) for the amplitude difference and (5.11) for the phase error Δ.

The complete algorithm is shown in figure 5.4. It results when $d\theta$ is replaced by $\omega_2 * T$, and all trigonometric functions are expressed using x_c and y_c. The sampled encoder signals are corrected using the recent set of error parameters, in a way that the amplitude of the corrected signals is around 1. Using x_c and y_c, the Fourier coefficient integrals are computed over one sampling period. The result is added to the respective error parameter, using the feedback constants $K_1..K_4$.

The algorithm is locally stable, if the actual errors are small and all K_is are positive. This was proved by linearization, assuming that all actual errors are zero and computing the average transition matrix for $\theta = 0..2\pi$. The eigenvalues of this matrix are (in discrete-time domain):

$$1 - \frac{2\,K_1\,\omega_2\,T}{A} \quad 1 - \frac{K_2\,\omega_2\,T}{A} \quad 1 - \frac{K_3\,\omega_2\,T}{A} \quad 1 - \frac{K_3\,\omega_2\,T}{A} \quad 1 - \frac{K_4\,\omega_2\,T}{2} \tag{5.12}$$

with the amplitude of the original signals A. Since the algorithm is not globally stable, the error parameters must be initialized near the actual ones; using $A_x = A_y = 2048$ and $O_x = O_y = \Delta = 0$ is sufficient.

The feedback increases proportionally with speed. This is a problem when the sampling theorem is violated regarding the encoder signals: Feedback will increase

further, but the information available per sampling period is limited. Especially when the speed is a whole number of encoder periods per sampling time, the fine interpolation angle will be constant. It was proved that in this case, the linearized system will have three poles at the stability limit.

Thus, the "speed" ω_2 is computed without using the line counter. It is in fact the alias frequency of the encoder signals, with a maximum at $\pi\, f_{\text{sampling}}$ (where $2\pi \widehat{=} 1$sampling period).

Suitable feedback parameters were found using a genetic algorithm (see appendix B.3. As a first step, several random parameter sets between 10^{-10} and 10^{10} were tested; the results were compared to give a coarse idea of the sensible parameter range. Further optimization was done with a genetic optimization method using 4 populations with 20 individuals each and different mutation and selection strategies. All individuals ever tested were recorded. This way, not only one optimal parameter set was computed, but also the suitable parameter range and the distance to instability were visible after 50 generations.

For each individual, a run-down record from setup II was corrected using the parameter set under test; the resulting speed deviation, filtered as described in section 5.5, was used as the target function. Since the run-down record contains a large speed range, the resulting parameter choice should be a robust one.

The possible parameter range for optimum results extends over some decimal periods:

$$\begin{aligned} 100 \le K_1 \le 30000 \\ 20 \le K_2 \le 20000 \\ 300 \le K_3 \le 10000 \\ 0.04 \le K_4 \le 15 \end{aligned} \tag{5.13}$$

However, if the algorithm is used in a control loop, stability is more sensitive, and the parameters must be chosen at the lower end of this range. Concerning order and cutoff frequency of the derivation filter, no performance differences occur between 200..1000Hz and 1st to 6th order, so a 1st order filter with 900Hz cutoff was chosen.

5.5. Open-loop experimental results

Figure 5.5 shows the results of off-line correction of sampled data from both setups. Run-out samples from setup II at $21\mu s$ sampling time and samples from steady-state speed control with low bandwidth from setup I ($100\mu s$ sampling time) are examined.

The raw encoder data were corrected and evaluated. Then the basic speed curve (for run-out from setup II) was approximated by a 3rd order polynomial and subtracted. A 1400Hz high-pass filter[1] was used to filter out low-frequency speed deviations which cannot result from encoder errors [44]. Finally, the speed was computed using discrete derivation. In figure 5.5, the bars show the relative speed error for the respective sample, with 100 % representing signal processing without correction. The numbers at the 100%-bars show the resulting speed deviation in this case. Separate data sets were used for error identification and correction. The speed error at setup II is much higher because of the faster sampling rate.

The speed deviation for both sensors without correction is comparable, keeping in mind that the different line count. However, correction of the 2048-line encoder's signals results in only 14% performance improvement in the best case. This is because this sensor has rather low systematic errors in contrast to much noise, which cannot be corrected. Even in a sample recorded at standstill, the speed deviation was 90% of the reference value. No further correction experiments were carried out for this encoder.

Error correction for the 5000-line encoder at setup II could reduce the error by a factor of 10. The "parametric table & single period table" method performs best because it provides the correction of harmonic errors together with the robustness of the parametric table method. The on-line algorithm performs only a little weaker than the table-based methods.

At setup I, no run-down is possible because of strong slot latching; instead, controlled turning was used with the controller bandwidth limited to less than 100Hz by means of a low-pass filter in the speed acquisition path [1]. The load servo was decoupled to minimize the mechanical resonance.

The parametric table and on-line correction method again provide good results. However, the position error table and single period table methods fail due to disturbance from the inverter. In the former case, filtering the table data restores performance. Since the acquisition of proper error correction tables is problematic at setup I, it was also tried to use the tables from setup II. This is in fact the best possible way, so those tables were used for the following on-line experiments.

5.6. Control loop experimental results

Performance of the correction methods in a control loop was measured at setup I, using tables from setup II. Several controls using different encoder error correction and speed acquisition methods [1] were designed so that the r.m.s deviation of

[1]The r.m.s speed deviation was computed in frequency domain where an ideal high-pass filter could be applied.

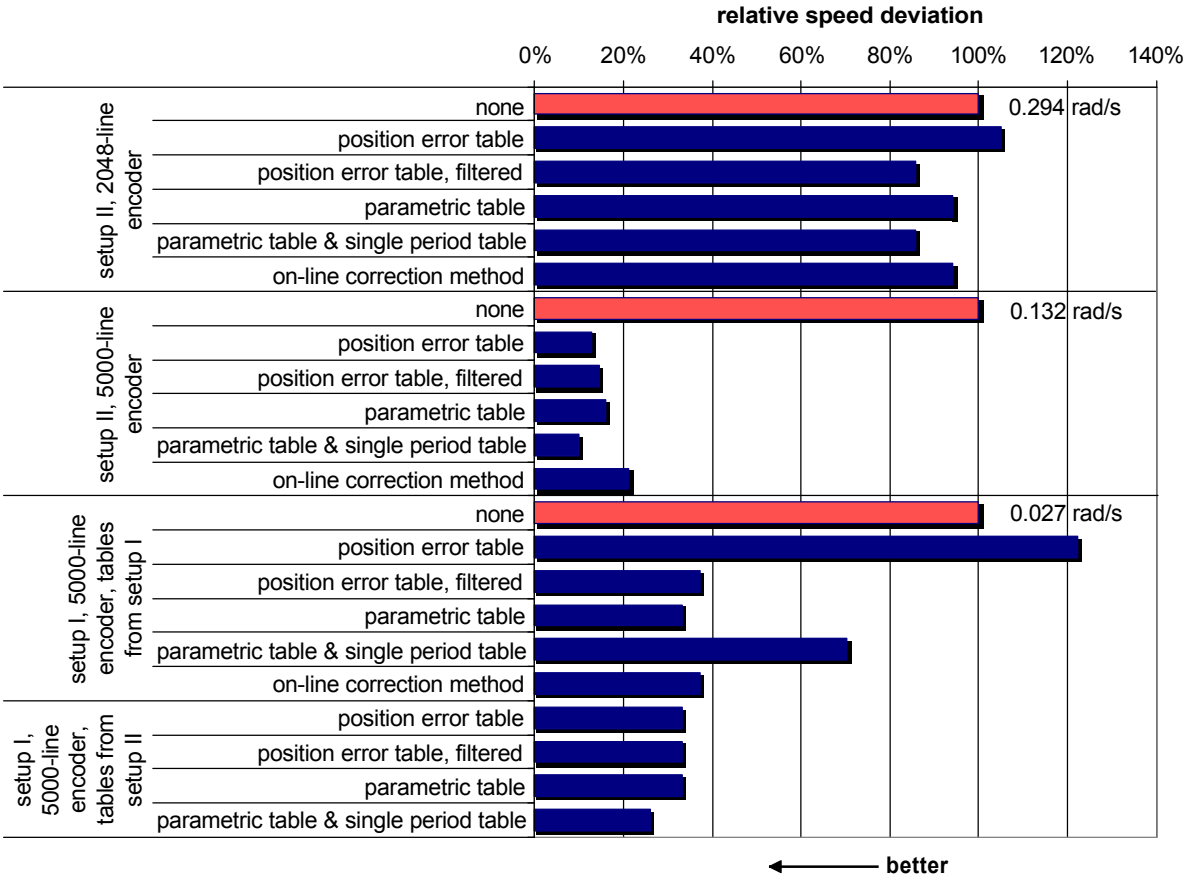


Figure 5.5.: Open loop experimental results

speed achieved at no-load operation was about $\sigma\Omega = 0.055$ rad/s at 1/2 rev/s. Deviation was in all cases computed by off-line discrete derivation without filtering. Before, parametric table and single position error table correction (see section 5.3) were applied to the recorded data. This is necessary to make the results comparable, since only the closed loop effect shall be considered here. After tuning the control, its dynamic stiffness against load step changes was used as the performance index [1].

Figure 5.6 shows the results of the closed loop performance comparison. Error correction improves the performance of all controls, while the performance difference between the three correction methods is small. This makes the on-line correction method especially interesting, because it does not require a pre-recorded table.

At higher controller gains, error correction still makes a larger gain possible, though this does not result in a better stiffness. This seems to be a general problem at high gains, not due to correction.

Comparing the different speed acquisition methods, it is remarkable that the advantage of the high-end methods diminishes when using the speed calculated from *corrected* encoder signals as reference criterion. This could mean that high-gain controls without encoder correction partially "compensate" the encoder errors by acting against it as if it were a real speed deviation, actually worsening the speed quality.

Additional problems could be expected at the speed where the frequency of speed error is equal to the first resonance frequency of the setup, because the encoder errors might excite the resonance. However, this is not the case - measurements at this speed revealed the same improvements due to correction as expressed in fig. 5.6.

5.7. Position accuracy

With high-precision applications, it is important to know the errors of the measured *position*, because a position command can only be tracked with those errors. The previous chapters examined the *speed* error. It will be tried here to estimate those errors from the measured speed errors; however, for an exact measurement, a reference position meter would have been required. A summary of encoder errors for both encoders is shown in table 5.1.

The position error of a sinusoidal encoder consists of three components: noise, systematic errors, and low-frequency error.

As in every measurement, there is noise in the position signal. The noise can be seen as white noise, meaning that it is equally distributed in the frequency

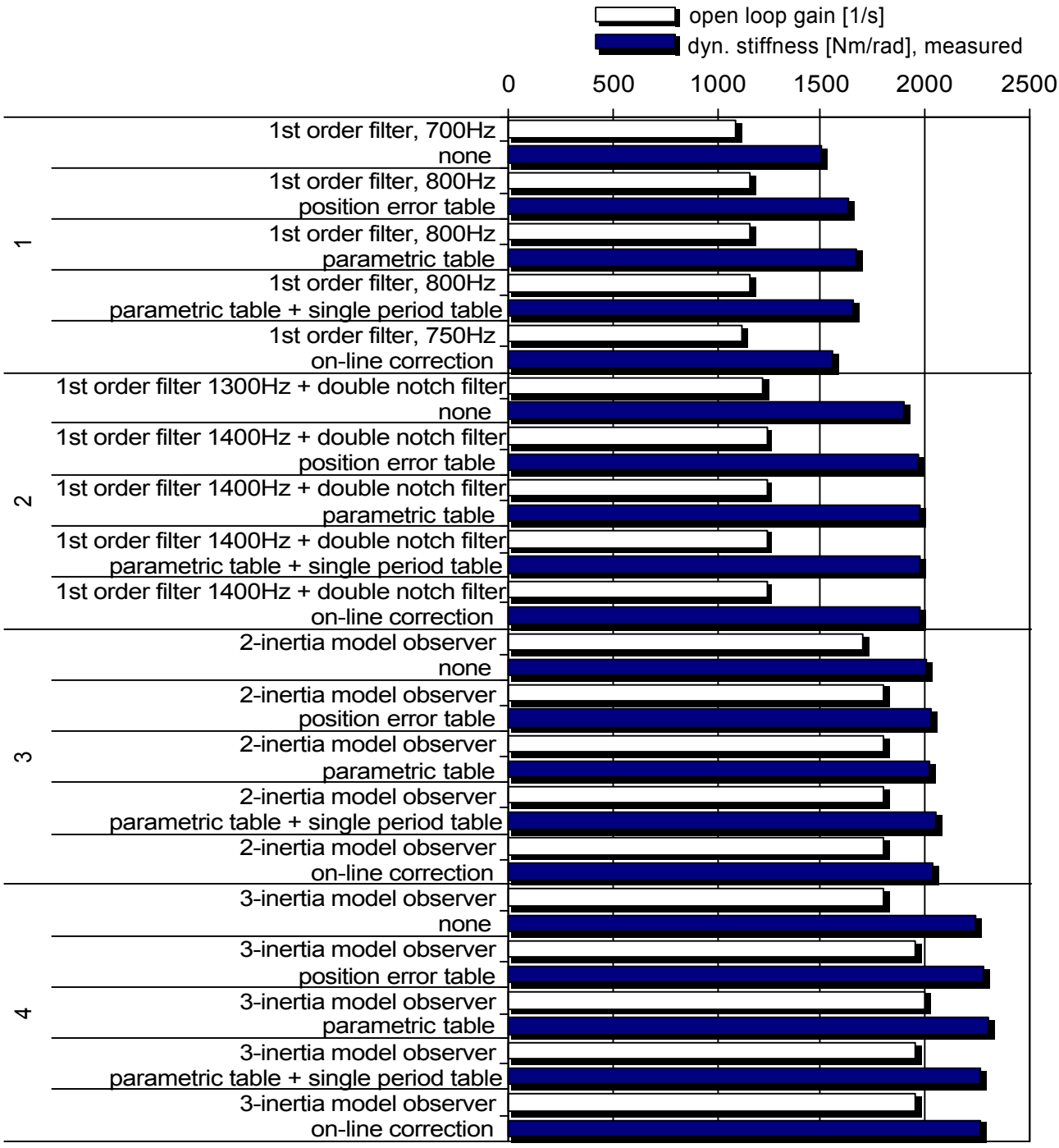


Figure 5.6.: Closed loop experimental results (5000-line encoder)

Table 5.1.: Encoder position errors

Encoder		Noise [rad]	Systematic track errors [rad]	Low-frequency error [rad]
5000-line position	r.m.s	$7.4 * 10^{-8}\,rad$ $5.9 * 10^{-5}\,lines$	$1.6 * 10^{-6}\,rad$ $1.3 * 10^{-3}\,lines$	not measurable
	peak	$3.3 * 10^{-7}\,rad$ $2.6 * 10^{-4}\,lines$	$4.8 * 10^{-6}\,rad$ $3.8 * 10^{-3}\,lines$	
2048-line position	r.m.s	$2.8 * 10^{-6}\,rad$ $9.1 * 10^{-4}\,lines$	$3.7 * 10^{-6}\,rad$ $1.2 * 10^{-3}\,lines$	not measurable
	peak	$1.2 * 10^{-5}\,rad$ $3.9 * 10^{-3}\,lines$	$1.1 * 10^{-5}\,rad$ $3.6 * 10^{-3}\,lines$	
5000-line speed	r.m.s	$0.001\frac{rad}{s}$	$\leq 0.034\frac{rad}{s}$	neglectable
2048-line speed	r.m.s	$0.038\frac{rad}{s}$	$\leq 0.064\frac{rad}{s}$	neglectable

spectrum and that, in time domain, random values are added to each sample. Noise can be measured best using a measurement at standstill.

Another imprecision are systematic encoder errors, as discussed in the previous sections. They can be read directly from the non-parametric correction table.

As a third component, there are usually errors due to imperfect mounting of the encoder. Such errors may contain sinusoidal components with the rotation frequency and its first harmonics. Due to their low frequency, those errors appear in the speed signal only at very high speeds. However, the speed deviations due to the servo motor's poles and notches are much larger at high speed than possible deviations due to imprecise encoder mounting; thus, those errors were not measurable.

5.8. Oversampling

Oversampling is primarily a method to reduce measurement noise. This is of maximum use if measurement noise is uncorrelated white noise. For the noise of the setup's encoder signals, it could be verified that the measurement noise is equally distributed in frequency domain and nearly uncorrelated in terms of covariance.

Table 5.2.: Encoder position errors with 100fold oversampling

Encoder	Noise [rad]	Systematic track errors [rad]
5000-line r.m.s position	$7.4 * 10^{-9}\,rad$	$1.6 * 10^{-6}\,rad$
2048-line r.m.s position	$2.8 * 10^{-7}\,rad$	$3.7 * 10^{-6}\,rad$
5000-line r.m.s speed	$0.0001\frac{rad}{s}$	$\leq 0.025\frac{rad}{s}$
2048-line r.m.s speed	$0.0038\frac{rad}{s}$	$\leq 0.046\frac{rad}{s}$

The r.m.s deviation of the average of N measurements, where single measurements have a deviation σ_0, is then

$$\sigma = \frac{1}{\sqrt{N}}\sigma_0 \tag{5.14}$$

Oversampling also reduces systematic errors, but only during rotation. Thus, it cannot improve precision positioning, but may improve the speed signal. If the number of samples is sufficiently high, continuous sampling over a whole sampling period can be assumed as an approximation. An error signal of sinusoidal form

$$e(\theta) = e_0 cos(N\theta - \gamma) \tag{5.15}$$

is then reduced to

$$e'(\theta) = \int_{\theta}^{\theta+\Omega T} e(\theta)d\theta = e_0 \frac{\sqrt{2 - 2cos(N\Omega T)}}{N\Omega T} cos(N\theta - \gamma') \tag{5.16}$$

The resulting error e' is of the same amplitude at standstill or low speed, is completely erased where $N\Omega T$ is a multiple of 2π, and stays below $\frac{2}{N\Omega T}$ for higher speed.

Tab. 5.2 shows the theoretical noise and systematic errors for 100fold oversampling, as it was proposed in [48]. From a comparison with fig. 5.1, it can be seen that oversampling would be a considerable improvement for the 2048-line encoder. However, with a low-noise encoder like the 5000-line type used in the experiments, encoder signal correction is a more effective measure requiring less effort.

In exchange for its benefits, oversampling introduces an additional delay of half the sampling rate, because data are measured over the whole sampling period instead of sampling them at one instant. This causes a little additional phase lag, reducing the usable control gains a little.

The hardware needed for oversampling consists, at first, of high-speed analog-digital-converters that are more expensive and more sensitive to noise. Since the position and not the track signals should be averaged, the computation of (1.2) must be partly realized in hardware, requiring a larger PLD (programmable logic device) and a memory device for the arctangent lookup table. The 3rd step is a summation register in the PLD that sums up the single samples until its contents is read out by the control software.

6. Using acceleration information

6.1. Introduction

This chapter describes methods to improve the position encoder-based control system by using an additional acceleration sensor. For position feedback, the 5000-line encoder without signal correction was used.

Acceleration information can be used in a couple of ways to enhance the speed control loop. There are three possible sources for acceleration information: the Ferraris acceleration sensor, the second derivative of the position signal, and the resonant system observers (sec. 3.4). Rigid-body system observers (sec. 3.2) have not been regarded, because they had a very poor performance in the speed control loop, making it very unlikely that they can compete with other methods for acquiring acceleration information.

The Ferraris sensor's acceleration signal can be integrated, forming a speed signal that can directly be used for speed control. However, as an integrator with no feedback would drift away, an observer has to be used [50, 51, 52] that uses the speed signal acquired from deriving the position to prevent drift. A more complex observer has been discussed by Fassnacht [50], but performed weaker. Fassnacht also proposed an algorithm to switch off the observer when the acceleration exceeds the Ferraris sensor's linear range; however, since the new sensors have adjustable gain amplifiers, this was found unnecessary. Stoeppler and Douglas [52] propose a way to compensate the sensor's phase lag.

Another possibility is acceleration feedback as shown in fig. 6.1(a), with the acceleration signal acquired either from the acceleration or the position sensor [53]. This measure improves the robustness against a varying load inertia, because acceleration feedback acts as a virtual inertia that does not require a higher motor power. Acceleration control ([54], fig. 6.1(b)), meaning the insertion of a acceleration control loop inside the speed loop, is exactly the same structure except a proportional gain in the feedforward path that is compensated when designing the speed controller. Basic investigations about acceleration feedback were done by Fargouche [5].

Hori [55] proposed another alternative called "disturbance torque observer" that is shown in fig. 6.1(c). It is a simple model of the disturbance torque according

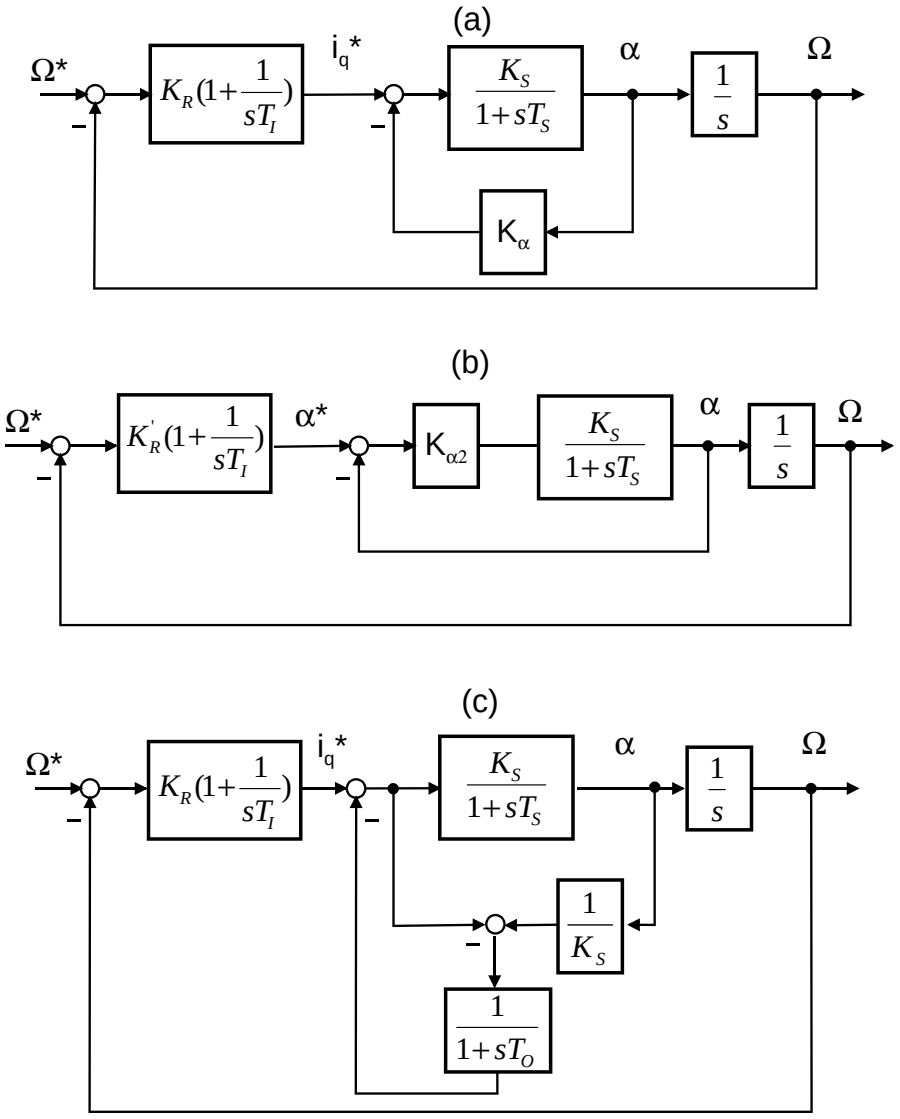


Figure 6.1.: (a) Acceleration feedback, (b) acceleration control and (c) disturbance torque observer

to the equation

$$T_{load} = J\alpha - k_T i_q \tag{6.1}$$

whose observed disturbance torque is compensated by adding an appropriate current command to the speed controller output. T_O is an arbitrary time constant for the observer. This stucture is not an observer in the Luenberger sense; instead of of the feedback constants, T_O is used for tuning.

This structure can be transformed to a PI element in the forward path and acceleration feedback with phase lag T_O and a fixed $K_\alpha = 1/K_S = 0.0038$. Acceleration feedback with delay is also regarded in sec. 6.4, and the integral element in the forward path is unnecessary because the speed controller is already of PI type.

Acceleration feedback or control is regarded as a means to improve the robustness against load inertia changes [54] or to improve disturbance rejection [55].

To damp mechanical resonance in a servo drive chain, the feedback of acceleration or torque from different points in the resonant system has been proposed [33], as well as feedback of speed differences between the inertias [38, 37]. Those methods are steps towards a state control; the model of the resonant system has to be known, an observer needs to be designed, and the feedback is designed to stabilize the model's poles. Since state control has been discussed in chapter 4, those approaches are not further regarded here.

Another idea would be the use of the acceleration sensor as an additional information source for the resonant system observers. This method was investigated by Nouira [6]. The two- or three-inertia model can be formulated as a system with two outputs speed and acceleration, with both of them compared to their measured counterpart and the differences fed back. Such observers are more difficult to design, because pole-placing is no longer a sufficient design criteria, but the observer could profit from the additional information. However, the use of an acceleration sensor and the use of a resonant system observer can be seen as two alternatives in increasing servo control performance. It is not very practical to require the additional sensor hardware *and* the exact resonant system model. Thus, this idea is not further regarded.

6.2. Acceleration sensor vs. 2nd derivative of position signal

The setup has a Ferraris acceleration sensor mounted together with the encoder (acc. sensor I). During the experiments, another sensor of the same type was mounted at the coupling between drive and load, as shown in fig. 6.2 (acc. sensor II).

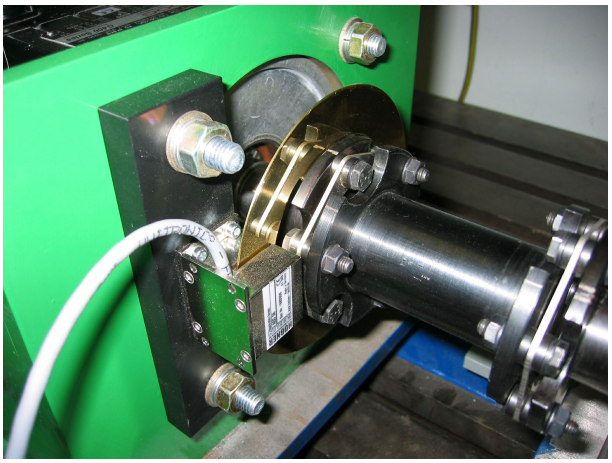


Figure 6.2.: 2nd acceleration sensor fixed on coupling

The working principle of this sensor is shown in fig. 6.3: It consists of a metal disc that rotates with the servo shaft, and one or two stationary sensor units. A sensor unit is a permanent magnet with a coil around it. The metal disc "sees" the magnet's field as a traveling one, thus eddy currents are created proportionally to the servo speed. As soon as the disc is accelerated, the eddy currents change and induce a voltage back into the coil. This voltage is used as the acceleration signal.

Because of the metal's resistance and inductance, the sensor has an intrinsic 1st order lag. The lag time can be roughly estimated as inductance divided by resistance, both as "seen" by the eddy currents. Thus, the disc should be of high-resistance material to minimize the phase lag; on the other hand, this also shrinks the signal amplitude.

The best identification method to identify the phase lag is the comparison between position encoder and acceleration sensor signal during noise excitation (see section 3.4.1). The position signal sample is derived twice using the difference quotient

$$\alpha_2(k) = \frac{\theta(k+1) - 2 * \theta(k) + \theta(k+1)}{T^2} \tag{6.2}$$

As the time values k indicate, the resulting sample has to be shifted so that the difference quotient is symmetric to the result; otherwise, additional delays will be created. The discrete Fourier transforms of α and α_2 are then plotted together in a Bode plot, where the phase lag in α_2 can easily be compensated. The compensation that matched the two graphs best gives the time constant.

The acceleration sensor used for the final measurements is a Huebner ACC94 type. A brass disk was used to get a fast acceleration signal. The sensor signal is

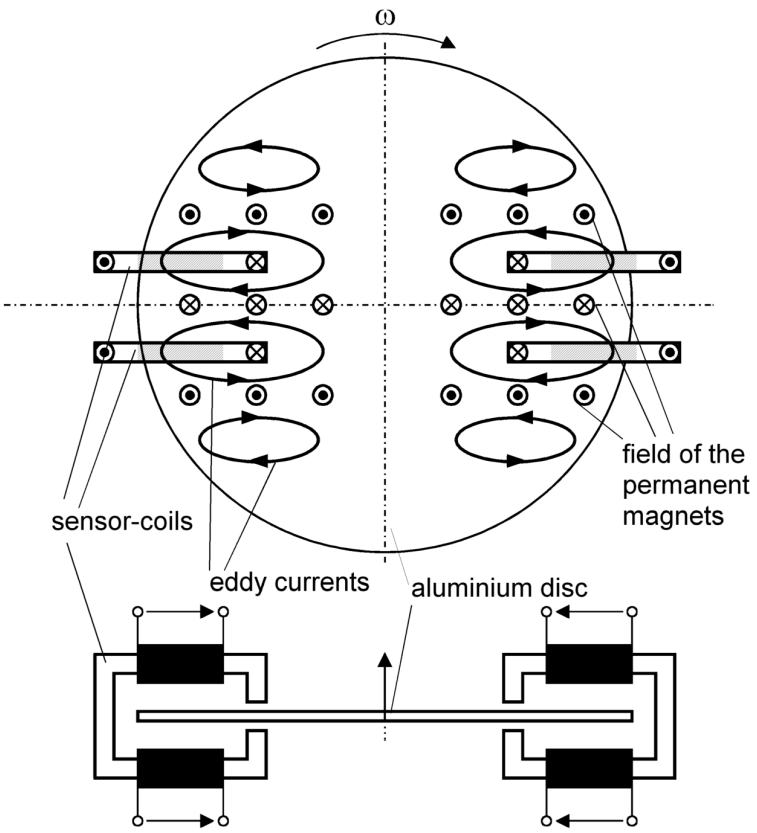


Figure 6.3.: Principle of the Ferraris acceleration sensor

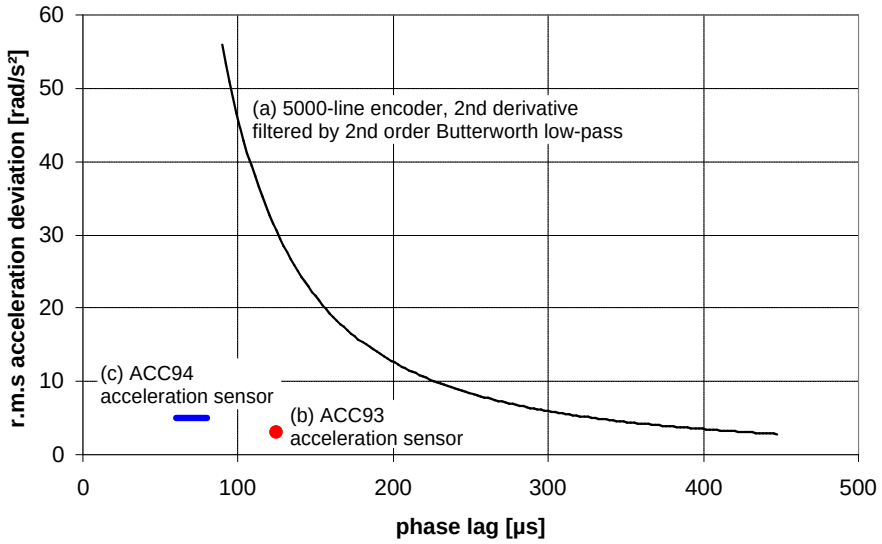


Figure 6.4.: Noise level and phase lag of acceleration signal from different sources

about $2.8\frac{\mu V}{rad/s^2}$, with a phase lag of 60 - 80μs

An imprecision of this sensor comes from the fact that the metal disc is not exactly round and centered. Thus, a component with rotation frequency (or a small multiple) occurs in the acceleration signal. Huebner company proposes to solve this problem by adding a second sensor at the opposite side of the disc, and using the average signal. However, the speed observer (see below) uses only the high-frequency part of the acceleration signal, such that this is not a problem. The disturbances caused by acceleration feedback (s. b.) are so low that they can easily be compensated by the PI speed controller, such as the disturbances from slot latching.

Another possibility of getting an acceleration signal is deriving the position signal twice. The problem in this case is that deriving amplifies noise and thus needs low-pass filtering to produce a usable signal, introducing an additional phase lag. A tradeoff between a fast and a smooth acceleration signal is necessary.

Fig. 6.4 shows the achievable noise vs. phase lag with different smoothing filters, compared to noise and phase lag of two acceleration sensors' signals. It can be seen that both acceleration sensors provide far better signals than can be achieved from the position signal.

6.3. Speed observer using acceleration sensor

The speed signal required for speed control can be acquired by integrating the acceleration sensor's output signal. However, the integrated output will drift

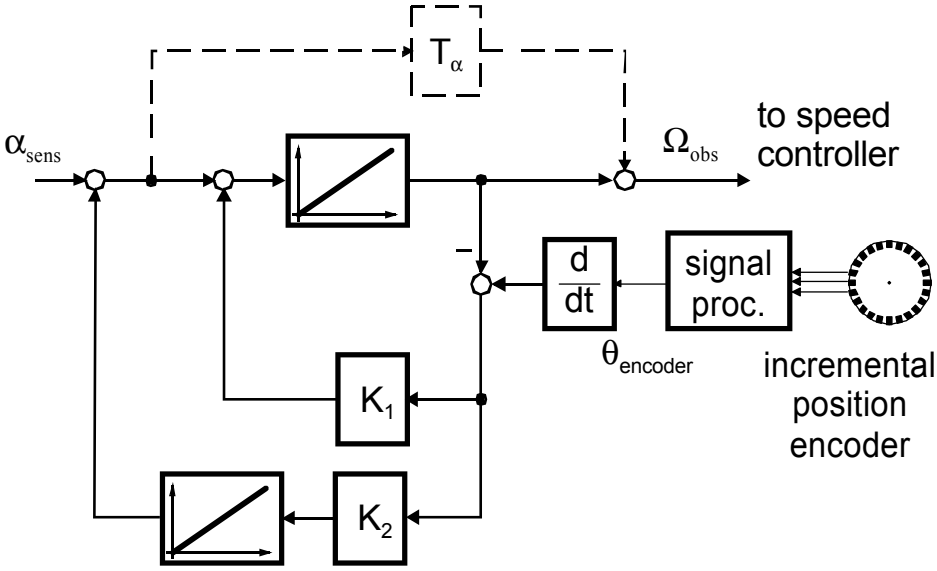


Figure 6.5.: Speed observer using acceleration signal

away from the actual speed due to disturbances. Mainly, the drifting offset of the sensor itself and its amplification electronics is critical here.

Based on the idea of integrating the acceleration, the observer as shown in fig. 6.5 (without the dashed block) has been used [37, 52]. The acceleration signal is integrated and used for control. The difference between the integrated acceleration and the derived position signal is fed back by a PI element with two arbitrary feedback constants.

To design the feedback constants, pole placement is a suitable method. As stated by Fassnacht [50], the pole locations should be both negative real, but at very different locations. This can be explained looking at the observer's transfer functions:

$$\frac{\Omega_{obs}}{\theta_{encoder}} = s\frac{s+\frac{\omega_1\omega_2}{\omega_1+\omega_2}}{(s+\omega_1)(s+\omega_2)} \tag{6.3}$$

$$\frac{\Omega_{obs}}{\alpha_{sens}} = \frac{1}{s}\frac{s^2}{(s+\omega_1)(s+\omega_2)} \tag{6.4}$$

where $-\omega_1$ and $-\omega_2$ are the pole locations. (6.3) contains the derivative s to form a speed signal from the position input. If $\omega_2 \gg \omega_1$, the numerator term in (6.3) approximately cancels the pole at $-\omega_1$. The rest is a 1st order low-pass with cutoff frequency ω_2. α_{sens} is integrated to form the speed; the rest of (6.4) is a 2nd order high-pass. It reaches an amplitude of 1 exactly at frequency ω_2 where (6.3) begins to diminish. Thus, ω_2 is a border frequency above which the integrated acceleration signal is used, while below the speed is determined mainly

from the derived position. ω_1 can be seen as the cutoff frequency below which information from the acceleration sensor is regarded as offset and thus suppressed. $\omega_1 = 2\pi 5Hz$ and $\omega_2 = 2\pi 300Hz$ were found to be optimal.

The sum transfer function $\frac{\Omega_{obs}}{s\theta_{encoder}} + \frac{s\Omega_{obs}}{\alpha_{sens}}$ equals 1, as one could expect. However, as for high frequencies the acceleration signal is used, its phase lag is found in the speed signal Ω_{obs} as well.

Fassnacht [50] proposes an algorithm for disabling the observer if the acceleration sensor's linear range is exceeded. Since the new acceleration sensors from Huebner provide amplifiers with configurable gain, the limit can be avoided, and this measure was found unnecessary. He also proposes a 4th order observer as an alternative to avoid deriving the position signal, but concludes that the results are inferior. To make the observer faster, one could think of reducing its order by setting K_2 to zero; however, in case of an offset in the acceleration signal, the speed output will have a deviation even at steady state, which is generally not desired.

[52] reported a way to compensate the acceleration sensor's phase lag completely: The forward integrator is bypassed by a proportional element, such that the numerator polynomial of the combination cancels the sensor lag. The proportional element was inserted as shown dashed in fig. 6.5); T_α is the acceleration sensor's time constant. This way, it does not change the observer structure, but is only computed afterwards. The acceleration is taken before the compensation via K_1, in order to reduce noise. The integral offset compensation via K_2, however, must be taken into account to maintain steady-state precision. The acceleration sensor's time should be underestimated rather than overestimated; $60\mu s$ were used in the experiments.

For controller design, the symmetrical optimum is not applicable because the delay of the observed speed is only the sensor lag, if it is not compensated. Thus, application of the symmetrical optimum for controller design would yield too large gains. The extended symmetrical optimum (1.9) is only applicable for high controller gains, that were achieved using resonant system observers. Thus, the damping optimum was used for controller design with the speed observer.

During the tests with the acceleration sensor mounted near the encoder, it turned out that the limiting problem is once again the 870 Hz resonance. Thus, it was tried to use the speed observer with an additional 1st order filter to damp high frequencies. In this case, the summed time constant of acceleration sensor and filter was in a range so that the Symmetrical Optimum could be applied.

6.4. Acceleration control and acceleration feedback

Acceleration feedback means feeding back the measured or observed acceleration to the current reference (fig. 6.1 (a)). In reality, the acceleration signal has an intrinsic phase lag, that is not shown in fig. 6.1 (a), and additional low-pass filtering may be used. During the experiments, it turned out that acceleration feedback without filtering did not produce sensible results using either acceleration sensor.

If acceleration feedback is counted to the "plant" of the speed controller, the plant transfer function is

$$\frac{\Omega}{i_q^*} = \frac{\frac{K_S}{1+K_S K_\alpha}(1+sT_\alpha)}{1+s\frac{T_S+T_\alpha}{1+K_S K_\alpha}+s^2\frac{T_S T_\alpha}{1+K_S K_\alpha}} \tag{6.5}$$

The plant gain is reduced by acceleration feedback (numerator!), which must be taken into account when designing the speed controller. However, it is difficult to consider the changed delay. If only the 1st order approximation of the denominator is used, controller design for $K_\alpha = 0$ is different from the design without acceleration feedback; this does not seem sensible. If the numerator is considered too, as the Taylor series suggests, the delay can become zero or even negative in some cases. Thus, a change of delay was neglected.

To examine the benefit of acceleration feedback in terms of dynamic stiffness, simulations were carried out with various combinations of 1st to 4th order filters and different feedback gains. A rigid-body model neglecting resonance was used, thus the results shown in fig. 6.6 are of general significance. The graphs show that the achieved dynamic stiffness is virtually independent of the used filter. The stiffness is improved with raising K_α up to $0.02 As^2/rad$. The steep decline for higher gains indicated that the control loop loses stability.

The explanation is that the transfer function (6.5) has a "gain peak", i. e. a frequency range where transfer gain is raised significantly above dc gain. This is much the same effect as for predictive filters (sec. 2.4) which also need to use derivatives of their input signal for prediction. The amplitude of the gain peak is approximately proportional to K_α, and has only a weaker dependence on T_α. Together with the rule that speed controller gain is raised with increasing K_α, this makes the stiffness depend only on K_α.

The best filter to use with the real setup was also determined experimentally. For each filter, K_α was raised from zero until a speed quality demand of 0.08rad/s was exceeded. It was tried to find the filter allowing the maximum acceleration feedback gain. The result with acc. sensor II was again that the filter order and cutoff frequency is not important. With acc. sensor I, only filters with low cutoff frequency can be used because of its resonant mount.

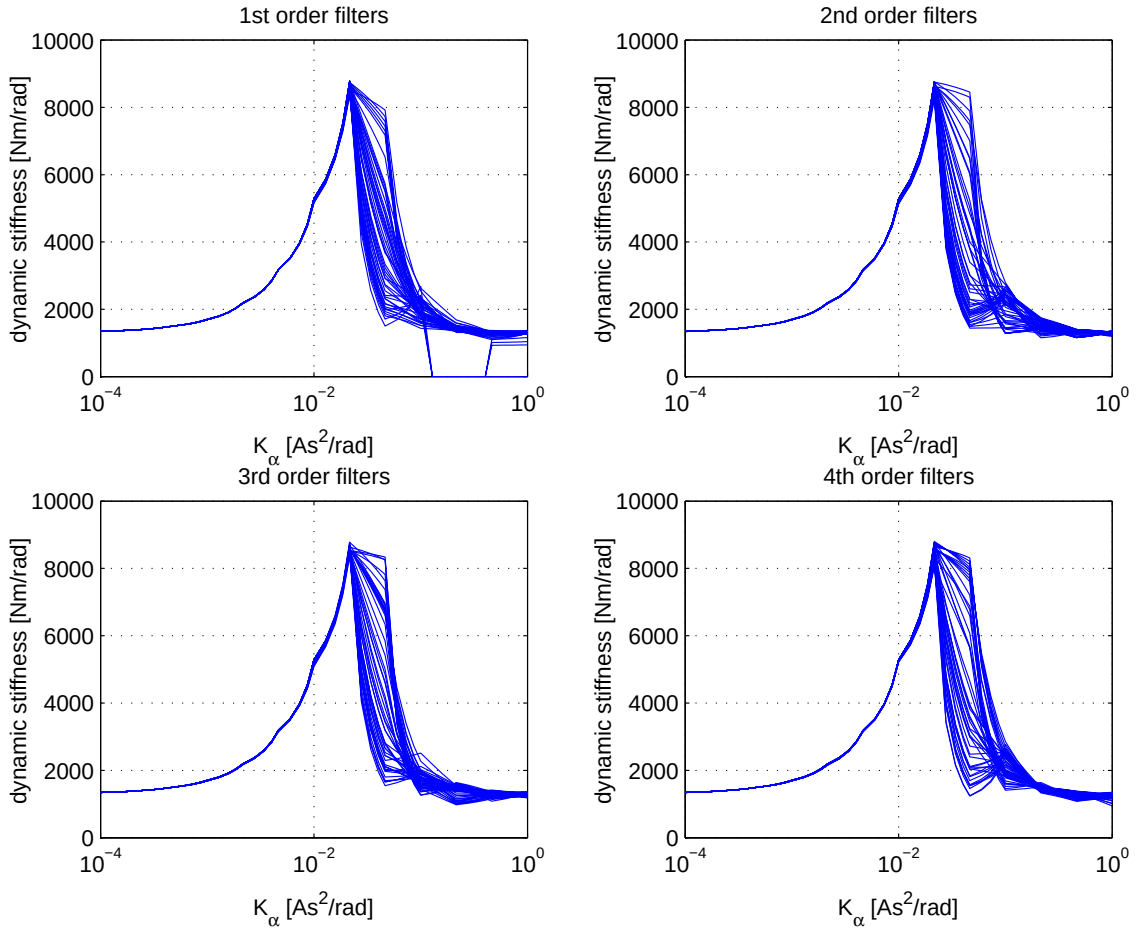


Figure 6.6.: Simulated dynamic stiffness of controls using filtered acceleration feedback

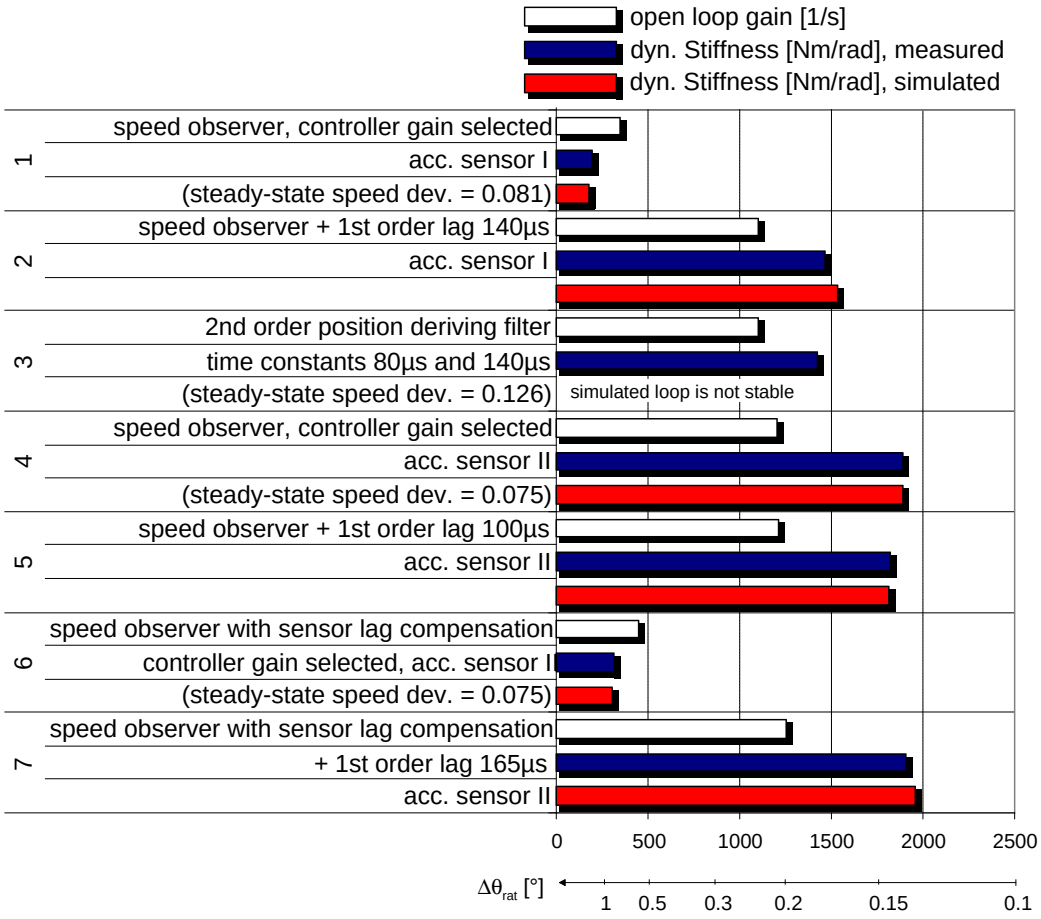


Figure 6.7.: Experimental and simulation results using speed observer

6.5. Experimental and Simulation Results

Fig. 6.7 shows the results acquired with the speed observer (sec. 6.3). In simulation, acc. sensor II was simulated as measuring the acceleration of the drive inertia, because its exact location at the coupling does not exist in the simulated 3-inertia model.

The speed observer using acc. sensor I performs very bad. This is because this sensor, which is mounted together with the encoder, is also subject to the 870 Hz resonance. Thus, the signal would need low-pass filtering which is not provided by the low lag of the ACC94. Thus, controller gain must be drastically reduced to limit control bandwidth far below 870 Hz.

In another approach, the speed observer was chained with a 1st order filter. This approach yields results comparable to those achieved with a simple 1st order filter. As a direct comparison, the a filter was designed which applies the same transfer function to the derived position (fig. 6.3 no. 3). It can be seen that the speed observer performs significantly better regarding steady-state quality

as well as dynamic stiffness. This difference was not enough to counterbalance the disadvantage resulting from the inappropriate "filter design": The two time constants form an overdamped 2nd order filter, which is certainly not ideal for the suppression of a certain frequency. In this case, an acceleration sensor with a time constant of $227\mu s$, which is the time constant of the 1st order filter with 700 Hz cutoff, would most likely have provided better results than the low-pass filter controls.

Much better results were achieved using acc. sensor II, which is mounted on the other end of the drive, with a substantially stiffer connection. Since the high-frequency part of the speed signal is provided by the acceleration sensor, it is now possible to extend the controller gain disregarding the 870 Hz resonance. The results are in the range of those achieved using low-pass + notch combinations, without knowing the resonant frequencies at all.

Yet it is not sensible to compensate the acceleration sensor's lag completely (fig. 6.3 no. 6). The configuration of servo, shaft and coupling flange still has a resonant frequency around 3000 Hz. If this resonance is not suppressed by a low-pass, it again limits the controller gain.

Fig. 6.8 shows the results achieved with acceleration feedback from either one of the acceleration sensors or the position sensor. In all cases except no. 9, a speed control with speed signal from 1st order 700Hz filter was used. The PI speed controller was designed as explained in sec. 6.4. It turned out that raising K_α from zero first improves steady-state performance, before error increases again very sharply.

Feedback without filtering may only be used with very low feedback gains, due to excitation of the resonant frequencies; this is even true for acc. sensor II. Thus, there is no measurable stiffness improvement. The feedback of filtered acceleration from acc. sensor II, however, produces equal results using 1st to 4th order filters. It is remarkably that filter order and cutoff frequency do not influence performance measurable; only very low cutoff frequencies below 100Hz are not usable for high-order filters; they yield low-frequency oscillations.

With acc. sensor I, the same acceleration feedback gain and thus an equal performance can be achieved; filters with low cutoff frequency have to be chosen because of the resonant sensor mount.

The improvement of dynamic stiffness is partially due to acceleration feedback and partially due to the rise of K_R. If acceleration feedback is used without changing the speed control parameters, fig. 6.6 looks different, and there are optimum frequencies how to chose the acceleration filter. The feedback gains are higher, but the achieved stiffness is the same.

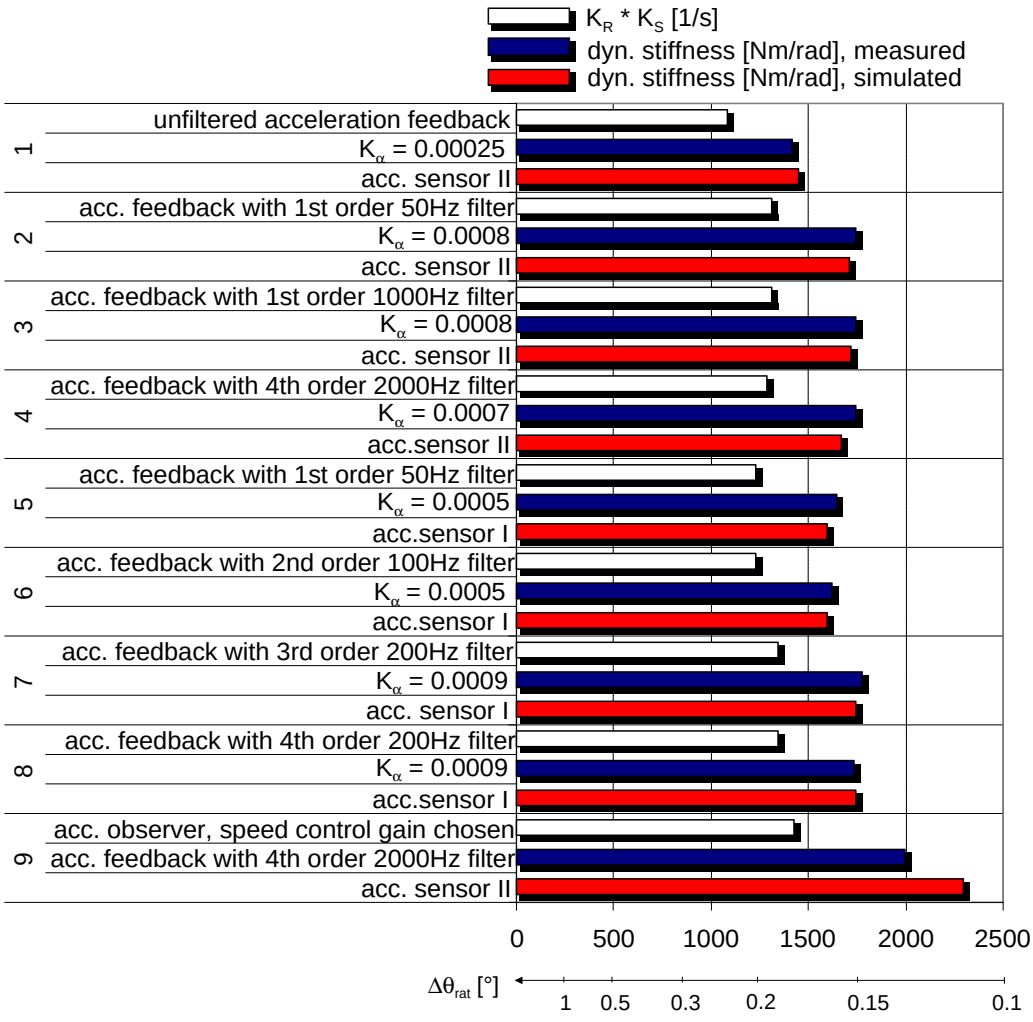


Figure 6.8.: Experimental and simulation results with acceleration feedback

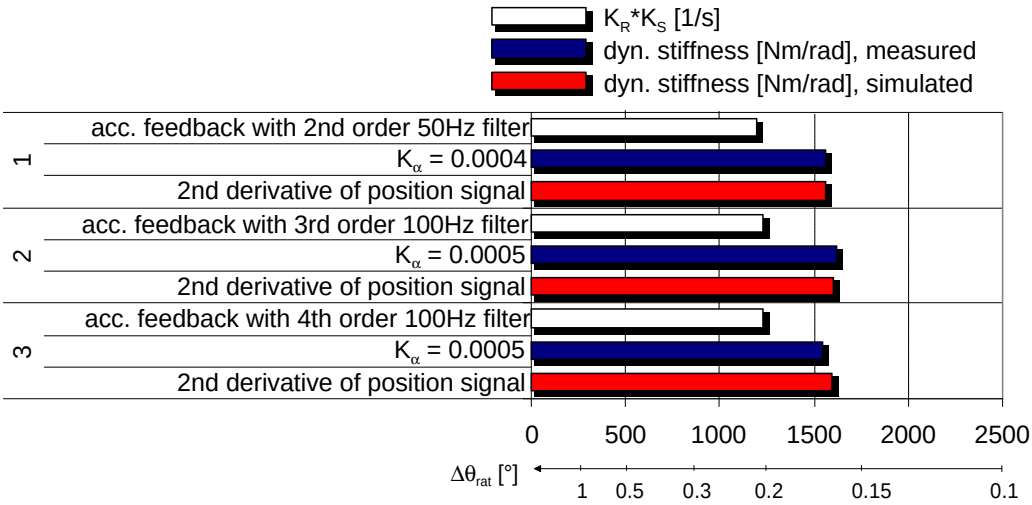


Figure 6.9.: Experimental and simulation results with acceleration feedback, using the 2nd derivative of the position signal

Fig. 6.9 shows the results using the 2nd derivative of the encoder position signal for acceleration feedback. It is possible only if filters with low cutoff frequency are chosen, because the signal suffers from a lot of differentiation noise and a resonant mount. The results are not as good as when using an acceleration sensor, even choosing filters with lower cutoff frequency is not successful.

Since both the speed observer using acc. sensor II and acceleration feedback were found to improve dynamic stiffness, it would be an idea to combine both measures (fig. 6.8 no. 9). However, the improvement compared to only using the speed observer is small. Since the reachable feedback gain is nearly independent from the used speed control, the benefit of acceleration feedback shrinks with increasing speed controller performance.

7. Load-side behavior of the experimental setup

7.1. Introduction

The three-inertia resonant system was found a valid model for the plant considered in this thesis. This suggests that the speed quality and dynamic stiffness may differ if measured at the load side. In that case, the load-side quality would be important for the customer, making the measurements shown in the previous chapters insignificant.

7.2. Load-side resonant behavior

The three-inertia resonant system is described in a linear state-space model. To examine its resonant behavior, the homogeneous solution for the system states is regarded. As equation (A.11) shows, it is a linear combination of terms depending on the eigenvalues and eigenvectors of $\mathbf{A}$.

For a conjugate complex pair of eigenvalues, the eigenvectors are also conjugate complex. The term representing the oscillation is derived in (A.11). For each system state, the absolute value of the respective eigenvector component represents the amplitude oscillation, while the complex phase of the respective eigenvector component determines the phase shift (relatively to the other states).

The component-wise absolute value of the three-inertia system's eigenvectors is shown in fig 7.1; these are fixed oscillation ratios for each resonant frequency. This means that an oscillation always occurs in all system states with a fixed ratio; if the oscillation can be damped at one place, it is reduced everywhere.

To verify the theory, measurements were carried out with different filter-based controls, with the drive-side and servo-side speed quality measured. The results are shown in fig. 7.2. It is visible that drive-side and load-side speed behave equally, except that the speed quality measured by the load-side encoder cannot fall below the encoder's imprecision level. The resonance ratio is approximately 1, though it is expected to be around 2 from fig. 7.1; this is because the load side encoder is in reality oscillating against the servo. It forms a fourth mass whose

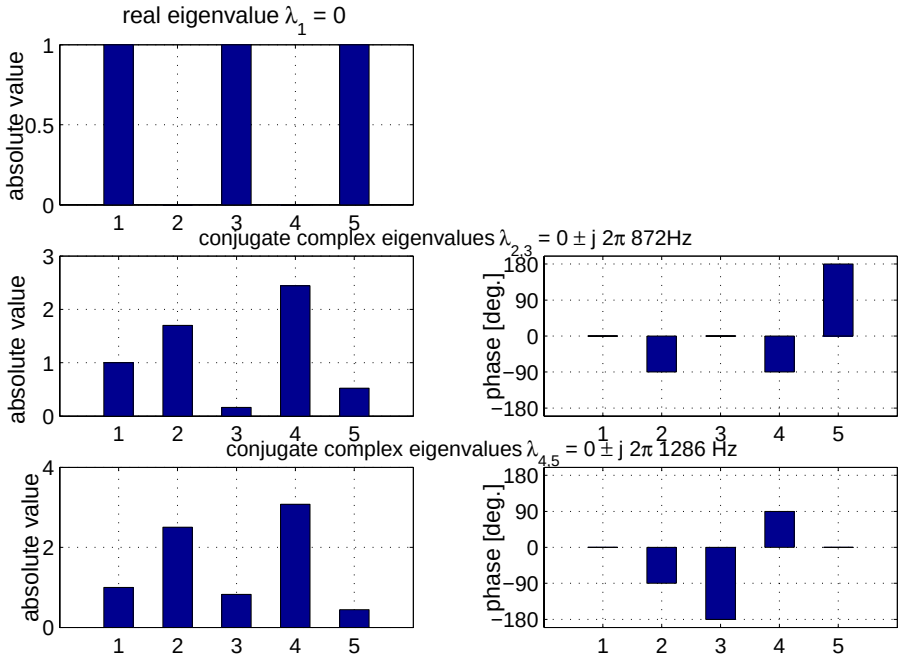


Figure 7.1.: Oscillation modes of the three-inertia system

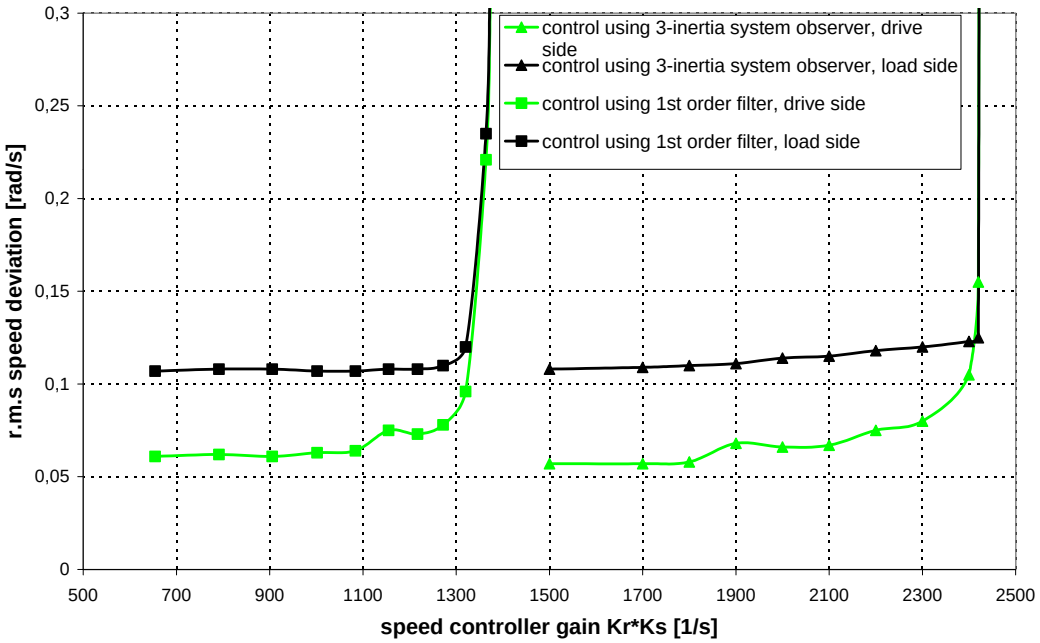


Figure 7.2.: Drive and load speed quality using different controls

oscillation amplitude is larger than the load servo's, because it is lighter.

7.3. Load-side static stiffness

As the three-inertia model already suggests, the plant has elastic couplings between the components and thus a limited static stiffness. It can be measured if one servo is position-controlled while the other makes a defined load; then, the encoders mounted on both motors can be compared. Measurements at different loads in the motor's torque range suggest a static stiffness of 12107 Nm/rad, and no measurable backlash. Though this should be equal to the spring constant c_{12} in the three-inertia model (table 3.1), it differs quite a lot. The reason is that the models were optimized to mirror the plant's resonant behavior, and not the static one; the exact plant model is of much higher order. The stiffness computed from the coupling's datasheet was even less precise.

The effect of the limited stiffness is visible in fig. 7.3. Graph (a) shows the load-side position during position control with reference position 0. The load torque changes at $t = 0$ from +50% to -50% rated torque. Though the drive is controlled to zero deviation, the load-side position has a considerable error due to the plant's limited stiffness. It is an interesting fact that though the plant regarded is a direct drive, the stationary position error is a relevant part of the dynamic deviation during load changes. In [1], this effect was neglected, thus the reported dynamic stiffness values were considerably higher.

The plant has an encoder fixed on the load servo, however, its precision is lower and the signals are considerably noisy. Thus, the best way to compute the load-side dynamic stiffness was to measure the drive-side deviation and add the stationary error. In simulation, an adapted model was used where c_{12} equals 12107 Nm/rad. This model is less precise than the original model concerning the frequency response. However, it simulates the static deviation correctly, thus the drive-side angular position could be simulated.

Lim et al. [49] proposed to compensate the steady-state load position error by feeding a delayed estimate back to the position control; the delay prevents the low-frequency resonant system from being excited; it is not necessary for the setup regarded here.

Simulations were done for a structure where the position controller and the integral part of the speed controller work on the load-side actual values. The proportional part of the speed controller must stay working on drive-side speed, because it would excite the resonance between drive and load otherwise. The result is shown in fig. 7.3(b). A three-inertia observer is used to compute the actual speed for the proportional speed controller, while integral speed controller and position controller use the directly acquired load-side speed and position. In comparison to the drive-side case, the new structure does not only compensate the steady-state position deviation, but also improves the dynamic stiffness. The

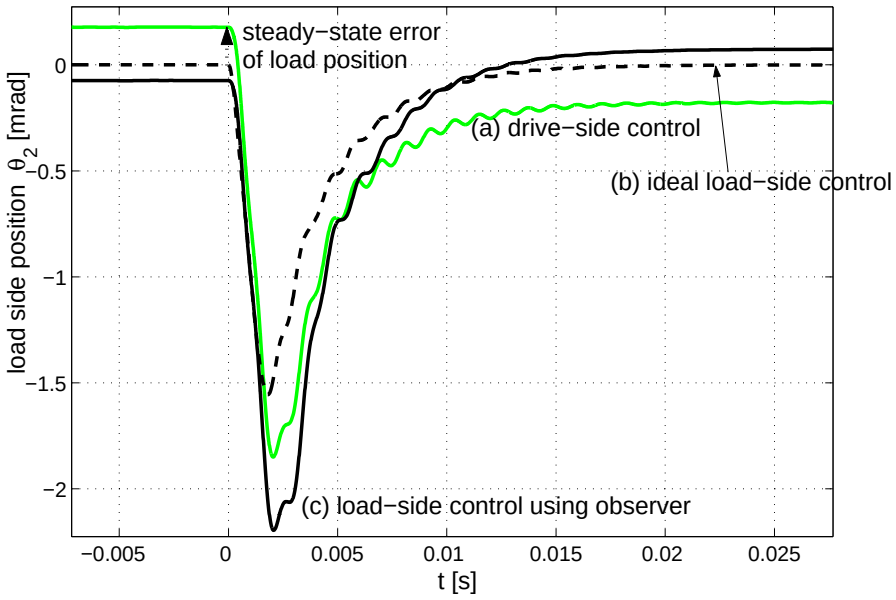


Figure 7.3.: Simulation of drive-side and load-side position control, 3-inertia observer and P/PI-cascade

reason is that a load change affects the load speed earlier than the drive speed; thus the load-side control will counteract earlier.

However, if a load-side encoder is not available, an observer must be used to determine the load-side position and speed. In fig. 7.3(c), the simulation result of a structure is shown where the speed controller's integral part uses the observed load speed, and the position controller uses

$$\hat{\theta}_2 = \theta_0 - \frac{T_{12}}{c_{12}} \tag{7.1}$$

as an actual position estimate. For c_{12}, the exact value has been used; the three-inertia observer estimates the spring torque T_{12}. It works with the model parameters shown in table 3.1, therefore the steady-state deviation cannot be exactly compensated. The dynamic performance has even been worsened by this structure. The reason is that there is a positive static deviation before the load change in case of the drive-side control, which increases the level of the whole graph, thus reducing the negative deviation. The observer is not fast enough to compensate for this disadvantage before the maximum deviation point.

The static compensation might be useful in applications where the steady-state deviation is of interest. However, as in this thesis the focus is on dynamic stiffness and the load-side encoder shall not be used for control, it is not capable of improving the control's dynamic stiffness.

8. The influence of controller timing

8.1. Introduction

In this chapter, the proposed speed acquisition and control methods are tested at a faster controller timing. Sampling time and control deadtime are varied. For position feedback, the 5000-line encoder without signal correction was used.

Some considerations exist for determining the sampling rate. Generally speaking, a too large sampling time will make high signal frequencies unreachable for the control - or even appear as alias frequencies. The sampling theorem states that if a frequency f_{signal} is to be observed correctly, the sampling frequency must be at least twice as high. It is usually proposed to use a sampling frequency around $10 * f_{\text{signal}}$.

Usual considerations are to choose a sampling frequency of 10 to 30 times the highest signal frequency that is to be considered, or 4 to 9 samples per rise time of the system [14]. Ackermann [15] proposes a numerical criterion. It is based on the system's "controllability area", meaning the set of initial state space vectors that is controllable to zero within a given time, with sampling rates of a fraction of that time, and subject to a control variable limit. This area grows with increasing sampling frequency, but is limited; the sampling frequency should be chosen where the fast growth ends.

These rules are meant as a hint for the minimum necessary sampling rate; faster sampling is generally desired. However, there are two main drawbacks of fast sampling: The noise in feedback signals is increased due to the larger system bandwidth (assuming white noise), and non-linearities become more important [15]. Besides, if derivatives need to be computed as in the speed acquisition (block (2) in fig. 1.4), low sampling times increase differentiation noise.

Another characteristic quantity for the control timing is the controller deadtime, i. e. the time between the sampling instants of input and output data. The deadtime may be chosen equal to the sampling time, or lower. This has the potential advantage of yielding a low effective deadtime in the control loop while keeping the sampling and differentiation noise. However, the effective deadtime is approximated as shown in eqn. (1.3), with the sampling time also playing a considerable part.

Table 8.1.: Pole locations for observers at $26.25\mu s/26.25\mu s$-timing

	observer poles [rad/s]
2 inertia system observer	$-3210 \pm j3830$, -3000, -3000
3 inertia system observer	$-1165 \pm j4347$, $-1390 \pm j7880$, -3000, -3000

The measurements shown in the previous chapters were done using a digital control with a timing of $100\mu s$ sampling time and $40\mu s$ deadtime. This timing was used for the position and speed control. The current controller is difficult to characterize: On one hand, it works continuously and can react immediately to a reference change. On the other hand, it does not incorporate small reference changes until the next switching instant; thus, the switching frequency of about 20 kHz can be seen as some kind of sampling frequency.

8.2. Experimental results

Fig. 8.1 shows measurements at a timing of $26.25\mu s$ sampling time and $26.25\mu s$ deadtime, which is the fastest possible for control loops in setup I. The observers used were again designed by pole placing, however, a re-design was necessary. The pole locations used are shown in table 8.1. They resemble the pole locations used at slower timing, except that the poles of the two inertia observer had to be damped stronger. This is explicable because the 2-inertia model is less precise, thus it needs a stronger feedback from the sensor to keep the estimated values close to the real ones.

The performance bars of the filter controls are in the same range as the controls at 100/40 μs. The observer controls gained a considerable amount of performance, up to a stiffness of nearly 2500 Nm/rad. This means that observer controls profit from an increasing bandwidth, while for filter controls, the advantages balance the disadvantages. The reason is that a filter's output relies completely on the input, an observer has a model which partially yields the output signal, and is not subject to differentiation noise. The resonant system observer controls investigated in this thesis all have rather precise models and weak feedback. Moreover, the observers are designed to damp the resonant frequencies, while the filter controls are intended to suppress them in the control spectrum in order not to excite them. The control deadtime is most critical when interfering with high-frequency signals.

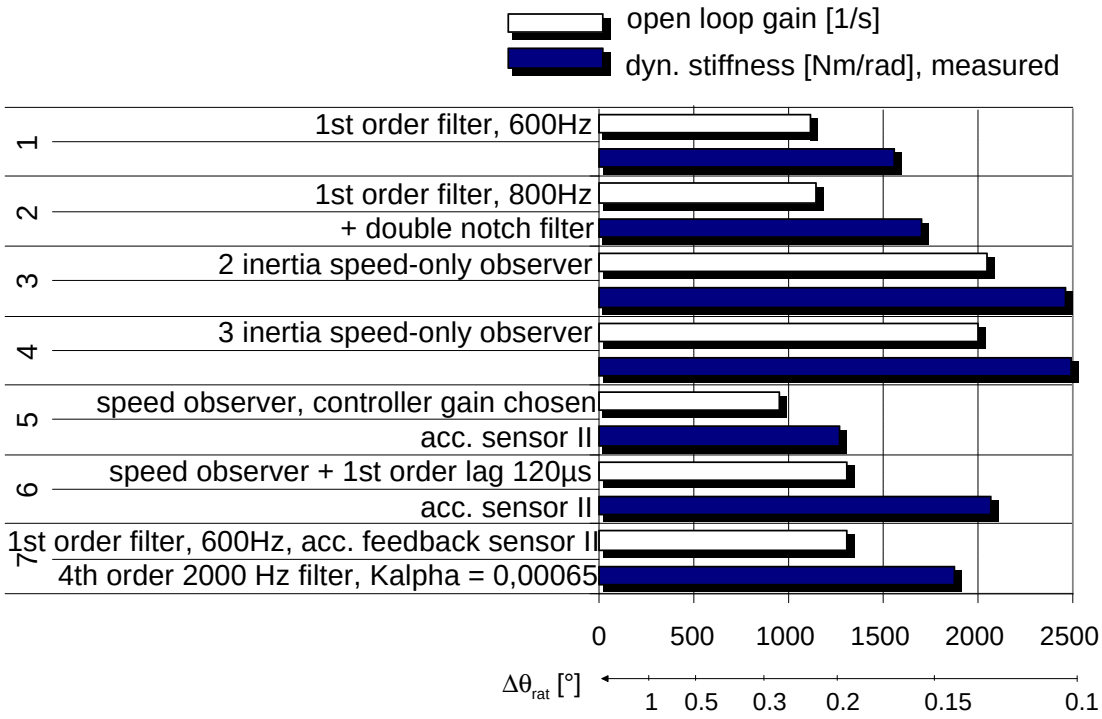


Figure 8.1.: Experimental and simulation results at $26.25\mu s/26.25\mu s$-timing

Control 6, using acceleration sensor and speed observer gained performance, because it does not rely on the derivative of position in the high-frequency range and thus is not subject to noise problems. Control 7, with acceleration feedback, performs a little weaker that at normal timing, because the main control action is done by the speed acquisition using low-pass filter and PI speed controller.

Fig. 8.2 shows, for comparison, the results achieved keeping the sampling rate at 100 μs and reducing only the control delay to 26.25 μs. Since the change, as compared to the 100/40 μs case, is rather small, the results are near the original values. The tendency is the same as in fig. 8.1: The filter controls did hardly profit from the reduction, however the observer controls improved their results. This is again because deadtime is critical for controls that deal with the resonant frequencies.

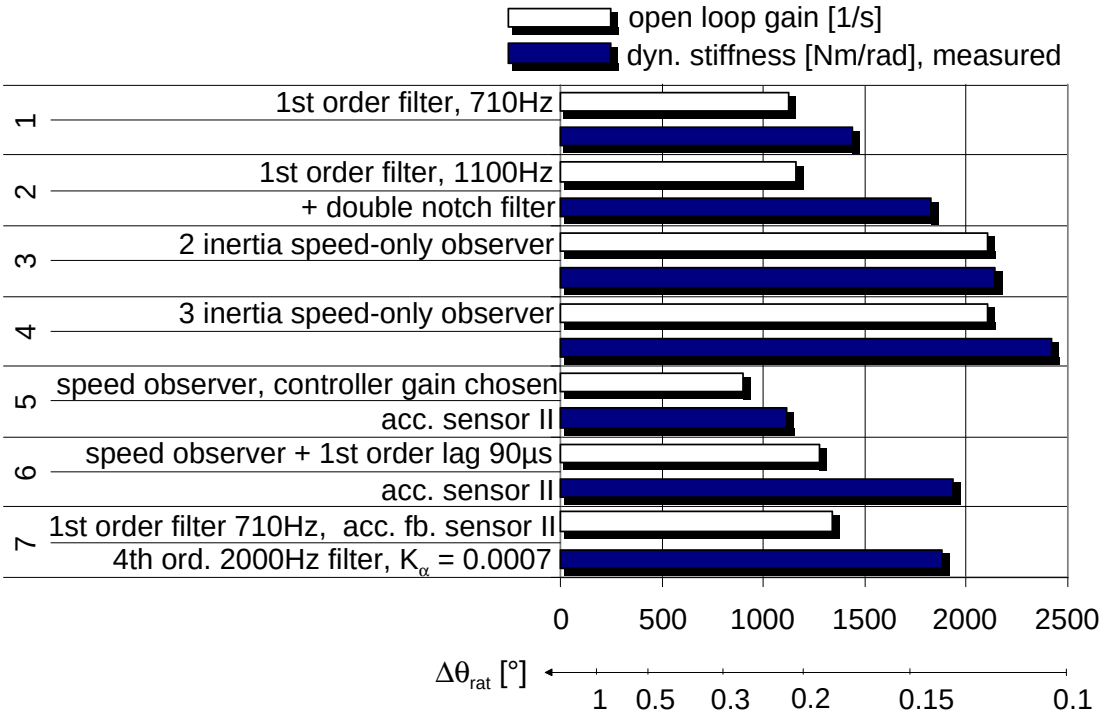


Figure 8.2.: Experimental and simulation results at $100\mu s/26.25\mu s$-timing

9. Computational Effort

9.1. Approach

Computational effort consists mainly of memory usage and computation time.

Memory usage is relevant only for table-based encoder error correction methods (see chapter 5); it can be calculated independently from the hardware. The amount of data is determined by the algorithm; the necessary resolution was determined regarding the 5000-line encoder's tables.

Computation time can only be determined for a certain processor. The PC used in the setup needs much time for I/O access, while it can perform computations very fast if the code remains in CPU cache memory. In digital signal processors (DSPs) used in industrial inverters, time is consumed mainly by executing instructions. Thus, the computation time was determined theoretically for two example DSPs: a TMS320C240 16bit fixed-point DSP with 40MHz clock [56], and the SHARC 21060L 32-bit, 66MHz floating point processor. Instead of programming the algorithms in assembly language, the operations were classified and counted in C-code, and the execution time of each operation was estimated. With the TMS320C240, fixed-point calculations were assumed; however, it was not verified whether the algorithms would actually work in fixed-point precision. The numbers of operations or cycles stated in this chapter always refer to the longest possible path, because this is the critical one in real-time control.

Interrupt latency and time for register saving and restoring were neglected.

9.2. Memory Usage

The memory usage of the correction tables is computed as amount of data times necessary resolution. The resolution can be calculated from two values: the maximum table data value and the maximum allowable error, such that the errors coming from table inaccuracy are yet considerably smaller than the precision achievable with the correction tables.

The non-parametric correction table for the Heidenhain encoder contains values up to $25 * 10^{-3}\, rad$ (with $2\pi = 1$ encoder period). The resulting angular precision can be computed from the speed precision, which is $\sigma\omega = 0.01 rad/s$ at

setup II using correction (see figure 5.5). Since discrete derivation was used to compute it, the angular deviation can be estimated as

$$\sigma\theta = \frac{\sqrt{2}}{T}\,\sigma\Omega = 3.5 * 10^{-3} rad \tag{9.1}$$

There is a factor of 7 between those values; thus, it should be sufficient to save correction table data in 8bit precision, with a resolution of 127. This results in a table size of 80kB for a 5000line-encoder.

Concerning the parametric table method, the relationship between parameter inaccuracies and position angle deviation can be computed using linearization at the working point $Ax = A_y = 1800$, $O_x = O_y = \Delta = 0$[1]. There is a factor of 1..4 between maximum parameter values and the needed parameter precision, making it possible to store the parametric table in 8bit precision, with a table size of 20kB for a 5000line-encoder. $\frac{A_y}{Ax} - 1$ should be stored to maximize precision.

9.3. Computation Time

9.3.1. Computation Time on TMS320C240

Table 9.1 shows the operations needed in the algorithms, their assembly language implementation [56], and the execution time. For a 40MHz DSP, one cycle is 50ns.

With fixed point processors it is often necessary to compute quantities in 32-bit precision. For this purpose, the TMS320F/C240-controllers, like many other 16-bit controllers, have 32-bit registers to store intermediate results. Thus, for a coarse approximation, a 32-bit computation does not take more effort than a 16-bit one, and the cycle numbers stated in table 9.1 can be used. However, it was not verified which precision would be necessary for each computation and whether the algorithm would work at all.

Encoder signal processing needs the computation of the 4-quadrant arctangent of the quotient of two values (1.2). This was assumed as suggested in [57]:

$$arctan(x) = \left\{ \begin{array}{ll} \text{5th order polynomial} & 0 \le x \le 1 \\ \frac{\pi}{2} - \arctan\left(\frac{1}{x}\right) & x > 1 \\ -\arctan(-x) & x < 0 \end{array} \right\} \tag{9.2}$$

Computation of the polynomial is done in Horner's Form, requiring 16 cycles. Additionally, the division is required and some conditional branches to figure out the quadrant. Including the division and additional conditional branches to figure

[1] The amplitude and offset are measured in count values of the 12bit converter output

Table 9.1.: Instructions of the TMS320C24x

Operation	Commands needed	Cycles
Add, Subtract	ADD or SUB, Overflow-Check, Load & Save data from/to memory	6
Multiply	MPY, Overflow-Check, Load & Save data from/to memory	6
Scalar Product of two vectors of length N	(N+1)x MPYA, 1x Overflow-Check & save data	N+6
Division	16x SUBC, Overflow-Check, Load & Save data from/to memory	21
Logical operations (AND, OR, ABS, NEG...)	AND, OR, ABS, NEG...	2
Memory move	MOVE	2
IF ... ELSE ... END-Instruction, Condition is true (denoted as *if(else)*)	SUB, BCND, B	7 + instructions
IF ... ELSE ... END-Instruction, Condition is false (denoted as *else*)	SUB, BCND	5 + instructions
IF ... END- Instruction, Condition is true (denoted as *if(end)*)	SUB, BCND	3 + instructions
IF ... END- Instruction, Condition is false (denoted as */if*)	SUB, BCND	5

out the quadrant, summing up to a maximum of 58 cycles for $atan2(y/x)$. Sine and cosine, which are needed for field-oriented control, can be realized as [57]

$$sin(x) = \left\{ \begin{array}{ll} \text{4th order polynomial} & 0 \leq x \leq \pi \\ -sin(-x) & -\pi \leq x < 0 \end{array} \right\} \tag{9.3}$$

$$cos(x) = \left\{ \begin{array}{ll} \text{5th order polynomial} & 0 \leq x \leq \pi/2 \\ -cos(\pi/2 - x) & \pi/2 < x \leq \pi \\ cos(-x) & -\pi < x < 0 \end{array} \right\} \tag{9.4}$$

requiring a maximum of 20 and 24 cycles, respectively.

A digital filter should be implemented as a series of 1st or 2nd order filters. The effort for a 2nd order IIR filter equation in direct form II [11] is 5 MAC operations or 10 cycles.

If the filter is Nth order, its implementation consists of two scalar products with N and N+1 elements (multiply & accumulate instructions can be used, one overflow check at the end), one addition and N memory move operations.

9.3.2. Computation Time on SHARC

Table 9.2 shows the operations needed on the SHARC processor.

The assembly language instructions and their execution time are discussed in [61]. All ALU operations (except the division) require only one cycle, one additional cycle was assumed for operand fetching and storing.

The SHARC features many special instructions doing multiple operations in parallel. Optimized realizations for advances functions are implemented in [62]; the computation times stated there are 82 cycles for $atan2(a/b)$ and 38 cycles for sine or cosine.

A digital filter should be realized as a cascade of 2nd order terms, even in floating-point arithmetic. The calculation time for a Nth order filter equation is $6 + 2N$ cycles [62].

9.4. Computational Effort for Control Modules

Table 9.3 shows the number of operations required for each module that is used in the control loop.

Field oriented control and space vector modulator are necessary on most industrial servo drives, but have not been used in this thesis. To take the calculation time of these modules into account, code was taken from [59] and [60], respectively, and evaluated in the same way.

Table 9.2.: Instructions of the SHARC

Operation	Commands needed	Cycles
Add, Subtract	ADD or SUB, save data	2
Multiply	MPY, , save data	2
Scalar Product of two vectors of length N	N x MAC, save data	N+1
Division	[62, atan]	9
Logical operations (AND, OR, ABS, NEG...)	AND, OR, ABS, NEG...	2
Memory move	MOVE	2
IF ... ELSE ... END-Instruction, Condition is true (denoted as *if(else)*)	SUB, BCND, B	2 + instructions
IF ... ELSE ... END-Instruction, Condition is false (denoted as *else*)	SUB, BCND	2 + instructions
IF ... END- Instruction, Condition is true (denoted as *if(end)*)	SUB, BCND	1 + instructions
IF ... END- Instruction, Condition is false (denoted as */if*)	SUB, BCND	2

Table 9.3.: Number of instructions required for control modules

	Add	Mul	Div	Scalar Prod.	Bit Op.	Move	Others
Encoder signal processing	2	2			8		1x else 1x atan2
Position + PI speed control	8	3				4	2x if(end) 1x /if
Position + Nth order state-space speed control	8 + N	3 + N				4	2x if(end) 1x /if
Speed computation using Nth order filter	N+4	4				N+1	1x if(end) 1x /if 1x filter
Speed computation using Nth order observer	2	1		N times (N+2)		N+1	–
Encoder correction using non-parametric table	4	3			3		1x else
Encoder correction using parametric table	2	2			1		–
Encoder correction using parametric table and single period table	5	5			3		1x else
Online encoder correction algorithm	16	20	3	1 + 2	1	1	1x if(end)
Field-oriented current control for induction motor	24	42	3			8	1x if(else) 8x if(end) 5x /if sin, cos
Space vector modulation	15	16	2			7	2x if(end) 3x if(else) 1x else

9.5. Results

With these numbers, the computation time of position control with several speed acquisition methods can be computed for both example processors (table 9.4). It must be compared to the timing used in the setup, which was $40\mu s$ or $26.25\mu s$ calculation time.

Table 9.4 shows that the computation time needed by the C240 to perform a minimal position control is already more than $26.25\mu s$; this controller will not be able to work at the faster timing rates discussed in chapter 8. However, most time is consumed by the standard modules that are necessary regardless of the speed acquisition method. The additional computation time required by advanced speed acquisition methods is small and will not degrade the control performance. Only the on-line encoder correction algorithm requires as much as $7.5\mu s$.

With the SHARC DSP, the computation time is smaller than those used in the experimental chapters; no performance limits will come from that issue.

The numbers shown here are a coarse estimation of the pure computation time necessary for the discussed control modules. Comparison to experience values shows that they are quite optimistic. For a more precise investigation, and to solve the question of feasibility on the fixed-point system, an implementation of the algorithms on the respective DSP would have been necessary.

Table 9.4.: Computation time for control modules and example controls

Control	Calculation time on 40MHz TMS320C240 [μs]	Calculation time on 66MHz SHARC [μs]
Encoder signal processing	2.575	1.636
P/PI cascade position control	2.025	0.455
Cascade position control with Nth order state-space speed control	$2.025 + 0.025N$	$0.424 + 0.015N$
Speed computation using Nth order filter	$1.425 + 0.3N$	$0.394 + 0.076N$
Speed computation using Nth order observer	$0.475 + 0.225N + 0.025N^2$	$0.106 + 0.045N + 0.015N^2$
Encoder error correction using non-parametric table	1.35	0.348
Encoder error correction using parametric table	0.75	0.212
Online encoder correction algorithm	7.5	1.636
Field-oriented current control for induction motor	14.175	3.985
Space vector modulation	6.675	1.470
Speed acquisition using 1st order filter, no signal correction, PI control	$27.1\mu s$	$7.9\mu s$
Speed acquisition using low-pass + double notch filter (5th order), no signal correction, PI control	$28\mu s$	$8\mu s$
3 inertia system observer (6th order), no signal correction, PI control	$29.3\mu s$	$8.5\mu s$
Speed acquisition using 1st order filter, non-parametric table, PI control	$28.5\mu s$	$8.2\mu s$
Speed acquisition using 1st order filter, parametric table, PI control	$27.9\mu s$	$8.1\mu s$
Speed acquisition using 1st order filter, on-line encoder error correction table, PI control	$34.6\mu s$	$9.5\mu s$
3 inertia system observer, on-line encoder correction, PI control	$36.8\mu s$	$10\mu s$

10. Conclusion

The thesis investigates different speed acquisition and control methods to achieve the maximum possible dynamic stiffness at a given tolerable speed error in steady-state. The methods investigated are based on different physical approaches. For the setup regarded, the limiting problem is a 870 Hz or 970 Hz mechanical resonance.

The steady-state speed error achieves good results if low-gain controls are used, with filters or observers with low cutoff frequencies. It is especially important to avoid exciting the plant's resonant frequencies (passive damping).

A high dynamic stiffness, on the other hand, is achieved using high controller gains. Since the rejection of a load step change is in all cases a rather slow action, and the position is the integral of speed, the suppression of resonant frequencies is not important (as long as the control stays stable). The filter delay at low frequencies is also a considerable issue.

Table 10.1 shows a ranking of the speed acquisition and control methods that were successfully tested. The methods are distinguished by the necessary hardware; faster timing is also a hardware requirement because it means the need of a faster DSP. The cell background indicates the need of information about the system.

Best results were achieved using resonant system observers at fast timing. Resonant system observers are rather insensitive to sensor precision issues, be it resolution, signal correction, or amplified noise due to a faster sampling rate. The results achieved with 2- and 3-inertia system observers are nearly equal, making the 2-inertia observer recommendable because the model is less difficult to identify and compute.

Notch filtering is a good method of improving the results with only coarse information about the resonant system. RLSN predictive filters perform even better, with less precision required in frequency information. The HN-IIR predictor provides many degrees of freedom and the possibility to place arbitrary notch frequencies. Thus it should be even more flexible and performant using a profound design method.

Among the two methods of using an acceleration sensor, the speed observer performs better if an optimal location is chosen for the sensor. Another advantage is that the observer design depends only on the two sensors, not on the system's

Table 10.1.: Ranking of speed acquisition and control methods

Achieved dynamic stiffness around	Methods using ... (hardware)			
	2048-line encoder	5000-line encoder	Acceleration sensor	5000-line encoder and fast timing
2500 Nm/rad				resonant system observers
2250 Nm/rad		resonant system observers RLSN predictive filters		
2000 Nm/rad	resonant system observers	Low-pass + notch filters	Acc. observer *and* acc. feedback	Acceleration observer Acceleration feedback
1750 Nm/rad		Low-pass filters + encoder signal correction	Acceleration observer Acceleration feedback	Low-pass + notch filters
1500 Nm/rad	Low-pass + notch filters	Low-pass filters		Low-pass filters
1250 Nm/rad				
1000 Nm/rad	Low-pass filters			

Information required about the system's resonant frequencies

Precise system identification required

resonant behavior. Concerning acceleration feedback (or acceleration control or disturbance torque observer, which are equivalent), an interesting point was that the delay in the acceleration feedback path does not influence performance within a wide range. This makes it possible to use any location for the acceleration sensor, since a low-pass filter can be used to cut off the resonant frequencies.

The encoder and acceleration sensor mount as used with this setup (fig. 1.2) cannot be recommended because it is a major source of resonance. Two alternatives exist: On the one hand, combined position and acceleration sensors in one housing have been announced. On the other hand, the acceleration sensor's advantage may be used that it is easy to attach to different points in the drive chain; this way, it may yield a speed signal that is optimal concerning mechanical resonance.

When designing an acceleration sensor, a trade-off must be found between a low delay, requiring small poles and a high-resistance disc, and a sufficient signal voltage to keep the noise low in relation. For use in the speed observer, the optimal acceleration sensor delay would be a delay that is sufficiently high to function as the low-pass filter in the speed control loop. This was around $150\mu s$ for the setup regarded here. For using acceleration feedback, the delay is not important. However, adding low-pass characteristic is much easier than compensating delay, thus it seems sensible to manufacture faster sensors.

Rigid-body observers did not show any advantage in this thesis, because they did yield non-optimal frequency characteristic. In literature, improvements due to rigid-body observers have been reported with incremental position encoders (without continuous interpolation) and for high-precision positioning.

Correction of encoder errors, from the results of this thesis, yields only a low improvement with simple speed acquisition methods, and none in connection with observers. There is an advantage concerning precision positioning or measurement, and the case may also be different for low-resolution encoders. In any case, encoder correction is useless when the encoder errors consist of noise. The performance differences between the investigated encoder correction methods are small.

The comparison of results with different sensors revealed that a control using filters is strongly dependent on the encoder signal quality. Using an observer, the performance is quite independent from the signal quality; therefore, cheaper sensors can be used. The improvement from 2048-line to 5000-line encoder yielded equal results as adding notch filters; 5000-line encoder together with notch filtering is equipollent to the best controls achieved using the 2048-line encoder.

At a faster control timing, the gain that may used with respect to the steady-state performance requirements is approximately the same. Anyway, a significantly better dynamic stiffness is possible.

Active damping of the resonant frequency using a state controller is possible but does not improve performance in terms of dynamic stiffness; thus, it is only to be applied in special cases.

As the resonant system model states, there are fixed resonance ratios between all state variables of the system. Thus, the speed quality at drive and load side is proportional. The limited static stiffness of the system plays a surprising role in limiting the dynamic stiffness, however a sensible compensation is only possible if a load-side encoder is available.

A coarse assessment of the required computation time shows that, on a 16-bit fixed-point DSP, computational time is a critical issue. However, most time is needed to compute the basic modules field-oriented control, modulation, and encoder signal processing. Of the algorithms investigated in this thesis, the on-line encoder correction needs most computation time by far.

A. State-space theory and state control

A.1. Introduction

This chapter describes basics of state-space theory and state control, as they are used in this thesis, and defines symbols. The equations are taken from [7] and [8].

A system with s single input and single output can be described as a transfer function

$$\begin{aligned} Y(s) &= \frac{B(s)}{A(s)}X(s) \\ B(s) &= b_0 + b_1 s + b_2 s^2 + \cdots + b_{m-1}s^{m-1} + b_m s^m \\ A(s) &= a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1}s^{n-1} + a_n s^n \\ m &\leq n \end{aligned} \tag{A.1}$$

where N(s) and D(s) are polynomials in s, X(s) represents the Laplace transform of the input signal, and Y(s) the Laplace transform of the output.

A more general way to describe a system is the state-space form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{A.2}$$

$\mathbf{x}$ is the vector of state variables, $\mathbf{u}$ represents the inputs, and $\mathbf{y}$ the outputs. This form has the advantage that information about the behavior of interior system states is available, making advanced control concepts possible.

The eigenvalues of the matrix $\mathbf{A}$ have the same meaning as the poles of the transfer function. The transfer from a transfer function to state-space equations can be done in many ways, since a state-space system contains more information. One way is the detectability normal form (see next section), where the transfer

function (A.1) equals the system

$$\dot{\mathbf{x}} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & 0 & & 0 & -a_1 \\ 0 & 1 & 0 & & 0 & -a_1 \\ \vdots & \vdots & \ddots & & 0 & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -a_{n-1} \end{pmatrix} \mathbf{x} + \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \\ 0 \\ \vdots \\ 0 \end{pmatrix} u \tag{A.3}$$

$$y = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \mathbf{x} \tag{A.4}$$

To calculate the transfer function of a SISO (single input single output) state-space system, the straight-forward equation is

$$\frac{N(s)}{D(s)} = \mathbf{C}\left(s\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} \tag{A.5}$$

A.2. Controllability and detectability

A state-space system is called fully controllable if every system state $\mathbf{x}_1$ can be reached from every initial state $\mathbf{x}_0$ by applying a function to the input(s) $\mathbf{u}$ for a finite time interval. This can be tested by computing the controllability matrix

$$\mathbf{M}_C = \begin{pmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{pmatrix} \tag{A.6}$$

If $\mathbf{M}_C$ has a rank of at least n, then the system is fully controllable.

A system is called detectable if every initial system state $\mathbf{x}_0$ can be reconstructed by observing the outputs for a finite time interval. The detectability matrix is

$$\mathbf{M}_D = \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{pmatrix} \tag{A.7}$$

The matrix $\mathbf{M}_D$ of a fully observable system has a rank of at least n.

A.3. System behavior in time domain

If the input variables are set to zero, the system is described by the differential equation system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{A.8}$$

The mathematical solution of these equations depends on the eigenvalues and -vectors of the matrix $\mathbf{A}$. If λ_i are n different eigenvalues and $\mathbf{v}_i$ the respective eigenvectors, then the solution is

$$\mathbf{x}(t) = \sum_{i=1}^{n} c_i \mathbf{v}_i e^{\lambda_i t} \tag{A.9}$$

(c_i are integration constants that must be chosen such that the initial conditions are satisfied)

This solution also applies for complex eigenvalues. For a conjugate complex eigenvalue pair $-\alpha \pm j\beta$ with respective eigenvectors $\mathbf{a} \pm j\mathbf{b}$, the integration constants must be also conjugate complex (or real and equal) to ensure that the solution is real. With integration constants $Ce^{\pm\gamma}$, (A.9) turns to

$$\mathbf{x}(t) = 2Ce^{-\alpha t}\left(\mathbf{a}cos(\beta t - \gamma) - \mathbf{b}sin(\beta t + \gamma)\right) \tag{A.10}$$

A more general solution, taking the inputs into account, is [7]

$$\mathbf{x}(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau + e^{\mathbf{A}t}\mathbf{x}(0) \tag{A.11}$$

where the matrix exponential $e^{\mathbf{M}}$ is defined as the result of applying the exponential function's Taylor series to the quadratic matrix $\mathbf{M}$.

A.4. State controller design

Fig. A.1 shows the block diagram of a state control system. All states are proportionally fed back to the input(s). The coefficients are stored in the vector K. The closed control loop has the system matrix

$$\mathbf{A}_R = \mathbf{A} - \mathbf{B}\mathbf{K} \tag{A.12}$$

It is the most basic demand that the state controller should stabilize the system. This can be guaranteed by the design method of pole placing: Given a

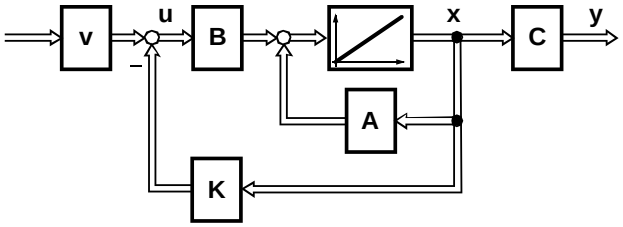


Figure A.1.: Block diagram of a state control system

set of desired poles, design K such that the poles of the closed loop, meaning the eigenvalues of $\mathbf{A}_R$, equal the desired poles. This is possible for every controllable system, if the poles are real or conjugate-complex pairs.

For a single-input system, there is a unique feedback vector K for every set of desired poles. The equation of Ackermann [16, acker] is a popular method of computing the feedback vector.

With a multi-input system, the solution of the pole-placing problem has additional degrees of freedom. Advanced pole placing algorithms [16, place] use this freedom to optimize the general robustness of the system.

Regarding a control system's step response, quality is often evaluated by overshoot and settling time. Overshoot *ovr* means the percentage of the step height that the actual value can raise above the new reference value. For a pole pair $a \pm jb$ that is considered dominant, overshoot can be approximated as

$$ovr = e^{-\frac{a}{b}\pi} \tag{A.13}$$

The settling time T_set that it takes until the actual value is kept in a range of, for example, 5% around the new reference value, can be approximated as

$$T_{set} = \frac{3}{a} \tag{A.14}$$

An alternative method to design the feedback vector is the linear-quadratic control. The approach is a cost functional

$$J = \int_0^\infty \left(\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}\right) dt \tag{A.15}$$

where u(t) and $\mathbf{x}$(t) are the inputs and system states occurring when the system approaches zero state starting from any initial state $\mathbf{x}_0$. For every pair of positive definite $\mathbf{Q}$ and $\mathbf{R}$, there is a unique solution $\mathbf{K}$, for single-input as well as multi-input systems. It is computed by [16, lqr].

$\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices. Large values in $\mathbf{R}$ indicate that control energy should be minimized, requiring a slow control. Large values in $\mathbf{Q}$ demand

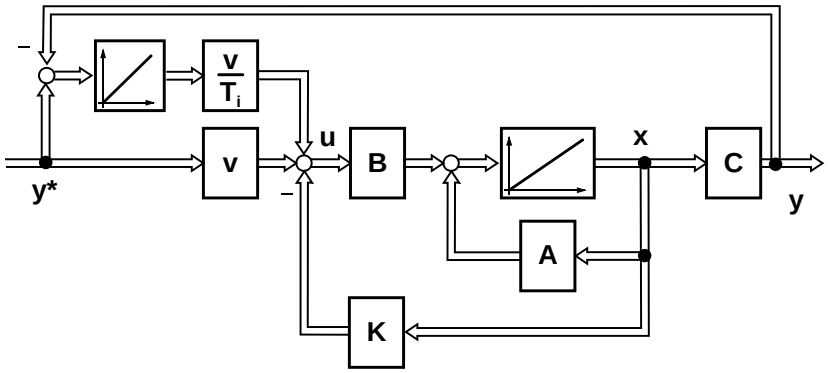


Figure A.2.: Block diagram of a PI state control system

a fast control system to force the system states to zero rapidly. The solution does not change if both $\mathbf{Q}$ and $\mathbf{R}$ are scaled by the same factor.

The reference value is fed into the state control system by a proportional prefilter [7]. If the number of inputs equals the number of outputs, the prefilter constant or matrix $\mathbf{V}$ can be determined such that the system output(s) will track the reference inputs in steady state:

$$\mathbf{V} = (\mathbf{C}(-\mathbf{A} + \mathbf{BK})^{-1}\mathbf{B})^{-1} \tag{A.16}$$

The prefilter constant is the equivalent to the proportional gain of a PI speed controller K_R (fig. 1.4).

A state controller uses only proportional gains. Thus, steady-state deviations will occur if disturbances exist. A solution for this problem is the addition of an integral gain in parallel to the prefilter, which will work like the integral part of a PI controller. Fig. A.2 shows a block diagram of this structure. The input to the system is the reference value for the quantities in the output vector. The prefilter $\mathbf{v}$, which is a scalar gain for single-input-single-output systems, has a similar meaning as the proportional gain in a PI control loop. Thus, similar methods may be used to design the integral time constant T_I.

A.5. Observers

A.5.1. The complete observer

For a state controller, all states need to be known. However, measuring them is often impossible or not desired. In such a case an observer is used, as shown in fig. A.3. An observer is a plant model that is calculated in real time, receiving the same input signals as the real plant. The model parameters and the estimated

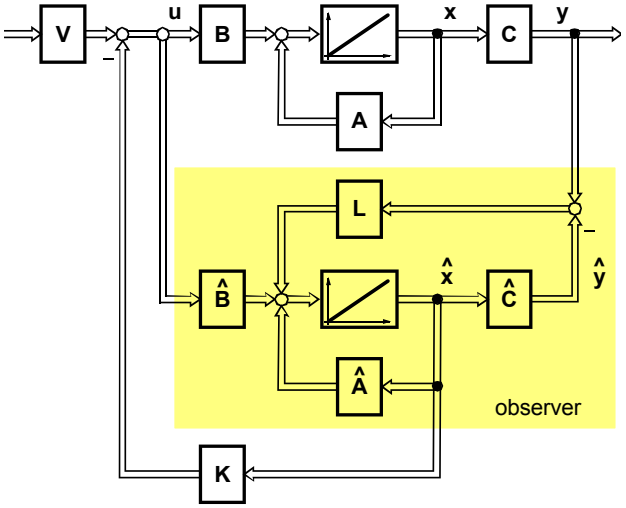


Figure A.3.: State control with observer

state vector are marked with a hat, pointing out that they are only estimated and may differ from the real values. The measurable plant outputs are compared to their modeled counterpart, and the difference is fed back into the observer with feedback gain matrix $\mathbf{L}$. The observer's system states are all known, so that they can be used to feed the state controller.

A.5.2. The reduced observer

Suppose that the state vector can be split up as

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \tag{A.17}$$

where the states contained in $\mathbf{x}_1$ are measured. Thus, there is no need to observe those quantities; the observer can be reduced to observe the states in $\mathbf{x}_2$.

The system equations (A.2) then turn to[1]

$$\begin{aligned} \begin{pmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{pmatrix} &= \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix} \mathbf{u} \\ \mathbf{y} &= \mathbf{x}_1 \end{aligned} \tag{A.18}$$

The goal is to design an observer for the unknown states $\mathbf{x}_2$, while $\mathbf{u}$ and $\mathbf{x}_1$ are known. The problem is that the unknown quantity $\dot{\mathbf{x}}_1$ is part of the equations. This may be solved by a linear transformation of the states

$$\tilde{\mathbf{x}}_2 = \mathbf{x}_2 - \mathbf{L}\mathbf{x}_2 \tag{A.19}$$

[1] Again, all plant parameters that are used in the model are only estimated values. The hats are left out in the following to keep the equations as simple as possible

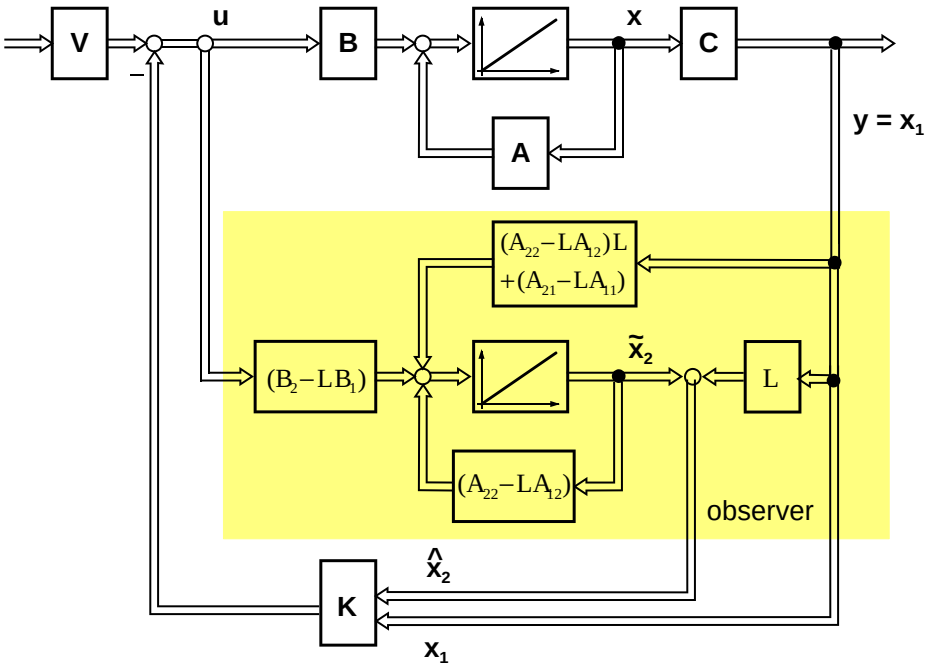


Figure A.4.: State control with reduced observer

using feedback matrix $\mathbf{L}$. The resulting system equations for $\tilde{\mathbf{x}}_2$ are

$$\begin{aligned}
\dot{\tilde{\mathbf{x}}}_2 &= \dot{\mathbf{x}}_2 - \mathbf{L}\dot{\mathbf{x}}_1 \\
\dot{\tilde{\mathbf{x}}}_2 &= \mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} - \mathbf{L}\left(\mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2 + \mathbf{B}_1\mathbf{u}\right) \\
\dot{\tilde{\mathbf{x}}}_2 &= \left(\mathbf{A}_{22} - \mathbf{L}\mathbf{A}_{12}\right)\tilde{\mathbf{x}}_2 + \left(\mathbf{A}_{22} - \mathbf{L}\mathbf{A}_{12}\right)\mathbf{L}\mathbf{x}_1 + \left(\mathbf{A}_{21} - \mathbf{L}\mathbf{A}_{11}\right)\mathbf{x}_1 \qquad \text{(A.20)} \\
&+ \left(\mathbf{B}_2 - \mathbf{L}\mathbf{B}_1\right)\mathbf{u} \qquad \text{(A.21)}
\end{aligned}$$

The block diagram of the observer defined by (A.20) is shown in fig. A.4. The observer feedback is provided by the terms containing the feedback matrix $\mathbf{L}$. For state control, the measured states $\mathbf{x}_1$ and the estimated states $\hat{\mathbf{x}}_2 = \tilde{\mathbf{x}}_2 + \mathbf{L}\mathbf{x}_2$ are used.

A.5.3. Observer feedback design

Similarly to the state controller, the feedback matrix $\mathbf{L}$ must be chosen, being responsible for stability and performance of the observer. Pole placing is again an appropriate method. If the poles are placed far left from the poles of the closed control loop, state controller and observer may be designed independently; otherwise, they form an joint system. It is very problematic if the observer model does not match the plant: in such a case, the poles of the system may move rapidly towards the instable region [37].

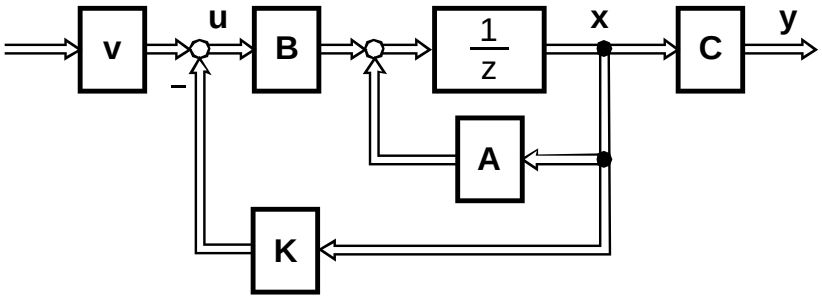


Figure A.5.: Block diagram of a discrete-time state control system

An observer may also be designed using the linear-quadratic method; it is then called a "static Kalman filter". There is a different interpretation of the design matrices $\mathbf{Q}$ and $\mathbf{R}$: The Kalman filter is derived from the idea of a state-space system that is disturbed by while noise, expressed as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{v} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{w} \end{aligned} \tag{A.22}$$

where $\mathbf{v}$ and $\mathbf{w}$ are vectors of white noise processes. If $\mathbf{Q}$ is the covariance matrix of $\mathbf{v}$ and $\mathbf{R}$ is the covariance matrix of $\mathbf{w}$, then the resulting static Kalman filter is the observer that will produce the best possible state estimation. Large values in $\mathbf{Q}$ indicate that there is a lot of disturbance in the model, and the observer should rely more on the sensors. This requires large feedback constants. On the other hand, large values in $\mathbf{R}$ indicate that the sensors used are quite imprecise or noisy; thus, the observer should rely on the system model, with only a weak feedback.

A.6. Discrete-time systems

Discrete-time systems can as well be expressed in state-space. All equations can be applied when s is replaced by z and $\mathbf{x}(k+1)$ is written instead of $\dot{\mathbf{x}}$.

Transformation from and to a transfer function works with the same equations as for the continuous-time system - of course, the resulting system is also time-discrete. Controllability and detectability are defined in the same way, and may be verified using the same equations.

The equation for the prefilter constant is slightly different:

$$\mathbf{V} = (\mathbf{C}(\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K})^{-1}\mathbf{B})^{-1} \tag{A.23}$$

A time-continuous system is transformed to a time-discrete one using the equations

$$\mathbf{A}_z = e^{\mathbf{A}_s T} \tag{A.24}$$

$$\mathbf{B}_z = \int_0^T e^{\mathbf{A}_s (T-\tau)} \mathbf{B} d\tau \tag{A.25}$$

$$\mathbf{C}_z = \mathbf{C}_s \tag{A.26}$$

where indices s and z denote the time-continuous and time-discrete representation, respectively, and T is the sampling time. The equations are the same as (A.11).

Pole placing can be done in the same way, except that the pole locations are different: If a pole of a continuous system is located at s_0, then the equivalent pole location of a time-continuous system is

$$z_0 = e^{s_0 T} \tag{A.27}$$

Thus, the area of stable pole locations, which was $Re(s_0) < 0$ for time-continuous systems, is $|z_0| < 1$ in time-discrete domain. For the zeros of the transfer function, there is no such simple equation.

In this thesis, the s-domain pole locations and s-domain block diagrams are given for all observers and state controllers, because they are more meaningful to many engineers, including the author. Design was done in z-domain using the transformed system model ($\mathbf{A}_z$, $\mathbf{B}_z$, $\mathbf{C}_z$) and desired pole locations computed with (A.27).

Linear-quadratic controller and observer designs exist as well in discrete-time domain, however the design matrices need to be changed [16, kalmd]:

$$\mathbf{Q}_z = \int_0^T e^{\mathbf{A}_s \tau} \mathbf{Q} e^{\mathbf{A}_s^T \tau} d\tau \tag{A.28}$$

$$\mathbf{R}_z = \frac{\mathbf{R}_s}{T} \tag{A.29}$$

Discrete-time observers may implement a prediction of the observed states one step ahead. For an explanation, consider fig. A.5 as the underlying model of an observer. At time k, all information is available to compute $\hat{\mathbf{x}}(k+1)$, which is located at the input of the unit delay block. The use of the predicted states for control improves control system performance, especially with resonant system observers where phase lag is very critical.

B. Optimization Algorithms

B.1. Algebraic Optimization: The Method of Lagrange Multipliers

Let f be a scalar function in a number of variables

$$f(\mathbf{x}) = f(x_1, x_2, ...x_n) \tag{B.1}$$

$\mathbf{x}$ is called the parameter vector, f is the target function. The goal is to find the parameter values where the global minimum of f occurs[1].

The optimization problem has an algebraic solution: set the n derivatives to zero

$$\begin{aligned} \frac{\delta f}{\delta x_1} &= 0 \\ \frac{\delta f}{\delta x_2} &= 0 \\ &\vdots \\ \frac{\delta f}{\delta x_n} &= 0 \end{aligned} \tag{B.2}$$

(B.3)

If all equations are true at a point $\mathbf{x}$, this is a necessary condition for a local minimum or maximum. Comparing the function values at those points will yield the global minimum.

The method of Lagrange multipliers applies when the solution is constrained by conditions of the type

$$g_i(\mathbf{x}) = 0 \tag{B.4}$$

[1] Optimization algorithms are usually defined to search for the minimum; a maximum may be found e. g. by searching for the minimum of $1/f$ or $-f$

In this case, necessary conditions for local extrema are

$$\frac{\delta f}{\delta x_1} + \lambda_1 * \frac{\delta g_1}{\delta x_1} + \ldots + \lambda_m * \frac{\delta g_m}{\delta x_1} = 0 \tag{B.5}$$

$$\frac{\delta f}{\delta x_2} + \lambda_1 * \frac{\delta g_1}{\delta x_2} + \ldots + \lambda_m * \frac{\delta g_m}{\delta x_2} = 0$$

$$\vdots$$

$$\frac{\delta f}{\delta x_n} + \lambda_1 * \frac{\delta g_1}{\delta x_n} + \ldots + \lambda_m * \frac{\delta g_m}{\delta x_n} = 0 \tag{B.6}$$

The Lagrange multipliers λ_i are part of the equation system, however their solution values are not needed.

Algebraically, only local extrema can be found. This is problematic since normally the global minimum or maximum is desired. If it is not ensured by the function type that only one extremum exists, there is no easy way of finding the global solution.

B.2. The Simplex algorithm

A class of optimization algorithms is called "gradient methods". Those algorithms start at a point in the parameter space $\mathbf{x}_0$, then determines the target function value. Further, the direction of the steepest descend of the target function is determined by computing the gradient ∇f. The algorithm picks a new parameter set located in this direction and continues iteratively.

Computing the gradient is often a problem, especially if discontinuities occur in the target function. To overcome this problem, the Simplex algorithm [16, fminsearch] picks a set of points in the parameter space, compares the target function values, and replaces points with high values by new points located in the opposite direction. The points form a shape called "Simplex", which is kept throughout the algorithm. The algorithm ends when the diameter of the Simplex and/or the difference of target function values over the Simplex are lower than specified borders.

The Simplex algorithm is more robust than gradient algorithms, however it is generally slower. It shares the most important disadvantage of gradient algorithms: The algorithm will approach the nearest local minimum from its starting point, without recognizing that there may be other minima. Thus, the solution may not be the global minimum, and may depend on the initial parameter set. It is very important to use a sensible initial parameter set, and to try different of them to at least recognize that there are local minima.

B.3. Evolutionary Algorithms

Evolutionary Algorithms [18] are mathematical models of the evolutionary process in nature. They start with a group of "individuals", i. e. parameter sets. The steps during one iteration are

- Compute the target function value of each individual. Since there is no connection between the individuals, this step is very suitable for parallel computing.
- Select the individuals with the best target function values. (selection)
- Combine the parameter values of those individuals to form new individuals (recombination)
- Perform small random modifications on the individuals to have the chance to acquire new features. (mutation)

Plenty of methods exist for selection, recombination, and mutation. It is also possible to divide the individuals into groups that use different methods.

Evolutionary algorithms are very robust methods in finding the global minimum because they compare multiple parameter sets that spread the whole parameter space. When using them for optimizing the feedback gain of the on-line encoder error correction method, it was found that the exact choice of methods is not of much importance for finding optimal results. A standard choice yielded good results: 4 populations with 20 individuals each were used, each with different mutation and selection strategies. All individuals ever tested were recorded. This way, not only one optimal parameter set was computed, but also the suitable parameter range was visible after 50 generations. The only drawback of evolutionary algorithms is that they take a lot of computing time for evaluating lots of individuals.

The most important step in an optimization is the proper choice of the target function.

C. Transfer functions of the resonant system observers

The resonant system observers combine the position sensor signal and current reference information to an output signal. With PI controllers, only the $\hat{\Omega}_1$ information is used. The observers' transfer functions have a common form

$$\hat{\Omega}_1 = \frac{N_1(s) k_T i_q^* + N_2(s)\theta}{D(s)} \tag{C.1}$$

In the following, the three polynomials N_1, N_2, and D are given for the resonant system observers used in section 3.4. The implication of the proportional part of the speed controller K_R is neglected as well as the fact that the observers are had to be transformed to discrete-time domain.

The parameter functions of the 2-inertia speed-only observer are:

$$\begin{gathered} N_1(s) = \\ \left(c_{01}\left(J_0 + J_1\right) - J_1 K_2 - J_0\left(K_2 + K_4\right)\right) s \\ + J_0\left(J_0 + J_1\right) K_1 s^2 \\ + J_0\left(J_0 + J_1\right) s^3 \end{gathered} \tag{C.2}$$

$$\begin{gathered} N_2(s) = \\ -\left(c_{01}\left(J_0 + J_1\right) K_4\right) s \\ + c_{01}\left(J_0 + J_1\right)\left(J_0 K_1 + J_1 K_3\right) s^2 \\ + J_0\left(J_0 K_2 + J_1\left(K_2 - K_4\right)\right) s^3 \\ + J_0 J_1\left(J_0 + J_1\right) K_3 s^4 \end{gathered} \tag{C.3}$$

$$\begin{gathered} D(s) = \\ -\left(c_{01}\left(J_0 + J_1\right) K_4\right) \\ + c_{01}\left(J_0 + J_1\right)\left(J_0 K_1 + J_1 K_3\right) s \\ + \left(c_{01}(J_0 + J_1)^2 - J_1\left(J_1 K_2 + J_0\left(K_2 + K_4\right)\right)\right) s^2 \\ + J_0 J_1\left(J_0 + J_1\right) K_1 s^3 \\ + J_0 J_1\left(J_0 + J_1\right) s^4 \end{gathered} \tag{C.4}$$

The parameter functions of the 3-inertia speed-only observer are:

$$\begin{gathered}
N_1(s) = \\
c_{12}\left(c_{01}\left(J_0+J_1+J_2\right)-\left(J_1+J_2\right)K_2-J_0\left(K_2+K_6\right)\right)s \\
+c_{12}J_0\left(J_0+J_1+J_2\right)K_1s^2 \\
+\left(c_{12}J_0\left(J_0+J_1+J_2\right)+J_2\left(-c_{01}\left(J_0+J_1+J_2\right)+\left(J_1+J_2\right)K_2+J_0\left(K_2+K_6\right)\right)\right)s^3 \\
-J_0J_2\left(J_0+J_1+J_2\right)K_1s^4 \\
-J_0J_2\left(J_0+J_1+J_2\right)s^5
\end{gathered} \tag{C.5}$$

$$\begin{gathered}
N_2(s) = \\
-c_{01}c_{12}\left(J_0+J_1-J_2\right)K_6s \\
+c_{01}c_{12}\left(J_0+J_1+J_2\right)\left(J_0K_1+J_1K_3-J_2K_5\right)s^2 \\
+\left(c_{12}J_0\left(J_0K_2+J_1\left(K_2-K_6\right)+J_2\left(K_2+K_6\right)\right)+c_{01}J_2\left(J_2K_4+J_0\left(K_4+K_6\right)+J_1\left(K_4+K_6\right)\right)\right)s^3 \\
-\left(J_0+J_1+J_2\right)\left(c_{01}J_2\left(J_0K_1+J_1K_3\right)+c_{12}J_0\left(-J_1K_3+J_2K_5\right)\right)s^4 \\
+J_0J_2\left(J_0\left(-K_2+K_4\right)+J_2\left(-K_2+K_4\right)+J_1\left(-K_2+K_4+K_6\right)\right)s^5 \\
-J_0J_1J_2\left(J_0+J_1+J_2\right)K_3s^6
\end{gathered} \tag{C.6}$$

$$\begin{gathered}
D(s) = \\
-\left(c_{01}c_{12}\left(J_0+J_1-J_2\right)K_6\right) \\
+c_{01}c_{12}\left(J_0+J_1+J_2\right)\left(J_0K_1+J_1K_3-J_2K_5\right)s \\
+\left(-c_{12}\left(J_1-J_2\right)\left(\left(J_1+J_2\right)K_2+J_0\left(K_2+K_6\right)\right)\right. \\
\left.+c_{01}\left(c_{12}\left({J_0}^2+2J_0J_1+{J_1}^2-{J_2}^2\right)+J_2\left(J_2K_4+J_0\left(K_4+K_6\right)+J_1\left(K_4+K_6\right)\right)\right)\right)s^2 \\
-\left(J_0+J_1+J_2\right)\left(c_{12}J_0\left(-J_1+J_2\right)K_1+c_{01}J_2\left(J_0K_1+J_1K_3\right)\right)s^3 \\
+\left(c_{12}J_0\left(J_1-J_2\right)\left(J_0+J_1+J_2\right)+J_2\left(-\left(c_{01}\left(J_0+J_1\right)\left(J_0+J_1+J_2\right)\right)\right.\right. \\
\left.\left.+J_1\left(\left(J_1+J_2\right)K_2+J_0\left(K_2+K_6\right)\right)\right)\right)s^4 \\
-J_0J_1J_2\left(J_0+J_1+J_2\right)K_1s^5 \\
-J_0J_1J_2\left(J_0+J_1+J_2\right)s^6
\end{gathered} \tag{C.7}$$

The parameter functions of the 2-inertia complete observer are:

$$\begin{gathered}
N_1(s) = \\
-\left(J_1\left(K_2-c_{01}K_5\right)+J_0\left(K_2+K_4-c_{01}K_5\right)\right)s \\
+\left(J_0+J_1\right)\left(c_{01}+J_0K_1\right)s^2 \\
+J_0\left(J_0+J_1\right)K_5s^3 \\
+J_0\left(J_0+J_1\right)s^4
\end{gathered} \tag{C.8}$$

$$
\begin{gathered}
N_2(s) = \\
-c_{01}\left(J_0+J_1\right)K_4 s \\
+c_{01}\left(J_0+J_1\right)\left(J_0K_1+J_1K_3\right)s^2 \\
+J_0\left(J_0K_2+J_1\left(K_2-K_4\right)\right)s^3 \\
+J_0J_1\left(J_0+J_1\right)K_3 s^4
\end{gathered} \tag{C.9}
$$

$$
\begin{gathered}
D(s) = \\
-\left(J_0+J_1\right)\left(c_{01}K_4+c_{12}\left(K_2-c_{01}K_5\right)\right) \\
+\left(c_{12}J_0J_1\left(K_1-K_3\right)+c_{01}\left(J_0+J_1\right)\left(c_{12}+J_0K_1+J_1K_3\right)\right)s \\
+\left({c_{01}J_0}^2K_5+{J_1}^2\left(-K_2+c_{01}K_5\right)-J_0J_1\left(K_2+K_4-\left(2c_{01}+c_{12}\right)K_5\right)\right)s^2 \\
+\left(c_{01}(J_0+J_1)^2+J_0J_1\left(c_{12}+\left(J_0+J_1\right)K_1\right)\right)s^3 \\
+J_0J_1\left(J_0+J_1\right)K_5 s^4 \\
+J_0J_1\left(J_0+J_1\right)s^5
\end{gathered} \tag{C.10}
$$

The parameter functions of the 3-inertia complete observer are:

$$
\begin{gathered}
N_1(s) = \\
-c_{12}\left(\left(J_1+J_2\right)\left(K_2-c_{01}K_7\right)+J_0\left(K_2+K_6-c_{01}K_7\right)\right)s \\
+c_{12}\left(J_0+J_1+J_2\right)\left(c_{01}+J_0K_1\right)s^2 \\
+\left({c_{12}J_0}^2K_7+J_2\left(J_1+J_2\right)\left(K_2-c_{01}K_7\right)+J_0\left(c_{12}J_1K_7+J_2\left(K_2+K_6-c_{01}K_7+c_{12}K_7\right)\right)\right)s^3 \\
-\left(J_0+J_1+J_2\right)\left(-c_{12}J_0+J_2\left(c_{01}+J_0K_1\right)\right)s^4 \\
-J_0J_2\left(J_0+J_1+J_2\right)K_7 s^5 \\
-J_0J_2\left(J_0+J_1+J_2\right)s^6
\end{gathered} \tag{C.11}
$$

$$
\begin{gathered}
N_2(s) = \\
-c_{01}c_{12}\left(J_0+J_1-J_2\right)K_6 s \\
+c_{01}c_{12}\left(J_0+J_1+J_2\right)\left(J_0K_1+J_1K_3-J_2K_5\right)s^2 \\
+\left(c_{12}J_0\left(J_0K_2+J_1\left(K_2-K_6\right)+J_2\left(K_2+K_6\right)\right)+c_{01}J_2\left(J_2K_4+J_0\left(K_4+K_6\right)+J_1\left(K_4+K_6\right)\right)\right)s^3 \\
-\left(\left(J_0+J_1+J_2\right)\left(c_{01}J_2\left(J_0K_1+J_1K_3\right)+c_{12}J_0\left(-J_1K_3+J_2K_5\right)\right)\right)s^4 \\
+J_0J_2\left(J_0\left(-K_2+K_4\right)+J_2\left(-K_2+K_4\right)+J_1\left(-K_2+K_4+K_6\right)\right)s^5 \\
-\left(J_0J_1J_2\left(J_0+J_1+J_2\right)K_3\right)s^6
\end{gathered} \tag{C.12}
$$

$$
\begin{gathered}
D(s) = \\
-\left(c_{01}c_{12}\left(J_0+J_1-J_2\right)K_6\right) \\
+c_{01}c_{12}\left(J_0+J_1+J_2\right)\left(J_0K_1+J_1K_3-J_2K_5\right)s \\
+\left(c_{01}J_2\left(J_2K_4+J_0\left(K_4+K_6\right)+J_1\left(K_4+K_6\right)\right)\right. \\
\left.+c_{12}\left(c_{01}{J_0}^2K_7-\left({J_1}^2-{J_2}^2\right)\left(K_2-c_{01}K_7\right)+J_0\left(J_2\left(K_2+K_6\right)-J_1\left(K_2+K_6-2c_{01}K_7\right)\right)\right)\right)s^2 \\
+\left(\left(J_0+J_1+J_2\right)\left(c_{12}J_0\left(J_1-J_2\right)K_1+c_{01}\left(c_{12}\left(J_0+J_1-J_2\right)-J_2\left(J_0K_1+J_1K_3\right)\right)\right)\right)s^3 \\
+\left({J_0}^2\left(c_{12}\left(J_1-J_2\right)-c_{01}J_2\right)K_7+J_1J_2\left(J_1+J_2\right)\left(K_2-c_{01}K_7\right)\right. \\
\left.+J_0\left(c_{12}{J_1}^2K_7-\left(c_{01}+c_{12}\right){J_2}^2K_7+J_1J_2\left(K_2+K_6-2c_{01}K_7\right)\right)\right)s^4 \\
-\left(\left(J_0+J_1+J_2\right)\left(c_{12}J_0\left(-J_1+J_2\right)+J_2\left(c_{01}\left(J_0+J_1\right)+J_0J_1K_1\right)\right)\right)s^5 \\
-J_0J_1J_2\left(J_0+J_1+J_2\right)K_7s^6 \\
-J_0J_1J_2\left(J_0+J_1+J_2\right)s^7
\end{gathered}
\tag{C.13}
$$

The parameter functions of the 2-inertia reduced observer are:

$$
\begin{gathered}
N_1(s) = \\
\left(c_{01}\left(J_0+J_1\right)-J_1K_2-J_0\left(K_2+K_4\right)\right)s \\
+J_0\left(J_0+J_1\right)K_1s^2 \\
+J_0\left(J_0+J_1\right)s^3
\end{gathered}
\tag{C.14}
$$

$$
\begin{gathered}
N_2(s) = \\
-c_{01}\left(J_0+J_1\right)K_4s \\
+c_{01}\left(J_0+J_1\right)\left(J_0K_1+J_1K_3\right)s^2 \\
+J_0\left(J_0K_2+J_1\left(K_2-K_4\right)\right)s^3 \\
+J_0J_1\left(J_0+J_1\right)K_3s^4
\end{gathered}
\tag{C.15}
$$

$$
\begin{gathered}
D(s) = \\
-\left(J_0+J_1\right)\left(c_{12}K_2+c_{01}\left(-c_{12}+K_4\right)\right) \\
+\left(c_{12}J_0J_1\left(K_1-K_3\right)+c_{01}\left(J_0+J_1\right)\left(J_0K_1+J_1K_3\right)\right)s \\
+\left(c_{01}{(J_0+J_1)}^2-J_1\left(-\left(c_{12}J_0\right)+J_1K_2+J_0\left(K_2+K_4\right)\right)\right)s^2 \\
+J_0J_1\left(J_0+J_1\right)K_1s^3 \\
+J_0J_1\left(J_0+J_1\right)s^4
\end{gathered}
\tag{C.16}
$$

The parameter functions of the 3-inertia reduced observer are:

$$
\begin{gathered}
N_1(s) = \\
c_{12}\left(c_{01}\left(J_0+J_1+J_2\right)-\left(J_1+J_2\right)K_2-J_0\left(K_2+K_6\right)\right)s \\
+c_{12}J_0\left(J_0+J_1+J_2\right)K_1s^2 \\
+\left(c_{12}J_0\left(J_0+J_1+J_2\right)+J_2\left(-\left(c_{01}\left(J_0+J_1+J_2\right)\right)+\left(J_1+J_2\right)K_2+J_0\left(K_2+K_6\right)\right)\right)s^3 \\
-\left(J_0J_2\left(J_0+J_1+J_2\right)K_1\right)s^4 \\
-\left(J_0J_2\left(J_0+J_1+J_2\right)\right)s^5
\end{gathered}
\tag{C.17}
$$

$$
\begin{gathered}
N_2(s) = \\
c_{01}c_{12}\left(J_0+J_1-J_2\right)K_6s \\
-\left(c_{01}c_{12}\left(J_0+J_1+J_2\right)\left(J_0K_1+J_1K_3-J_2K_5\right)\right)s^2 \\
+\left(-\left(c_{12}J_0\left(J_0K_2+J_1\left(K_2-K_6\right)+J_2\left(K_2+K_6\right)\right)\right)-c_{01}J_2\left(J_2K_4+J_0\left(K_4+K_6\right)+J_1\left(K_4+K_6\right)\right)\right)s^3 \\
+\left(\left(J_0+J_1+J_2\right)\left(c_{01}J_2\left(J_0K_1+J_1K_3\right)+c_{12}J_0\left(-\left(J_1K_3\right)+J_2K_5\right)\right)\right)s^4 \\
+J_0J_2\left(J_0\left(K_2-K_4\right)+J_2\left(K_2-K_4\right)+J_1\left(K_2-K_4-K_6\right)\right)s^5 \\
+J_0J_1J_2\left(J_0+J_1+J_2\right)K_3s^6
\end{gathered}
\tag{C.18}
$$

$$
\begin{gathered}
D(s) = \\
-c_{01}c_{12}\left(J_0+J_1-J_2\right)K_6 \\
+c_{01}c_{12}\left(J_0+J_1+J_2\right)\left(J_0K_1+J_1K_3-J_2K_5\right)s \\
+\left(-\left(c_{12}\left(J_1-J_2\right)\left(\left(J_1+J_2\right)K_2+J_0\left(K_2+K_6\right)\right)\right)\right. \\
\left.+c_{01}\left(c_{12}\left({J_0}^2+2J_0J_1+{J_1}^2-{J_2}^2\right)+J_2\left(J_2K_4+J_0\left(K_4+K_6\right)+J_1\left(K_4+K_6\right)\right)\right)\right)s^2 \\
-\left(\left(J_0+J_1+J_2\right)\left(c_{12}J_0\left(-J_1+J_2\right)K_1+c_{01}J_2\left(J_0K_1+J_1K_3\right)\right)\right)s^3 \\
+\left(c_{12}J_0\left(J_1-J_2\right)\left(J_0+J_1+J_2\right)-J_2c_{01}\left(J_0+J_1\right)\left(J_0+J_1+J_2\right.\right. \\
\left.\left.+J_1\left(\left(J_1+J_2\right)K_2+J_0\left(K_2+K_6\right)\right)\right)\right)s^4 \\
-J_0J_1J_2\left(J_0+J_1+J_2\right)K_1s^5 \\
-J_0J_1J_2\left(J_0+J_1+J_2\right)s^6
\end{gathered}
\tag{C.19}
$$

List of Figures

List of Tables

Bibliography

Publications of the project

[1] A. Bähr, P. Mutschler: *Comparison of Speed Acquisition Methods based on Sinusoidal Encoder Signals*, Optim 2002 conference, Brasov, Romania; conference record pp. 497-502; http://www.srt.tu-darmstadt.de

[2] A. Bähr, P. Mutschler: *Systematic Error Correction Methods for Sinusoidal Encoders and their Application in Servo Control*, EPE 2003 conference, Toulouse, France

[3] A. Bähr, P. Mutschler: *Theoretical and experimental investigation of a servo control's dynamic stiffness and smoothness*, Optim 2004 conference, Brasov, Romania; in press; http://www.srt.tu-darmstadt.de

[4] J. Maletschek, *Prädizierende Filter für die Antriebsregelung*; Diplomarbeit, TU Darmstadt 2004

[5] O. Fargouche, *Auslegung und Simulation von Antriebsregelkreisen mit Beschleunigungsaufschaltung und -regelung*; Diplomarbeit, TU Darmstadt 2002

[6] H. Nouira, *Auslegung von Beobachter und Zustandsregelung zur Bedämpfung eines Mehrmassenschwingers*; Diplomarbeit, TU Darmstadt 2001

Introduction and textbooks

[7] O. Foellinger, *Regelungstechnik*; 8. Auflage, Huethig Buchverlag 1994

[8] J. Adamy, *Regelungstechnik II*; lecture script, TU Darmstadt 1999

[9] D. Schröder, *Elektrische Antriebe Bd. 2: Regelung von Antrieben*; Springer-Verlag 1995

[10] U. Tietze, Ch. Schenk, *Halbleiter-Schaltungstechnik*; Springer-Verlag 1993

[11] A.V. Oppenheim, R. W. Schäfer: *Zeitdiskrete Signalverarbeitung*, R. Oldenbourg Verlag Muenchen Wien 1999

[12] T. Soederstroem, P. Stoica: *System Identification*, Prentice Hall 1989

[13] W. Beitz, K.-H. Küttner, *Dubbel- Taschenbuch für den Maschinenbau*; 17. Auflage, Springer-Verlag 1990

[14] K. Aström, Bjorn Wittenmark, *Computer-controlled Systems - Theory and Design*; Prentice Hall 1990

[15] J. Ackermann, *Abtastregelung*, Springer-Verlag 1988

[16] Mathworks Inc., *MATLAB support site*, http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml

[17] Wolfram Research, *Mathematica home page*, http://www.wolfram.com/products/mathematica

[18] H. Pohlheim: *Evolutionäre Algorithmen*, Springer Verlag 2000; http://www.geatbx.com

[19] M. Kazmierkowsky, A. Dzieniakowsky, W. Sulkowski: *Novel Space Vector Based Current Controllers for PWM-Inverters*; IEEE Transactions on Power Electronics Vol. 6 pp. 158-165, January 1991

[20] M. Weck, P. Krueger, Ch. Brecher, F. Remy: *Statische und dynamische Steifigkeit von linearen Direktantrieben*; Antriebstechnik Nr. 12, Jg. 36 (Dec. 1997), pp. 57-63

[21] On-Time Informatik GmbH, *RT-Target website*; http://www.on-time.com/

Filters

[22] S. Vukosavic, M Stojic: *Suppression of Torsional Oscillations in a High-Performance Speed Servo Drive*; IEEE Transactions on Industrial Electronics, Vol.45 No.1 pp. 108-117, February 1998

[23] A. Gees: *Accelerometer-enhanced Speed Estimation for Linear-Drive Machine Tool Axes*; Diss. Ecole Polytechnique Federale de Lausanne 1996

[24] U. Brahms: *Regelung von Lineardirektantrieben für Werkzeugmaschinen*; Diss. Universität Hannover 1998

[25] S. Brueckl: *Regelung von Synchron-Linearmotoren für hochgenaue Vorschubantriebe bei Werkzeugmaschinen*; Diss. TU Muenchen 2000, VDI Fortschritt-Berichte Reihe 8, Nr. 831

[26] P. Heinonen, Yrjö Neuvo, *FIR-Median Hybrid Filters with Predictive FIR Substructures*; IEEE Transactions on Acoustics, Speed, and Signal Processing, vol. 36, no. 6, June 1988

[27] S. Väliviita, S. Ovaska, *Polynomial Predictive Filtering in Control Instrumentation: A Review*; IEEE Transactions on Industrial Electronics, vol. 46 no.5, oct. 1999

[28] S. Ovaska, O. Vainio, T. Laakso, *Design of Predictive IIR Filters via Feedback Extension of FIR Forward Predictors*; IEEE Transctions on Instrumentation and Measurement, vol. 46, issue 5, oct. 1997

[29] S. Väliviita, S. Ovaska, *Delayless recursive differentiator with efficient noise attenuation for control instrumentation*; Elsevier Signal Processing 69 (1998), pp. 267-280

[30] S. Ovaska, *FIR Prediction Using Newton's Backward Interpolation Algorithm with Smoothed Successive Differences*; IEEE Transactions on Instrumentation and Measurement, vol. 40, no. 5, oct. 1991

[31] S. Ovaska, O. Vainio, *Recursive Linear Smoothed Newton Predictors for Polynomial Extrapolation*; IEEE Transactions on Instrumentation and Measurement, vol. 41, no. 4, aug. 1992

[32] O. Vainio, S. Ovaska, *Tachometer Signal Smoothing with Analog Discrete-Time Polynomial Estimators*; IEEE Transactions on Industrial Electronics vol. 41 no. 2; April 1994

Observers and resonant systems

[33] Y. Hori, H. Iseki, K. Sugiura: *Basic Consideration of Vibration Suppression and Disturbance Rejection Control of Multi-Inertia System using SFLAC (State Feedback and Load Acceleration Control)*; IEEE Transactions on Industry Applications, Vol.30 No.4 pp. 889-896 , July/ August 1994

[34] G. Ferretti, G. Magnani, P. Rocco: *LQG Control of Elastic Servomechanisms Based on Motor Position Measurements*; IEEE 5th International Workshop on Advanced Motion Control in Coimbra 1998, pp. 617-622

[35] M. Goslar: *Ein Beitrag zur anwendungsorientierten Zustandsregelung elektrischer Hochleistungsantriebe*; Diss. TU Clausthal 1998

[36] J. Fassnacht, P. Mutschler: *Benefits and Limits of Using an Acceleration Sensor in Actively Damping High Frequent Mechanical Oscillations*; 36th IEEE-IAS Annual Meeting in Chicago 2001, Conference Record pp. 2337-2344

[37] J. Fassnacht, *Schwingungsbedämpfung in Servosystemen mit der direkten Drehmomentmittelwertregelung*, PhD thesis 2002, available online at http://elib.tu-darmstadt.de/diss/000273/

[38] K. Suguira, Y. Hori: *Vibration Suppression in 2- and 3-Mass System Based on the Feedback of Imperfect Derivative of the Estimated Torsional Torque*; IEEE Transactions on Industrial Electronics Vol. 43 No. 1 pp. 56-64, February 1996

[39] G. Ellis, R. D. Lorenz: *Resonant Load Control Methods for Industrial Servo Drives*, IEEE Industry Applications Conference 2000, Conference Record vol. 3 pp. 1438-1445

[40] S. Yang, S. Ke: *Performance Evaluation of a Velocity Observer for Accurate Velocity Estimation of Servo Motor Drives*, IEEE Transactions on Industry Applications, vol. 36 no. 1 pp. 98-104, January/ February 2000

[41] I. Mueller, P. Mutschler, *Two Reliable Methods for Estimating the Mechanical Parameters of a Rotating Three-Inertia System*; EPE-PEMC 2002 Conference Dubrovnik & Cavtat

[42] D. Alders, R. Kennel, J. O. Krah, J. Quan, *BiQuad-Filter for Active Damping of Low Frequency Oscillations in Two-Inertia Drive Systems*; proceedings of IEMDC 2003, Madison, Wisconsin, pp. 1836-1831

Encoder signals and error correction

[43] B. Hoescheler: *Erhoehung der Genauigkeit bei Wegmeßsystemen durch selbstlernende Kompensation systematischer Fehler*, conference record SPS/IPC/Drives 1999, pp. 617-626

[44] B. Hoescheler: *Entwicklung eines Pruefstands zur Echtzeitsimulation mechanischer Systeme unter Verwendung einer hochaufloesenden Lageerfassung*, PHD thesis, University of Erlangen-Nuernberg 2001

[45] A. Bünte, S. Beineke: *Suppression of Systematic Resolver and Encoder Errors*, conference record PCIM 2001, pp. 305-310

[46] A. Probst: *Untersuchungen zur Verbesserung des Lastverhaltens elektrischer Antriebe am Beispiel einer mikrorechnergeregelten Asynchronmaschine*, PHD thesis, University of Technology Darmstadt 1994

[47] S. Venema, *A Kalman Filter Calibration Method For Analog Quadrature Position Encoders*, Master Thesis, University of Washington 1994; http://brl.ee.washington.edu/publications/Th010.pdf

[48] R. Kirchberger, B: Hiller: *Oversamplingverfahren zur Verbesserung der Erfassung von Lage und Drehzahl an elektrischen Antrieben mit inkrementellen Gebersystemen*; SPS/IPC/Drives 1999, Nuernberg

[49] H. Lim, J. Seo, C. Choi, *Torsional displacement compensation in position control for machining centers*; preprint available at http://csl.snu.ac.kr/publication/paper/hyuk_lpc.pdf

Using acceleration information

[50] J. Fassnacht, P. Mutschler, *An observer to improve the speed signal using a Ferraris acceleration sensor*; paper no. 195, Proceedings of EPE 2001, Graz, Austria

[51] U. Weinmann: *Regelung von Direktantrieben mit Ferraris Sensoren - Praxisbericht*; SPS/IPC/Drives 2002, Conference record pp. 751-756

[52] G. Stoeppler, S. Douglas: *Alternative control structure to determine the velocity signal by employing an acceleration sensor to control a servo motor*; Proceedings of PCIM 2002, Nuernberg

[53] P. Schmidt, R. Lorenz, *Design Principles of Acceleration Feedback to Improve Performance of dc Drives*; IEEE transactions on Industry applications von. 28 no. 3, May/June 1992

[54] I. Godler, K. Hashimoto, T. Ninomiya, *Stability, gain and sampling time of discrete-time acceleration control loop*; Proceedings of IEEE International Symposium on Industrial Electronics 2001, Pusan, Korea

[55] Y. Hori, *Disturbance suppression on an acceleration control type servo system*; Conference record PESC 1988, pp. 222-229

Computational effort

[56] Texas Instruments, *TMS320F/C24x DSP Controllers Reference Guide*, literature number SPRU160C, 1999; http://dspvillage.ti.com

[57] Texas Instruments, *Fixed-Point Math Library for C24x*, literature number SPRC068; http://dspvillage.ti.com

[58] Texas Instruments, *TMS320F/C240 DSP Controllers Reference Guide, Peripheral Library and Specific Devices*, literature number SPRU161C, 1999; http://dspvillage.ti.com

[59] Ingo Mueller, *Field oriented Control*; practical training for students, TU Darmstadt 2004

[60] Jan Weigel, *AC Drive with Voltage Source Inverter*; practical training for students, TU Darmstadt 2004

[61] Analog Devices, *ADSP-21160 SHARC DSP Instruction Set Reference*; 1999

[62] Analog Devices, *ADSP-21xxx Applications Handbook File Area: Code Examples*; http://www.analog.com/processors/processors/sharc/technicalLibrary/codeExamples/applicationsHandbook.html

Curriculum Vitae

Alexander Bähr
born in Karlsruhe, Germany, 28.08.1975
married

since 2004	working as control software engineer at Lust Antriebstechnik, Lahnau, Germany
2000 - 2004	PhD project at the Institute of Power Electronics and Control of Drives, Darmstadt University of Technology
1995 - 2000	Study of Electrical Engineering at Darmstadt University of Technology
1994 - 1995	Civil Service at old people's home Augustinum, Bad Soden
1987 - 1994	Secondary school at Helene-Lange-Gymnasium and Friedrich-Dessauer-Gymnasium, Frankfurt-Höchst