Control of Flexible Servo Systems

Wallenborg, Anders

1987

Document Version:
Publisher's PDF, also known as Version of record

Control of Flexible Servo Systems

Anders Wallenborg

Department of Automatic Control
Lund Institute of Technology
April 1987

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Licentiate Thesis

Date of issue
April 1987

Document Number
CODEN: LUTFD2/(TFRT-3188)/1-104/(1987)

Author(s)
Anders Wallenborg

Supervisor
Karl Johan Åström

Sponsoring organisation

Title and subtitle
Control of Flexible Servo Systems.

Abstract

Analysis and design of high performance mechanical servo systems require that flexible modes and nonlinearities are considered. This thesis treats the design of a speed control system for a flexible servo where the nonlinear friction cannot be neglected. It is shown that a control law based on linear state feedback and an observer can be used to provide active damping of the resonant modes.

With a colocated speed sensor, i.e. the measured signal is the drive motor speed, the achievable bandwidth is limited by the appearance of limit cycle oscillations at zero speed. The limit cycles are caused by the Coulomb friction in the drive motor, and their appearance is related to the stability of the regulator. It is shown that an unstable linear regulator is a sufficient condition for limit cycle oscillations in this type of system. This, in combination with the inherent windup problem, leads to the conclusion that unstable regulators should be avoided if possible in flexible servo systems.

The stability of the regulator depends mainly on a dimensionless quantity, which includes the load inertia, the spring constant of the elastic modes and the closed loop natural frequency. Analytic expressions have been derived from which approximate values of the stability limit can be calculated with good accuracy. The limit cycles can be avoided by using a non-colocated speed sensor, i.e. by measuring the inertial load speed instead of the drive motor speed.

Another approach is to "linearize" the system by introducing a nonlinear friction compensation scheme in the control law. This works well in simulations but not so well in practice. Simple models have been used to analyse and predict the behavior of the flexible servo system. They give good insight into qualitative properties and also, in most cases, quite accurate quantitative results.

All results have been verified experimentally on a simple prototype flexible servo system with an analog controller.

Key words
Control design, servo systems, resonant modes, acgive damping, nonlinear systems, describing function, limit cycles, robotics.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
104

Recipient's notes

Security classification

PREFACE

This work has been carried out at the Department of Automatic Control in Lund during a period of two years. It started off as a small feasibility study of the design of high performance control systems for robot servos, using modern control theory and taking the flexible modes of the system into account. During this early phase of the project, I discovered that in some cases the simulations showed a limit cycle oscillation in the closed loop system. This limit cycle phenomenon was interesting and motivated further research. It was therefore decided to build a prototype flexible servo in the lab at the department. Hopefully this would enable us to verify the existence of these limit cycles in "real life". Imagine my joy when my first prototype analog controller was connected to the system, and the thing started oscillating! From that point the work continued with theoretical analyses, now and then interrupted by new experiments.

There are many aspects of control system design that require practical experience. This is something that is difficult to learn from books. Therefore it has been both instructive and rewarding to carry out a complete control design project, including modelling and parameter estimation, analysis, control design, simulations, and finally the implementation of the control law. This thesis is an attempt to summarize both the theoretical results and my practical experiences from the project.

Although this has been mainly a one man project, many of my colleagues have helped me in different ways. I would therefore like to take this opportunity to acknowledge their contributions.
First of all I would like to thank my supervisor, professor Karl Johan Åström. He originally proposed the study of flexible servo systems, and he has contributed with much valuable advice and many useful ideas. Per Hagander, Lars Nielsen, Sven Erik Mattsson and Bernt Nilsson have read different versions of the manuscript and I appreciate their valuable criticism. In particular I am grateful to Per Hagander, with whom I have spent many hours in fruitful and enlightening discussions. Sven Erik Mattsson also demonstrated some of the hazards with unstable controllers.
The excellent software facilities at the department have been a great help. The manuscript has been prepared with the text processing program 'Report', written by Leif Andersson. I would like to thank Michael Lundh, who wrote the original version of the MACSYMA program 'Ackermann'. Mats Lilja has generously shared his vast experience in the use of Ctrl-C with me. He also gave me the idea to the proof of Lemma 5.1. Many thanks are due to Rolf Braun for helping me with a number of practical problems in the experimental work.

Finally, I would like to thank my wife Kerstin for her understanding, support and patience.

Lund, April 1987

Anders Wallenborg

CONTENTS

1. INTRODUCTION

Motivation

Motion control problems have received increased attention during the last few years. One reason for this is the breakthrough of robots and other mechanical manipulators in industrial automation. Robots are now being used in a wide variety of applications, and the ever-increasing demand for improved performance requires more and more sophisticated control strategies.

Improved performance normally implies both faster and more accurate motion. With the traditional industrial robot construction, high speed and acceleration require quite large and expensive servo motors. Therefore, the current trend in robotics is towards smaller and lighter mechanical systems, whereby improved speed and acceleration can be achieved with smaller and less expensive actuators. These lightweight constructions have one major drawback, however. The robot can no longer be considered as a rigid body, and mechanical resonance phenomena appear. Accurate and fast robot motion thus requires that the flexibles modes of the system are taken into account in the control design. Improved performance also often requires that the presence of nonlinearities, such as static friction, must be considered. Hence a linear model and design may be inadequate, and we are faced with a much more complicated control problem.

There are a number of other areas where motion control problems of this type also appear, for example in the design of large flexible space structures. The need to consider flexible modes is in fact a control problem, which concerns the design of any type of high performance mechanical servo system.

Background

Several authors have studied the control of flexible servo systems (based on linear models and control algorithms) and nonlinear friction compensation in rigid body servo systems. See for example [1], [4] and [14]. I believe, however, that the combined effects of flexible modes and nonlinear friction in the system have not yet been fully investigated.

This work started as a feasibility study of the control of flexible servo systems using modern control theory, with robotic applications in mind. The intended approach was to use linear state feedback control to provide active damping of the resonant modes, and if necessary some sort of friction compensation scheme to eliminate the nonlinear friction. Simulation experiments soon indicated that nonlinear friction in combination with a high bandwidth linear control design could give rise to limit cycle oscillations. The oscillations occur at zero speed, and can not be explained as a slip-stick phenomenon. I believe that these oscillations have not been observed and analysed before.

A prototype flexible servo system was built to verify the existence of such oscillations. Having established that the problem existed in "real life", the project focused on exploring the mechanism behind these oscillations and how they could be avoided.

Thesis outline

This thesis treats the problem of controlling the speed loop of a flexible servo system. The thesis is organized as follows. A prototype flexible servo system is presented in Chapter 2. Chapter 3 describes the mathematical model of the system. A linear regulator design which provides active damping of the resonant modes is presented in Chapter 4. This chapter also contains some simulations with the linear regulator and a nonlinear servo model including static and Coulomb friction. The simulations demonstrate the existence of limit cycle oscillations if the closed loop bandwidth is chosen too high. Some theoretical results obtained with the describing function method and a simplified nonlinear friction model are given in Chapter 5. The existence of limit cycles, as well as their amplitude and frequency, is predicted for the nominal design case. It is then proved that an unstable regulator is a sufficient criterion for the existence of limit cycles. In Chapter 6 we explore the bandwidth constraints imposed on the linear control design by the appearance of limit cycles. It is shown that these constraints can be relaxed by choosing a different sensor location. Some simple friction compensation schemes are also investigated. Chapter 7 deals with the implementation of the control algorithms and experimental verification of the simulation results. Finally, the results are summarized in Chapter 8 with conclusions and suggestions for further research.

2. A PROTOTYPE FLEXIBLE SERVO SYSTEM

Experimental arrangement

A laboratory flexible servo system has been built to enable experimental testing and verification of theoretical results and simulations. The laboratory system is a two-inertia system with an electric DC drive motor connected to an inertial load by a weak shaft. A simple bearing arrangement was provided for the inertial load by fitting a second DC motor to the other side of the weak shaft. The physical dimensions of the system were chosen so that the resonance frequency of the oscillatory modes would be reasonably low, while at the same time keeping the system within "desk-top" size. This was accomplished by using a thin wire as the weak shaft. A picture of the prototype system is shown in Figure 2.1 .

In order to keep the instrumentation simple we have studied the speed control problem only. This means that no position transducers were needed. The drive motor of the servo system was fitted with a DC tacho generator to enable speed measurements on the drive motor shaft. By using the second DC motor as a tacho generator it is also possible to measure the speed of the inertial load.

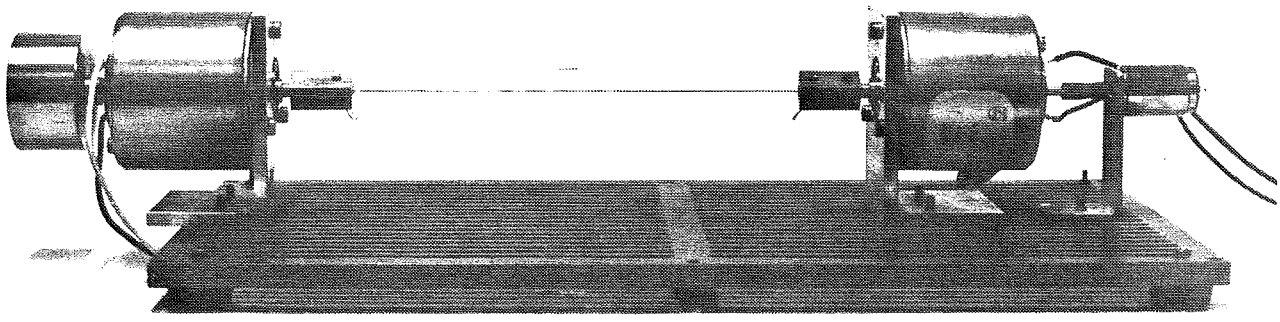


Fig. 2.1 The prototype flexible servo system. The DC drive motor and tacho generator is to the right, and the inertial load is to the left.

Input and output signals

Our main objective is to design a speed control system for the inertial load, while taking the flexible modes of the system into account. The natural choice of measured process output signal would therefore be the inertial load speed. In most servo applications, however, the drive motor speed is the only available signal. The drive motor speed is therefore used as the measured process output in the control algorithm. The inertial load speed signal will be used primarily for diagnostic purposes to check the control system performance.

The drive motor is equipped with an electronic amplifier with current feedback. We will consider the input voltage to the drive amplifier as the input to the flexible servo system.

3. MATHEMATICAL MODEL

3.1 LINEAR MODEL

Let the variables related to the drive motor have index 1 and the variables related to the inertial load have index 2. The flexible servo system is modelled by two shafts with moments of inertia J_1 and J_2, coupled with an elastic spring. The spring constant is denoted k, and the damping coefficient d. The effect of viscous friction in bearings etc. is described by an individual damping on each shaft with the damping coefficients d_1 and d_2. See Figure 3.1 .

The drive motor is a permanent magnet DC motor with ironless rotor, driven by an electronic amplifier with current feedback. The dynamics of the motor and the amplifier are neglected. The motor is thus assumed to give a torque M propotional to the input current I. Because of the current feedback, the torque M is also proportional to the input voltage u of the drive amplifier:

$$M = k_m \cdot I = k_m \cdot k_i \cdot u$$

Torque balance for the two shafts yields

$$\begin{cases} J_1 \cdot \dot{\omega}_1 = k(\theta_2 - \theta_1) - d_1\omega_1 - d(\omega_1 - \omega_2) + k_m k_i \cdot u \\ J_2 \cdot \dot{\omega}_2 = -k(\theta_2 - \theta_1) - d_2\omega_2 + d(\omega_1 - \omega_2) \end{cases}$$

By definition we also have

$$\begin{cases} \dot{\theta}_1 = \omega_1 \\ \dot{\theta}_2 = \omega_2 \end{cases}$$

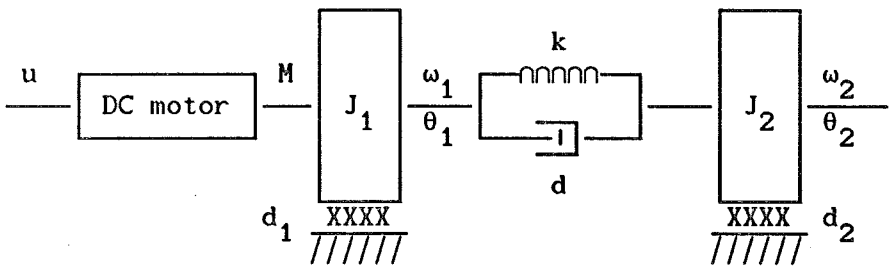


Fig. 3.1 Linear flexible servo model

where

ω_1 = angular velocity of shaft 1

ω_2 = angular velocity of shaft 2

θ_1 = position of shaft 1

θ_2 = position of shaft 2

Introduce the state variables

$$x_1 = \omega_1$$

$$x_2 = \omega_2$$

$$x_3 = \theta_2 - \theta_1$$

and the measured output signal

$$y = k_{\omega 1} \cdot \omega_1$$

where $k_{\omega 1}$ is the drive motor tachometer gain. The input signal, u, is the input voltage to the drive amplifier.

Note that we only need the position difference as the third state since our control objective is speed control of the servo. If position control is required a fourth state must be added, e.g. $x_4 = \theta_1$. This gives the following speed loop state space model of the flexible servo:

$$\dot{x}(t) = \begin{bmatrix} \frac{-(d_1+d)}{J_1} & \frac{d}{J_1} & \frac{k}{J_1} \\ \frac{d}{J_2} & \frac{-(d_2+d)}{J_2} & \frac{-k}{J_2} \\ -1 & 1 & 0 \end{bmatrix} \cdot x(t) + \begin{bmatrix} \frac{k_m \cdot k_i}{J_1} \\ 0 \\ 0 \end{bmatrix} \cdot u(t) \qquad (3.1)$$

$$y(t) = \begin{bmatrix} k_{\omega 1} & 0 & 0 \end{bmatrix} \cdot x(t)$$

An alternative state space model for the case when we measure ω_2 instead of ω_1 is easily formulated by replacing the output definition with

$$y(t) = \begin{bmatrix} 0 & k_{\omega 2} & 0 \end{bmatrix} \cdot x(t)$$

where $k_{\omega 2}$ is the gain of the tachometer on shaft 2.

Transfer functions

The state space model (3.1) has the transfer function

$$G_1(s) = \frac{B_1(s)}{A(s)} = \frac{k_i k_m k_{\omega 1}\left[J_2 s^2+(d_2+d)s+k\right]}{A(s)} \tag{3.2}$$

where

$$A(s) = J_1 J_2 s^3+\left[J_2(d_1+d)+J_1(d_2+d)\right]s^2+$$
$$+\left[k(J_1+J_2)+d_1 d_2+dd_1+dd_2\right]s+(d_1+d_2)k$$

The transfer function from u to $y_2 = k_{\omega 2}\omega_2$ is

$$G_2(s) = \frac{B_2(s)}{A(s)} = \frac{k_i k_m k_{\omega 2}\left[ds+k\right]}{A(s)} \tag{3.3}$$

where A(s) is defined as in (3.2). If the linear damping coefficients d, d_1 and d_2 are small, the servo system has two poorly damped oscillatory poles with a natural frequency approximately given by

$$\omega_o^2 = \frac{k(J_1+J_2)}{J_1 J_2}$$

and one slow pole on the negative real axis approximately given by

$$s = -\frac{d_1+d_2}{J_1+J_2}$$

This pole corresponds to the rigid body dynamics. The steady state gain from u to y_1 is

$$G_1(0) = \frac{k_i k_m k_{\omega 1}}{d_1+d_2}$$

Note that small linear damping coefficients gives a high steady state gain.

Parameter values

The parameters J_1, k_m, k_i, $k_{\omega 1}$ and $k_{\omega 2}$ were given by component data sheets and circuit diagrams. The inertial load, J_2, was calculated from its physical dimensions (with the inertia of motor 2 included).

By performing a simple frequency response experiment it was found that the damping d in the elastic spring could be neglected. The frequency response of the transfer function $G_2(s)$ from the input voltage u to $y_2 = k_{\omega 2}\omega_2$ was recorded in the frequency interval 0.1 - 10 Hz. At higher frequencies the gain of the system was so small that the output signal could not be distinguished from the measurement noise. $G_2(s)$ has a zero at

$$s = -\frac{k}{d}$$

This zero has a significant influence on the frequency response only at frequencies near or above the breakpoint $\omega_b = k/d$. No influence from the zero could be noticed in the measured frequency response. The high frequency asymptote was found to have a slope of -60 dB per decade and a phase shift of -270 degrees. This agrees well with the transfer function (3.3) if d is zero. The damping d must therefore be so small that the breakpoint from the zero is well above the frequency interval where $|G_2(i\omega)|$ is significantly greater than zero, and hence d can be neglected.

When performing the frequency response experiments, it was found that the phase curve was badly distorted by the presence of nonlinear (Coulomb) friction. This depends on the discontinuous nature of the Coulomb friction, which causes problems when the direction of rotation changes. To overcome this problem, a bias was added to the sinusoidal input. The bias amplitude was selected larger than the sinusoidal amplitude, so that zero crossings in the measured output speed were avoided. With a constant direction of rotation, the Coulomb friction can be regarded as a constant torque disturbance and it does not influence the results of the frequency analysis.

The remaining parameters were determined by fixing shaft 1 and 2, respectively, and studying the decay of an initial disturbance in the position of the free shaft. With J_1 and J_2 given, the spring constant k can be calculated from the resonance frequency. Approximate estimates of d_1 and d_2 can be obtained from the decay of the oscillations. These estimates of the viscous damping will be rather crude, however, since the behaviour of the system differs from the linear model due to the presence of static (Coulomb) friction. Improved estimates were obtained by simulating a nonlinear model including static friction and comparing the simulations with experimental data.

The following parameter values were obtained for the laboratory setup:

$k_m = 0.1$ Nm/A
$k_i = 0.25$ A/V
$J_1 = 22 \cdot 10^{-6}$ kgm^2
$J_2 = 150 \cdot 10^{-6}$ kgm^2
$k = 2.4 \cdot 10^{-3}$ Nm/rad
$d = 0.0$ Nm/rad/s
$d_1 = 1 \cdot 10^{-5}$ Nm/rad/s
$d_2 = 1 \cdot 10^{-5}$ Nm/rad/s
$k_{\omega 1} = 0.1$ V/rad/s
$k_{\omega 2} = 0.1$ V/rad/s

With these parameter values we get the following linear state space model:

$$\dot{x} = \begin{bmatrix} -0.45 & 0 & 109 \\ 0 & -0.07 & -16.0 \\ -1 & 1 & 0 \end{bmatrix} \cdot x(t) + \begin{bmatrix} 1136 \\ 0 \\ 0 \end{bmatrix} \cdot u(t) \qquad (3.4)$$

$$y = \begin{bmatrix} 0.1 & 0 & 0 \end{bmatrix} \cdot x(t)$$

The state space model (3.4) has the transfer function

$$G_1(s) = \frac{113.6(s^2+0.07s+16)}{(s+0.1)(s^2+0.4s+125)} \qquad (3.5)$$

A singularity diagram with the corresponding poles and zeros is shown in Figure 3.2. The transfer function from u to the alternative output $y_2 = k_{\omega 2}\omega_2$ is

$$G_2(s) = \frac{1818}{(s+0.1)(s^2+0.4s+125)} \qquad (3.6)$$

Note the different zero structure with two complex zeros in $G_1(s)$ but no zero in $G_2(s)$. This will be important in the control design (see Chapter 4).

Simplified linear model

The A-matrix elements in eqn (3.1) and (3.4) corresponding to the linear damping terms (element [1,1], [1,2], [2,1] and [2,2]) are very small. A simplified state space model can be obtained by setting $d = d_1 = d_2 = 0$. This gives

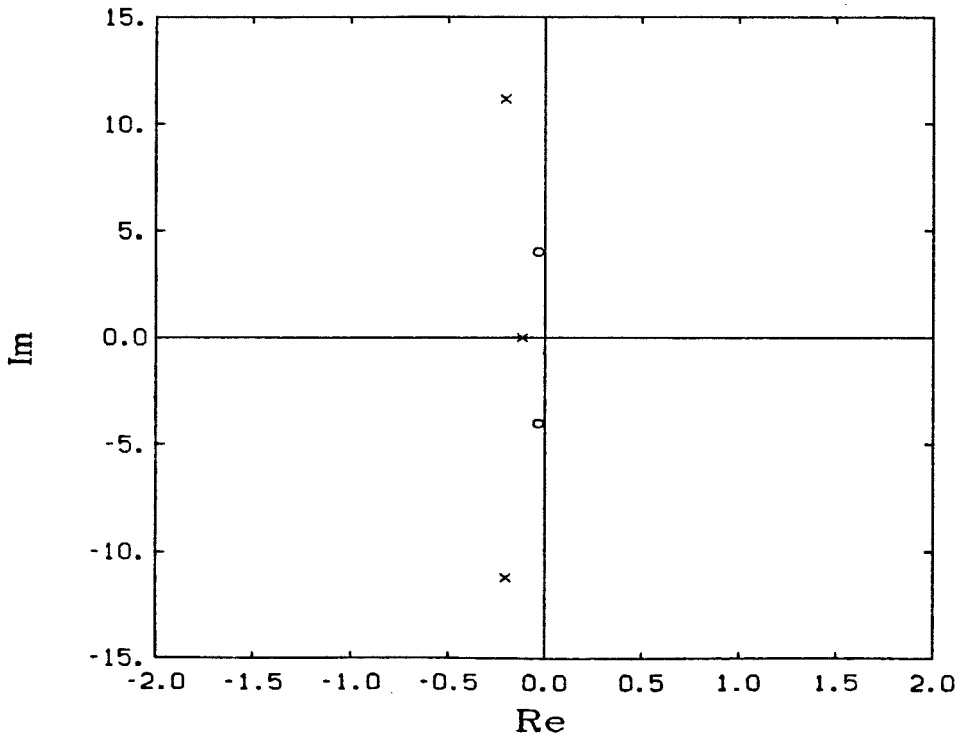


Fig. 3.2 Singularity diagram for the open loop system (x = poles, o = zeros).

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & \frac{k}{J_1} \\ 0 & 0 & \frac{-k}{J_2} \\ -1 & 1 & 0 \end{bmatrix} \cdot x(t) + \begin{bmatrix} \frac{k_m \cdot k_i}{J_1} \\ 0 \\ 0 \end{bmatrix} \cdot u(t) \qquad (3.7)$$

$$y(t) = \begin{bmatrix} k_{\omega 1} & 0 & 0 \end{bmatrix} \cdot x(t)$$

This simplified model can be used for approximate analytical calculations. The corresponding transfer function is

$$G_1(s) = \frac{k_i k_m k_{\omega 1} \left[J_2 s^2 + k \right]}{s \left[J_1 J_2 s^2 + k(J_1 + J_2) \right]}$$

The resonant poles have moved to the imaginary axis, and the rigid body pole is now at the origin.

3.2 NONLINEAR MODEL INCLUDING STATIC FRICTION

Friction forces in mechanical systems are normally considered to consist of three parts: viscous friction, static friction, and Coulomb friction [2]. The viscous friction is proportional to the speed and has already been included in the linear model (3.1). The static friction is sometimes called "stiction", and it is assumed to exist at zero speed only. If the net torque acting on a shaft is less than the maximum static friction, the net torque will be balanced by the static friction. Thus the total torque is zero and the shaft remains at rest. The Coulomb friction is a constant braking force that depends only on the direction of rotation.

The combined effect of static friction and Coulomb friction is modelled with a nonlinear friction torque M_{fi} acting on shaft i (i = 1,2). It is described by

$$M_{fi} = \begin{cases} -F_i \cdot \mathrm{sign}(\omega_i) & ; \ (\omega_i \neq 0) \\ -M_i & ; \ (\omega_i = 0, \ |M_i| < F_i) \\ -F_i \cdot \mathrm{sign}(M_i) & ; \ (\omega_i = 0, \ |M_i| > F_i) \end{cases} \tag{3.8}$$

where F_i is the magnitude of the friction torque and M_i is the total torque acting on shaft i (excluding the nonlinear friction) [1]. Note that the friction model (3.8) is symmetric, i.e. the friction torque is the same in both directions of rotation. The model can easily be modified to cover the unsymmetric case, but for simplicity only the symmetric case is considered here. It is also assumed that the magnitudes of the static friction and the Coulomb friction are equal, a simplification that may be unrealistic. It is necessary for practical reasons, however, since it is difficult to estimate the static friction torque. The torque balance equations for the nonlinear model are

$$\begin{cases} J_1 \cdot \dot{\omega}_1 = k(\theta_2 - \theta_1) - d_1\omega_1 - d(\omega_1 - \omega_2) + k_m k_i u + M_{f1} \\ J_2 \cdot \dot{\omega}_2 = -k(\theta_2 - \theta_1) - d_2\omega_2 + d(\omega_1 - \omega_2) + M_{f2} \end{cases} \tag{3.9}$$

and will be used in the nonlinear simulation model to investigate the effects of the nonlinear friction terms on the performance of the closed loop system.

Parameter values

The magnitude of the Coulomb friction torque was estimated by measuring the input voltage required to run the servo system in steady state at lowest possible speed. At low speeds, the viscous friction torques can be neglected, and the input signal is proportional to the sum of the Coulomb friction on shaft 1 and 2. Since

the two DC motors on shaft 1 and 2 are identical, we assume that $F_1 = F_2$. The estimated Coulomb friction torque values were:

$$F_1 = F_2 = 5 \cdot 10^{-4} \text{ Nm}$$

These friction values were verified by simulating the initial value disturbance experiments described earlier in the section on parameter values for the linear model.

4. LINEAR CONTROL DESIGN

4.1 CONTINUOUS STATE FEEDBACK AND OBSERVER

The control design will be based on pole placement with state feedback. The reasons for using the state space approach are as follows:

(1) We already have a state-space model of the system.

(2) The method is straight-forward and easy to understand.

(3) The absence of zeros in the transfer function from u to y_2 (3.6) show that pole placement is sufficient to obtain a well damped inertial load speed step response, which is our main control objective. A state feedback control law should be an appropriate method to achieve the desired pole placement.

We can only measure one state, ω_1, so the other state values must be estimated. Since the measured signal is distorted by ripple from the tacho generator, we estimate the full state vector including ω_1. This means that the observer is also used as a filter to decrease the influence of the noise on y_1. The observer in the control algorithm produces an estimate of ω_2 as a spin-off result. This is valuable since ω_2 is the variable that we actually want to control, although in many applications it cannot be measured.

In certain applications, integral action may be required to guarantee zero steady state error. This can be done by augmenting the state space model with an extra state corresponding to the integral term. For the sake of simplicity, integral action in the controller will not be considered here. The theory of state feedback control and observers is treated in most modern textbooks on control of linear dynamic systems, e.g. [8].

State feedback design

The linear open loop system (3.4) has two poorly damped oscillatory modes with the natural frequency

$$\omega_o = 11.2 \text{ rad/s}$$

and one slow real pole at

$$s = -0.12$$

The system is controllable, and consequently the closed loop poles can be placed

arbitrarily with state feedback. Introduce the control law

$$u = l_r y_r - L\hat{x} \quad ; \quad L = (l_1 \; l_2 \; l_3) \tag{4.1}$$

where $\hat{x}$ is the estimated state vector. L is chosen to give the closed loop system desired poles, and l_r is chosen to give a steady state gain of 1 from the reference value y_r to the process output.

The choice of desired poles can be made as follows. In order to achieve active damping of the resonant modes we move the resonant poles to a well-damped location (ζ=0.7) while approximately keeping their natural frequency ω_o. To improve the system bandwidth the slow real pole is moved to the same distance from the origin as the complex poles. The desired closed loop system thus has the characteristic equation

$$(s+\alpha_{cl})(s^2+2\zeta_{cl}\omega_{cl}s+\omega_{cl}^2) = 0$$

where we choose $\alpha_{cl} = \omega_{cl} = 12$ and $\zeta_{cl} = 0.7$.

The pole placement philosophy of keeping ω_o constant while changing ζ can be interpreted as introducing "electronic damping" in the system. By keeping ω_o constant the spring constant in the system is not modified, and hence the regulator does not have to produce any "reaction force" when the system is at rest. This is demonstrated by a simple example.

EXAMPLE 4.1

Consider a second order system described by

$$m\frac{d^2y}{dt^2} + d\frac{dy}{dt} + ky = u$$

Introduce a control law with proportional and derivative action:

$$u = -K_p y - K_d \dot{y}$$

With an appropriate state space realisation, this control law can also be interpreted as state feedback. The derivative gain K_d will add to the damping coefficient d, and thus provide active "electronic" damping of the resonant modes. A control design with constant natural frequency implies that K_p=0. This means that the control signal u will be zero in steady state. □

Note that the pole placement may have to be modified in order to fulfil specifications on step response (e.g. rise time, overshoot, etc.) or other performance measures. The choice presented above should be considered as a nominal design. With this design we get the following controller gains:

$$L = 10^{-2}\begin{bmatrix}2.49 & 6.86 & -19.2\end{bmatrix}$$

$$l_r = 0.95$$

The controller gains are very small. More reasonable gain values can be obtained by scaling the state variables and transforming the state space model into a realisation better suited for implementation. For further details see Chapter 7, which treats implementation issues.

Observer design

The state vector estimate $\hat{x}$ is calculated with the observer

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \quad ; \quad K = (k_1 \; k_2 \; k_3)^T \tag{4.2}$$

where the observer gain vector K is chosen such that the reconstruction error

$$\tilde{x} = x - \hat{x}$$

converges to zero at a desired rate. The process model (3.4) is observable, which implies that the convergence rate can be chosen arbitrarily. Let us assume that the characteristic equation for the reconstruction error is

$$(s+\alpha_{ob})(s^2+2\zeta_{ob}\omega_{ob}s+\omega_{ob}^2) = 0$$

The observer is normally designed so that its dynamics are faster than the closed loop system. This implies that initial state estimate errors will decay faster than the closed loop dynamics. Further design considerations include a trade-off between convergence rate and noise sensitivity. As a nominal design we place the observer poles in the same pattern as the closed loop system, but at a distance $\alpha\omega_{cl}$ from the origin. The factor α is given the nominal value $\alpha = 1.5$. This implies that $\zeta_{ob} = 0.7$, $\omega_{ob} = \alpha_{ob} = 18$, and the corresponding K vector is

$$K = \begin{bmatrix}427 & 467 & 59.6\end{bmatrix}^T$$

The observer gain values are quite large. More reasonable values can be obtained by proper scaling of the state variables. See Chapter 7 for further details.

Design based on simplified model

When the linear damping coefficients are small, the open loop poles and zeros are very close to the imaginary axis (cf. Figure 3.2). Since the closed loop poles are moved to a well damped location, we can neglect the open loop damping in the control design. This means that the design is based on the simplified linear model (3.7). The regulator gains obtained with this design are quite close to the values given above. The closed loop poles of the full state space model (3.4) with a regulator designed from the simplified model (3.7) are also very close to the desired closed loop poles.

4.2 DIGITAL CONTROL DESIGN

The continuous state space model (3.4) was sampled with the sampling interval h = 40 ms, corresponding to approximately four samples per rise time with a closed loop natural frequency ω_{cl}=12 rad/s. A discrete time control law based on state feedback and an observer which uses the latest measured output value can be implemented as

$$\begin{cases} \hat{x}(k|k) = \hat{x}(k|k-1) + K(y(k)-C\hat{x}(k|k-1)) \\ u(k) = l_r y_r(k) - L\hat{x}(k|k) \\ \hat{x}(k+1|k) = \Phi\hat{x}(k|k) + \Gamma u(k) \end{cases} \tag{4.3}$$

where $\hat{x}(k|k-1)$ denotes the estimate of x(k) based on measurements available at time k-1. For further details, see [18]. Select L to get desired closed loop poles, K to get desired observer dynamics, and l_r to give a steady state closed loop gain of one. The desired continuous closed loop and observer poles (with $\omega_{cl}=\alpha_{cl}$=12, ζ_{cl}=0.7, $\omega_{ob}=\alpha_{ob}$=18 and ζ_{ob}=0.7) are transformed into corresponding discrete poles in the z-plane with the mapping

$$z = e^{sh}$$

With this design the controller gains will be

$$L = 10^{-2}\begin{bmatrix} 1.68 & 3.75 & -8.69 \end{bmatrix}; \quad l_r = 0.554$$

$$K = \begin{bmatrix} 8.186 & 8.155 & 1.056 \end{bmatrix}^T$$

The regulator gain values are different than in the continuous time case. This

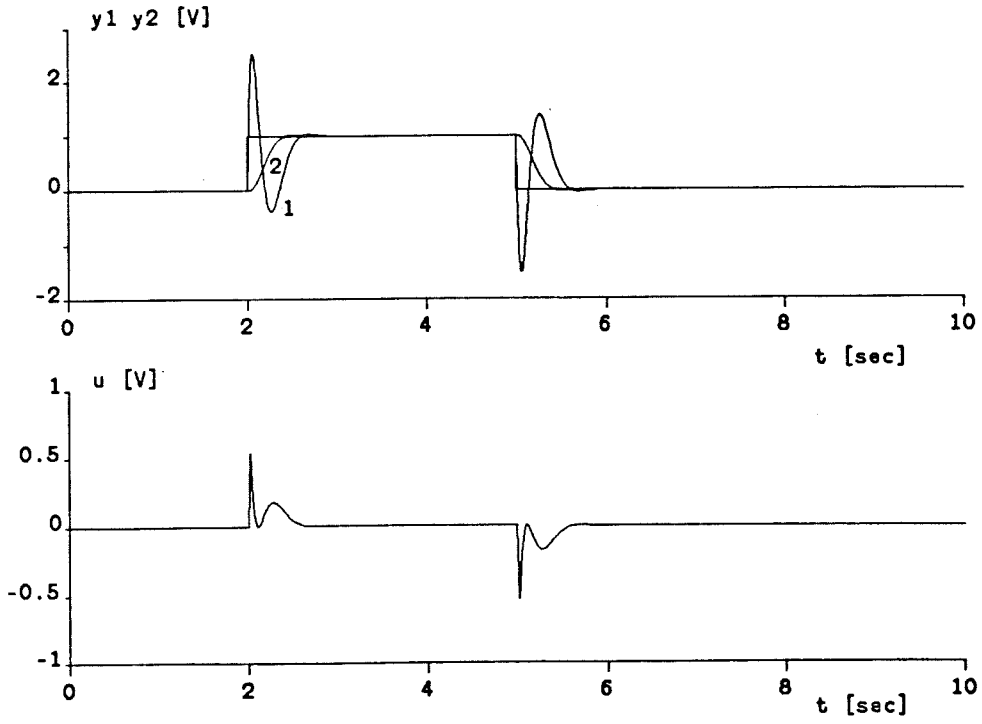


Fig. 4.1 Closed loop step response, linear model (ω_{cl} = 12).

depends on the comparatively slow sampling rate, which makes it necessary to use sampled data theory in the control design. High frequency noise on the measured process output signal normally requires the use of an analog prefilter to avoid aliasing effects. The dynamics of this prefilter then have to be considered in the control design. The analog prefilter has been omitted here for simplicity.

4.3 SIMULATIONS

To evaluate the performance of the control design, the closed loop step response was simulated with SIMNON [6], [7]. The reference step amplitude, y_r = 1 V = 10 rad/s, was selected to keep the maximum torsion of the string within given bounds. In the simulations with the full nonlinear servo model (cf. Section 3.2), a non-zero initial condition $\omega_1(0)$ = 1.0 rad/s was used to emulate small initial start-up disturbances.

Linear model

A simulation of the step response with the linear servo model and the nominal continuous control design (cf. Section 4.1) is shown in Figure 4.1. Note that the

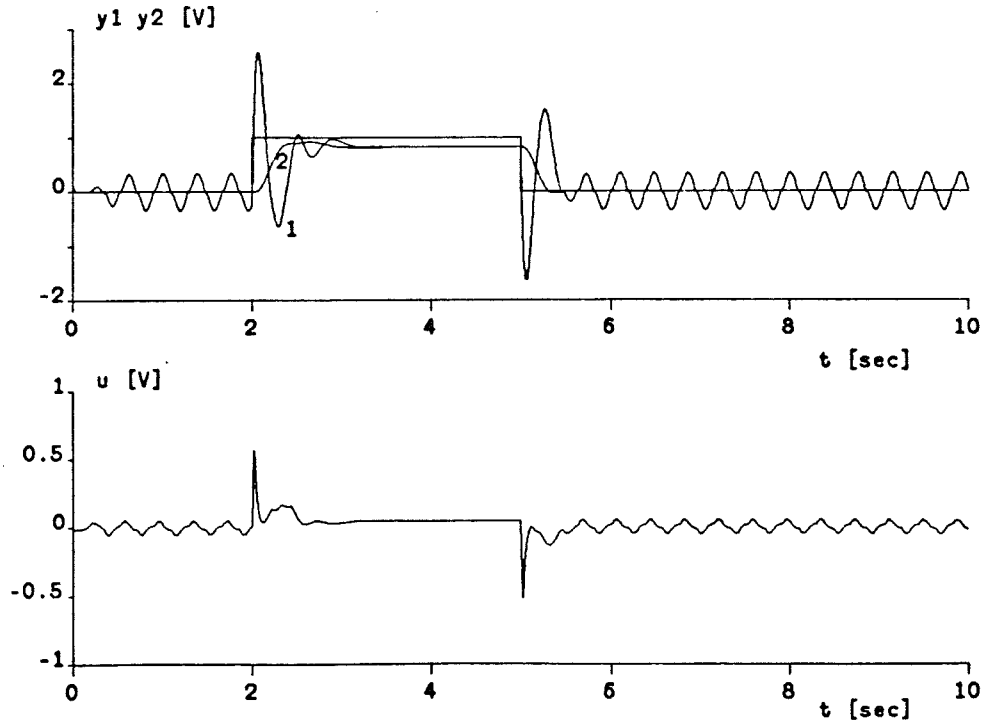


Fig. 4.2 Closed loop step response, nonlinear model (ω_{cl} = 12).

drive motor speed y_1 exhibits a significant overshoot, whereas the step response of y_2 is well damped just as we wish. This depends on the zeros in the open loop transfer functions G_1 and G_2 from the input u to y_1 and y_2 respectively (cf. eqn (3.5) and (3.6)), and on the fact that the state feedback control law only moves the poles of the system. Thus the closed loop system has the same zeros as the open loop system. G_1 has two complex zeros which cause the large overshoot in the y_1 step response, whereas G_2 has no zeros and thus y_2 has a well damped closed loop step response in agreement with the desired behaviour.

Nonlinear model. Limit cycle oscillations

The closed loop step response was also simulated with the nonlinear servo model including Coulomb friction (cf. Chapter 3), and the result is shown in Figure 4.2. A limit cycle oscillation is clearly visible when the reference value is zero. This is a nonlinear effect, which cannot be explained with linear models. By simulating the closed loop system with the Coulomb friction on either shaft 1 or shaft 2 set to zero, it was found experimentally that the limit cycle oscillations are caused by the nonlinear friction in the drive motor. The oscillations appear only when the speed reference is zero. This can be explained by the discontinuity in the Coulomb friction at zero speed. At non-zero speed values, the Coulomb friction

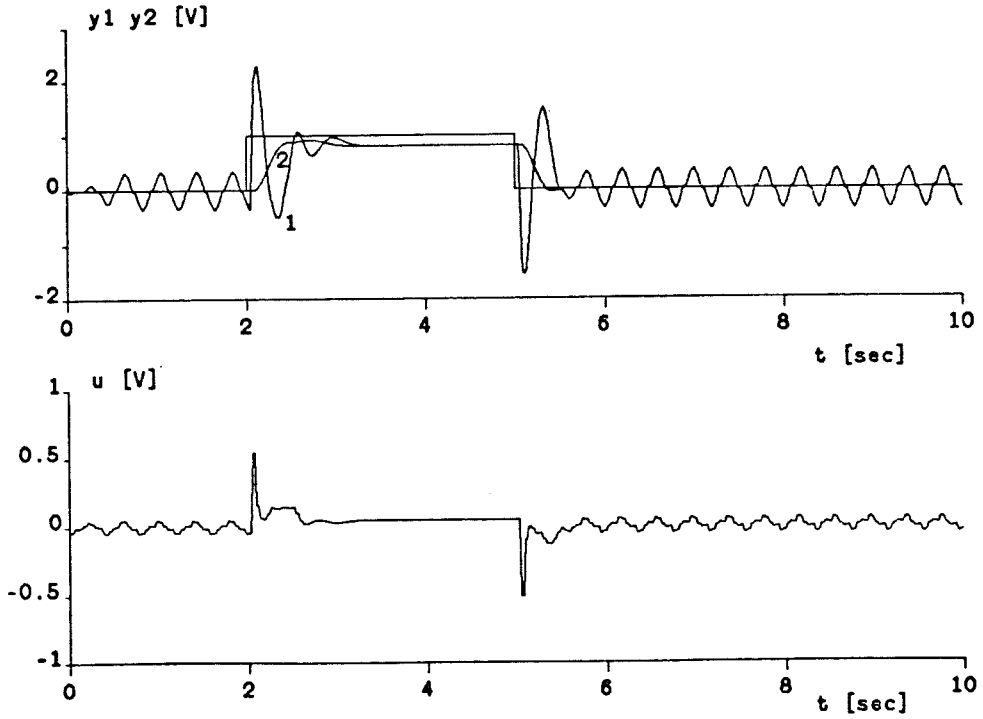


Fig. 4.3 Closed loop step response, nonlinear model with digital control (ω_{cl} = 12).

can be regarded as a constant load disturbance. Hence it does not affect the stability of the system as long as the direction of rotation does not change.

Digital control

The closed loop step response with the digital controller described in Section 4.2 and the nonlinear servo model is shown in Figure 4.3. There is no significant difference in transient response, and the limit cycle oscillation has approximately the same amplitude and frequency as in the simulation with a continuous controller (Figure 4.2).

Reduced bandwidth eliminates limit cycles

The limit cycle oscillation can be eliminated by reducing the specified closed loop bandwidth, i.e. by reducing ω_{cl}. This is shown in Figure 4.4, where we have simulated a control design with $\omega_{cl} = \alpha_{cl} = 8$. The conclusion is that the limit cycle oscillations caused by the Coulomb friction in the drive motor impose a limit on the achievable closed loop bandwidth. This is a serious constraint, which motivates further studies of the limit cycle phenomenon.

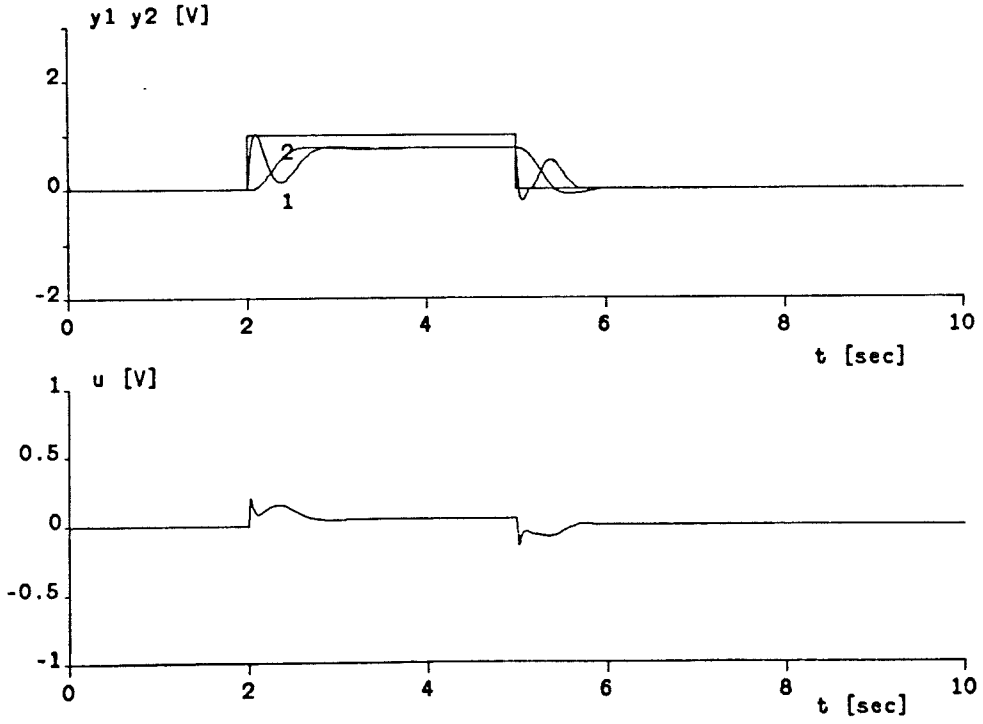


Fig. 4.4 Closed loop step response with nonlinear model. Reduced bandwidth design (ω_{cl} = 8).

5. NONLINEAR ANALYSIS

5.1 THE DESCRIBING FUNCTION METHOD

The describing function method [2] is an approximate method for the analysis of periodic oscillations in nonlinear systems. In its standard version it requires that the system can be decomposed into one linear and one nonlinear subsystem as shown in Figure 5.1. It is assumed that the input to the nonlinearity is a sinusoid with amplitude C. The nonlinearity is replaced by an amplitude dependent gain, $Y_N(C)$, which is called the describing function of the nonlinearity. To be more precise, the describing function is defined as the complex ratio of the fundamental component of the output from the nonlinearity to the sinusoidal input.

A stability criterion

A fundamental problem is to investigate whether a periodic oscillation, or limit cycle, can exist in the closed loop system. This can be determined with a stability criterion analogous to the classical Nyquist criterion for linear systems:

> Assume a sinusoidal input to the nonlinearity (NL) with amplitude C. If the Nyquist curve $G(j\omega)$ of the linear subsystem (L) encircles the point $-1/Y_N(C)$, the amplitude C of the oscillation will increase. If $G(j\omega)$ does not encircle $-1/Y_N(C)$, then C will decrease.

The stability criterion is based on the following major assumptions [2], [9], [15]:

(1) The system is autonomous, i.e. unforced and time invariant.

(2) The nonlinearity (NL) is frequency independent.

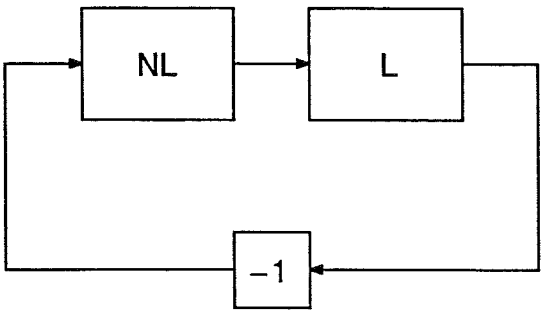


Fig. 5.1 Block diagram for basic nonlinear system decomposed into linear (L) and nonlinear (NL) subsystem.

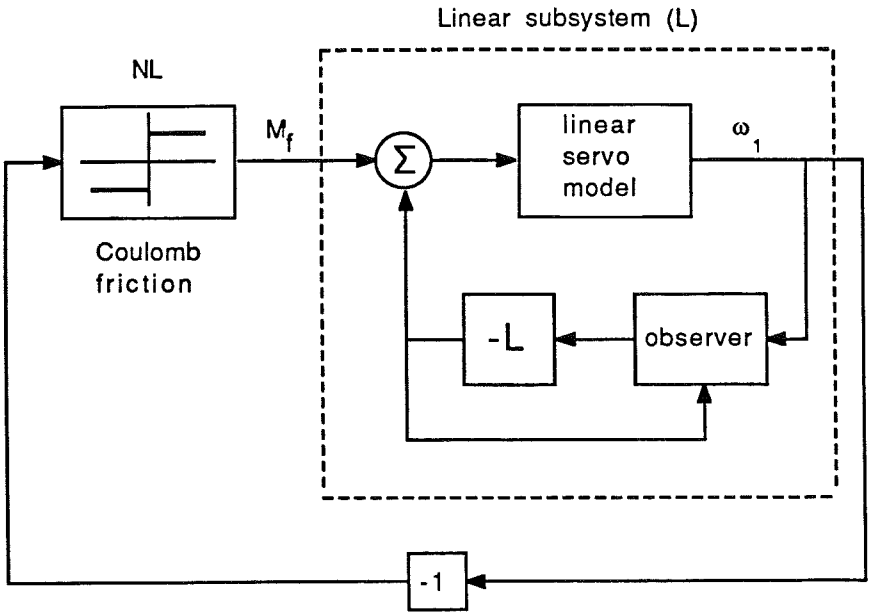


Fig. 5.2 Servo system with speed control divided into linear and nonlinear subsystem.

(3) The linear subsystem (L) is stable and has low-pass character. This implies that the higher harmonics at the output from the nonlinearity are filtered so that the input to the nonlinearity is approximately sinusoidal.

(4) The nonlinearity (NL) does not introduce any subharmonic oscillations, i.e. its output does not contain any components with frequencies lower than the frequency of the sinusoidal input.

Assumption (4) is difficult to verify. It depends on the properties of the nonlinearity in combination with the linear subsystem.

Application to the flexible servo system

The closed loop flexible servo system can be decomposed according to Figure 5.1 by neglecting the friction on shaft 2 and approximating the Coulomb friction on shaft 1 with an ideal relay function. To make the system autonomous, we let the speed reference y_r be zero. The decomposed system is shown in Figure 5.2. With the torque on shaft 1 as the input u, and the Coulomb friction torque M_f regarded as an external input, the open loop servo system can be described by the state space equation

$$\begin{cases} \dot{x} = Ax + B(M_f+u) \\ y = Cx \end{cases} \tag{5.1}$$

where A and C are defined by equation (3.1) and

$$B = \left[\frac{1}{J_1} \quad 0 \quad 0\right]^T$$

Let the regulator have the structure described in Chapter 4 (cf. equations (4.1) and (4.2)):

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K(y-C\hat{x}) \\ u = l_r y_r - L\hat{x} \end{cases} \tag{5.2}$$

With $y_r = 0$, the linear subsystem (L) is described by the state space equation

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BL \\ KC & A-BL-KC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} M_f$$

$$y = (C \; 0) \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \tag{5.3}$$

Let us assume that the nonlinear friction M_f is modelled by

$$M_f = \begin{cases} F & \omega<0 \\ -F & \omega>0 \end{cases} \tag{5.4}$$

where F is the magnitude of the Coulomb friction torque (cf. equation (3.8)). This nonlinearity fulfills assumption (3) for the stability criterion, and the describing function for the nonlinear subsystem (cf. Figure 5.1) is

$$Y_N(C) = \frac{4F}{\pi C}$$

This implies that the point

$$-\frac{1}{Y_N(C)} = -\frac{\pi C}{4F}$$

will move along the negative real axis as C varies. The Nyquist curve for the linear subsystem (5.3) with nominal control design, i.e. $\omega_{cl} = 12$, is shown in

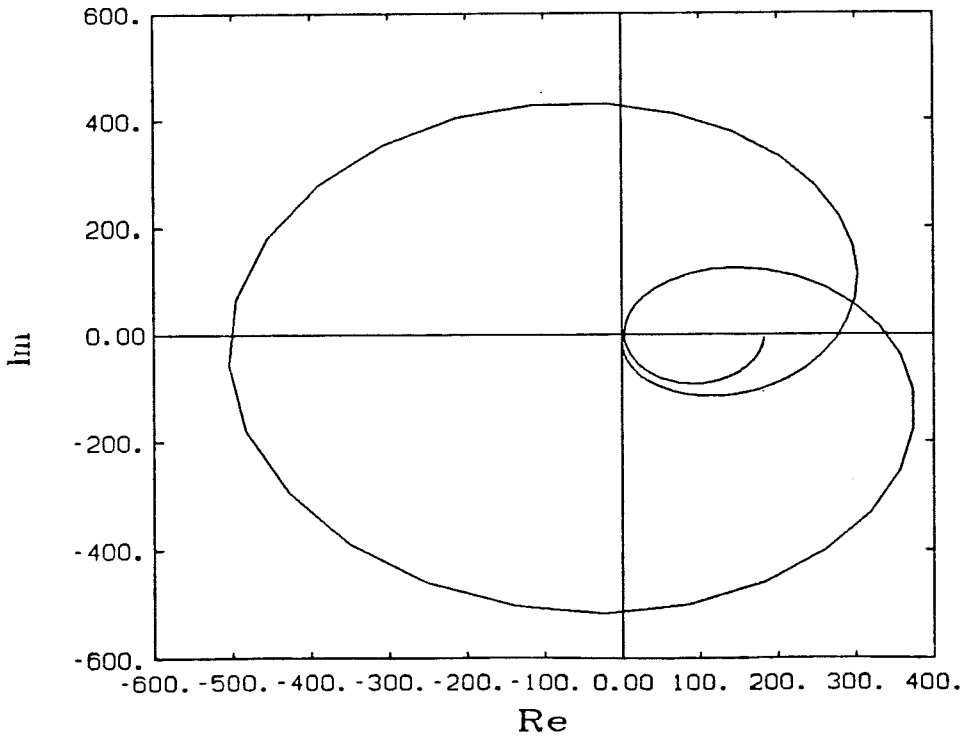


Fig. 5.3 Nyquist curve for the linear subsystem (5.3) with $\omega_{cl} = 12$.

Figure 5.3. It intersects the negative real axis approximately at $G(j\omega) = -500$. The oscillation amplitude C is given by

$$G(j\omega) = -\frac{1}{Y_N(C)} = -\frac{\pi C}{4F} \quad \Rightarrow \quad C = \frac{-4F \cdot G(j\omega)}{\pi}$$

With $F = 5 \cdot 10^{-4}$ Nm the amplitude is C = 0.3 Volt. The oscillation frequency ω can be calculated from the condition

$$\arg G(j\omega) = -180^{\circ}$$

which in our case gives $\omega = 15.8$ rad/s or $f \approx 2.5$ Hz.

Note that the describing function method is based on an approximation, and consequently does not always give correct results. In this case, however, the method predicts a limit cycle with amplitude and frequency in good agreement with the simulation results. Thus we can state with some confidence that a very simple friction model (5.4) in combination with the describing function method is enough to provide a theoretical explanation of the observed limit cycles. The absence of limit cycles with the lower bandwidth design ($\omega_{cl} = 8$) is explained by the corresponding Nyquist curve (see Figure 5.4), which never intersects the negative real axis and thus by the describing function stability criterion the

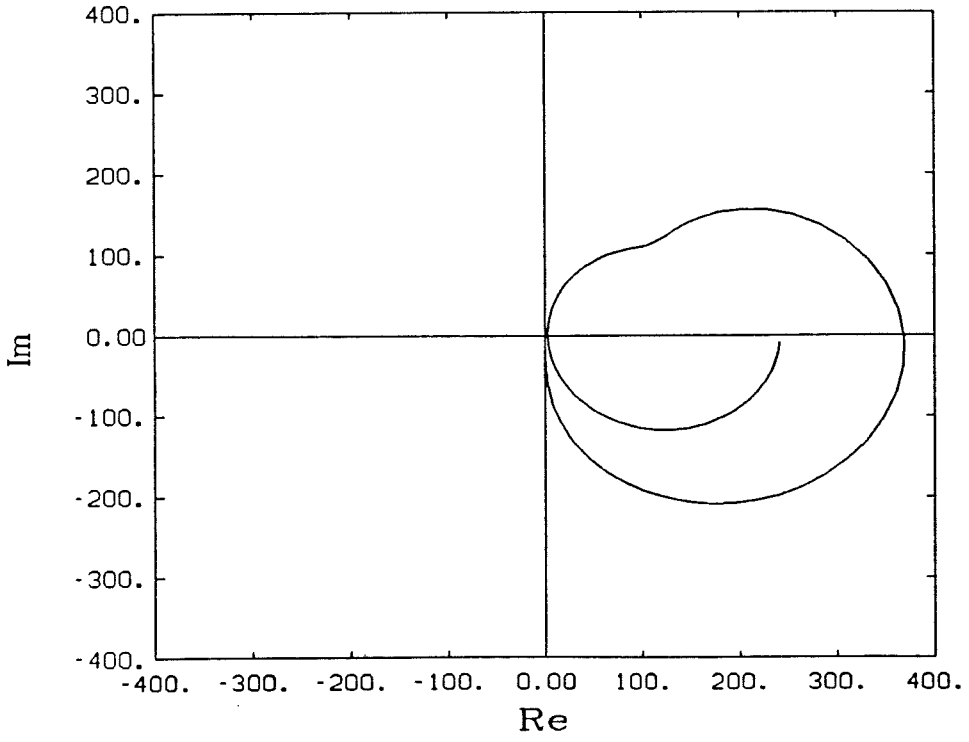


Fig. 5.4 Nyquist curve for the linear subsystem (5.3) with $\omega_{cl} = 8$.

system is stable.

Tsypkin's method

There are also exact methods to analyze oscillations in relay systems, for example Tsypkin's method [2], [16]. In this method a specific waveform is assumed at the output of the nonlinearity. The output signal from the nonlinearity is expressed as a Fourier series, and the output of the linear subsystem is calculated by summing the outputs corresponding to each term in the Fourier series. The satisfaction of certain switching conditions then gives the oscillation frequency. With the relay friction model (5.4), Tsypkin's method gives an oscillation frequency ω = 16.3 rad/s in the nominal design case. This is close to the approximate value (15.8 rad/s) obtained with the describing function method.

5.2 A SUFFICIENT CONDITION FOR INSTABILITY

An interesting observation is that the presence of limit cycles seems to be related to the stability of the controller. The controller obtained with nominal design (ω_{cl} = 12) is unstable, whereas the reduced bandwidth design (ω_{cl} = 8) gives a stable controller. This is not just a coincidence. It will be shown that an unstable

controller is in fact a sufficient condition for limit cycles to appear according to the describing function criterion. For this purpose we will first prove the following lemma.

LEMMA 5.1

Consider a system with a rational transfer function G(s). Assume that the system is strictly proper, asymptotically stable and that the steady state gain G(0) is finite and positive. Also assume that G(s) has no zeros on the imaginary axis. Then the Nyquist curve G(jω) must intersect the negative real axis if G(s) is non minimum-phase, i.e. if G(s) has zeros in the right half plane.

Proof:

Let the process transfer function be

$$G(s) = \frac{B(s)}{A(s)}$$

By assumption, the denominator A(s) is stable i.e. it has all its zeros in the left half plane. Factor the numerator polynomial B(s) as

$$B = B_1 B_2$$

where B_1 has all its zeros in the right half plane and B_2 has all its zeros in the left half plane. Rewrite the transfer function G(s) as

$$G = \frac{B_1}{B_1^*} \cdot \frac{B_1^* B_2}{A} = G_1 \cdot G_2$$

where the zeros of B_1^* are the zeros of B_1 reflected in the imaginary axis. Notice that $G_1(j\omega)$ is an all-pass filter with unity gain. If G(s) has zeros in the right half plane, then $\deg(B_1) > 0$ and the high frequency phase shift of $G_1(j\omega)$ is

$$\arg(G_1(j\omega)) = -n \cdot \pi + \varepsilon$$

where $n = \deg(B_1)$ and $\varepsilon \to 0$ as $\omega \to \infty$. The remaining part of the transfer function, G_2, is minimum-phase and asymptotically stable. It is also strictly proper since $\deg(B_1^*) = \deg(B_1)$. Consequently, for high frequencies the phase shift of $G_2(j\omega)$ is

$$\arg(G_2(j\omega)) = -m \cdot \frac{\pi}{2} + \delta$$

where $m = \deg(A) - \deg(B)$ and $\delta \to 0$ as $\omega \to \infty$. Thus we can conclude that if G(s) is non minimum-phase, then

$$\lim_{s \to \infty} \arg(G(j\omega)) = \varphi_\infty = -(2n+m) \cdot \frac{\pi}{2} \leq \frac{-3\pi}{2}$$

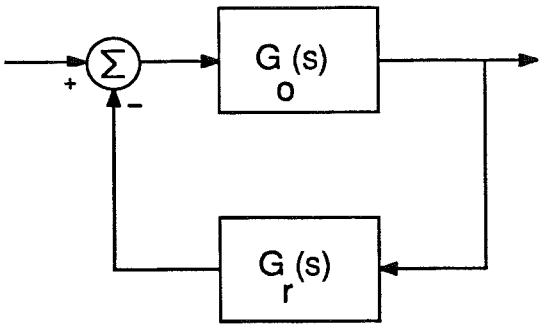


Fig. 5.5 Block diagram of a linear system with a process $G_o(s)$ and a regulator $G_r(s)$ in the feedback loop.

In the absence of zeros on the imaginary axis, $\varphi(\omega) = \arg(G(j\omega))$ is a continuous function of ω. We know that $\varphi(0) = 0$, since G(s) is assumed to have a positive steady state gain, and that the asymptotic phase shift is $\varphi_\infty \leq -3\pi/2$. Then there must exist a finite frequency ω_o such that

$$\varphi(\omega_o) = -\pi$$

and hence $G(j\omega)$ intersects the negative real axis at $\omega = \omega_o$. □

Remark:

The lemma can be extended to cover some transfer functions with non-rational elements, for example a pure time delay. In that case the assumptions can be relaxed to the rational part being a proper transfer function, since the non minimum-phase element has an infinite phase shift when $\omega \to \infty$. The assumption $G(0) > 0$ can then also be omitted. □

Application to the flexible servo system

The linear part of the closed loop flexible servo system (cf. Figure 5.2) has the structure shown in Figure 5.5, with one process and one controller block. The process transfer function $G_o(s)$ from the torque on shaft 1 to y_1 can be calculated from (5.1):

$$G_o(s) = \frac{B_o(s)}{A_o(s)} = C(sI-A)^{-1}B = \frac{4544(s^2+0.07s+16)}{(s+0.1)(s^2+0.4s+125)} \qquad (5.5)$$

This transfer function is similar to (3.5). The only difference is in the steady state gain, since here we consider the torque on shaft 1 as the input signal. With zero speed reference, the controller transfer function $G_r(s)$ can be calculated from (5.2):

$$G_r(s) = \frac{B_r(s)}{A_r(s)} = L(sI-A+BL+KC)^{-1}K \tag{5.6}$$

The closed loop transfer function is

$$G = \frac{B_o A_r}{A_o A_r + B_o B_r} = \frac{Q}{P} \tag{5.7}$$

Note that the controller poles appear as zeros in the closed loop system. The numerator and denominator of the closed loop system G have the following polynomial degrees:

$$\deg(Q) = \deg(B_o)+\deg(A_r) = 5$$

$$\deg(P) = \deg(A_o)+\deg(A_r) = 6$$

Hence the closed loop system is strictly proper. The closed loop poles are always asymptotically stable (for any sensible control design!), and the two complex zeros of $B_o(s)$ are in the left half plane. The physical properties of the system imply that the closed loop steady state gain is positive, since a positive torque input will yield a positive speed output. Now consider the case of an unstable controller. The transfer function G(s) then has at least one zero, but no poles, in the right half plane. From Lemma 5.1 we can conclude that the Nyquist curve of the closed loop linear subsystem must intersect the negative real axis. Assuming that the nonlinear friction can be modelled by an ideal relay function (5.4), the describing function stability criterion in combination with Lemma 5.1 says that an unstable controller will always give rise to limit cycle oscillations. This is a sufficient but not necessary condition, since the Nyquist curve may intersect the negative real axis even with a stable controller. We must also keep in mind that the describing function method is based on an approximation (cf. Section 5.1 above).

Numerical calculations show that the closed loop natural frequency stability limit for the regulator is $\omega_{cl} = 9.9$ rad/s. Figure 5.6 shows the deformation of the Nyquist curve as ω_{cl} approaches and exceeds the stability limit. Note that the Nyquist curve does not intersect the negative real axis when ω_{cl} is below the stability limit.

In addition to the relation between regulator instability and the appearance of limit cycle oscillations, there are also other difficulties in having an unstable regulator. See for example [10, pp. 350-351] and the discussion in Section 7.1.

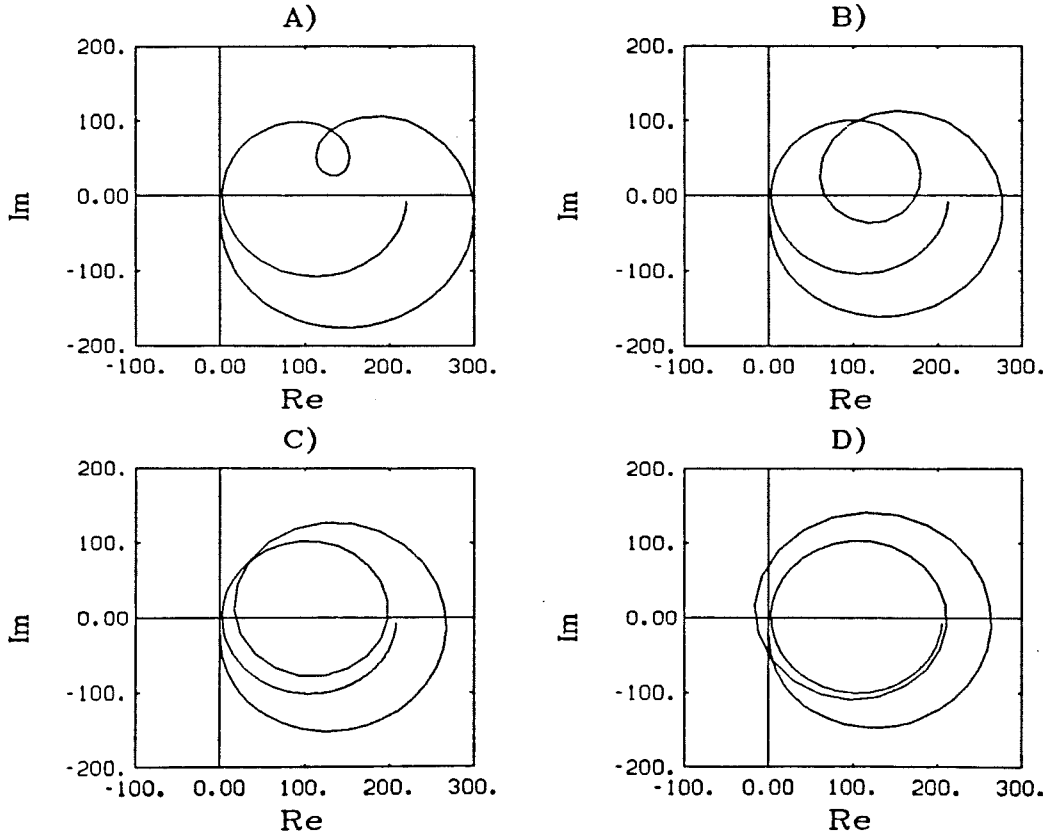


Fig. 5.6 Nyquist curve for the linear subsystem (5.3) with varying ω_{cl}: A) ω_{cl}=9, B) ω_{cl}=9.5, C) ω_{cl}=9.8, D) ω_{cl}=10.

5.3 A SUFFICIENT STABILITY CONDITION

A special case of the Circle Criterion [17] can be used to prove asymptotic stability for the closed loop nonlinear system. However, this approach requires stronger assumptions about the nature of the nonlinear friction.

LEMMA 5.2

Consider a nonlinear system described by the equations

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \\ u(t) = -\Phi(t,y(t)) \end{cases} \tag{5.8}$$

where A, B and C are matrices with the dimensions (nxn), (nx1), and (1xn), respectively. Let all eigenvalues of A have negative real parts, and the linear subsystem described by (A, B, C) be controllable and observable. It is further assumed that the nonlinear function $\Phi(t,\cdot)$ is continuous and satisfies

$$\begin{cases} \Phi(t,0) = 0, \quad \forall\ t \geq 0 \\ \sigma\Phi(t,\sigma) \geq 0, \quad \forall\ \sigma \in R,\ \forall\ t \geq 0 \end{cases} \tag{5.9}$$

Then a sufficient condition for the solution $x(t) = 0$ to be globally asymptotically stable is that the transfer function

$$G(s) = C(sI-A)^{-1}B$$

satisfies the inequality

$$\mathrm{Re}[G(j\omega)] > 0, \quad \forall\ \omega \in R \tag{5.10}$$

Proof (outline):

This lemma is a special case of the Circle Criterion [17]. The proof is based on Lyapunov's direct method and the Kalman-Yacubovitch Lemma, using a Lyapunov function of the form $V(x) = x'Px$. For further details, see [17, Section 5.5]. □

Remarks:

(i) Note that Lemma 5.2 can be applied to time-varying nonlinearities. It is important that the nonlinear function $\Phi(t,\sigma)$ is continuous, however, since Lyapunov's direct method requires that the Lyapunov function $V(x)$ is continuously differentiable. If this is not the case, the problem becomes much more involved.

(ii) This stability criterion is quite conservative. There are many systems where the linear subsystem (A, B, C) does not satisfy (5.10), but where the closed loop nonlinear system (5.8) is stable. One example can be found in Section 6.2 (see Figure 6.3). □

The simple relay friction model (5.4) does not satisfy the conditions in Lemma 5.2, since it does not have a defined value at the origin. It may be argued, however, that instantaneous action is not possible in nature. Therefore, it would be reasonable to replace (5.4) with a continuous function which approximates an ideal relay function, for example a high gain with saturation. This nonlinearity satisfies (5.9), and it can approximate the ideal relay characteristic to any desired accuracy by selecting a sufficiently high gain. Furthermore, the only requirement on the nonlinearity in Lemma 5.2 is that it satisfies (5.9). Hence Lemma 5.2 in fact covers a whole class of nonlinear functions, and it can be used to prove stability even if the exact friction model is unknown or time-varying.

Application to the flexible servo system

To apply Lemma 5.2 to the flexible servo case, we must first verify that the linear subsystem (5.3) satisfies the conditions in the Lemma. With a reduced

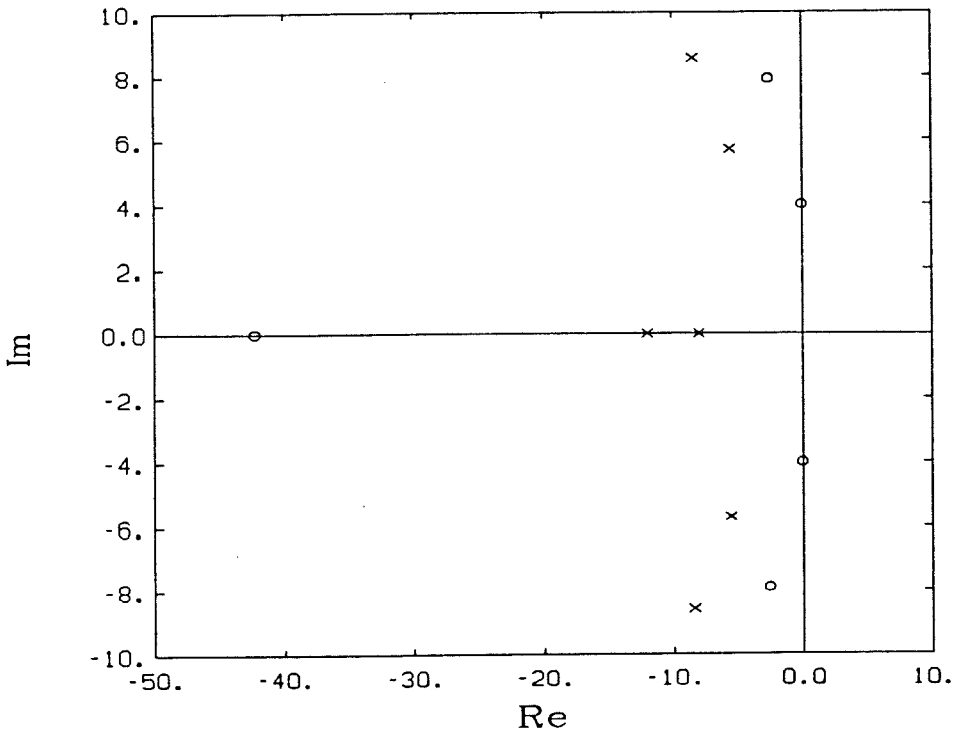


Fig. 5.7 Singularity diagram for the linear subsystem (5.3) with reduced bandwidth design (ω_{cl} = 8). x = poles, o = zeros.

bandwidth design (ω_{cl} = 8), the linear subsystem is stable, controllable and observable. Figure 5.4 indicates that the linear subsystem also satisfies condition (5.10). Due to the limited resolution of the computer plot, this must be verified with a more detailed examination. Figure 5.7 shows the poles and zeros of the linear subsystem. Note that two zeros are very close to the imaginary axis. They are the zeros of the process, cf. (5.5) and (5.7). Figure 5.8 shows the Bode plot of the linear subsystem (5.3) with ω_{cl} = 8 rad/s. It can be seen in Figure 5.7 that all singularities correspond to resonance frequencies less than 10 rad/s. Thus the Bode plot in Figure 5.8 shows all relevant parts of the phase curve, and it is clear that the phase shift is less than -90° for all frequencies. This is confirmed by an analysis of the sequence of poles and zeros projected on the imaginary axis in Figure 5.7.

We must also confirm that (5.10) is satisfied as $\omega \to \infty$. This can be done by a formal series expansion of G(s) at $s = \infty$:

$$G(s) = \frac{g_1}{s} + \frac{g_2}{s^2} + \ldots \qquad (5.11)$$

Neglecting higher order terms, we get

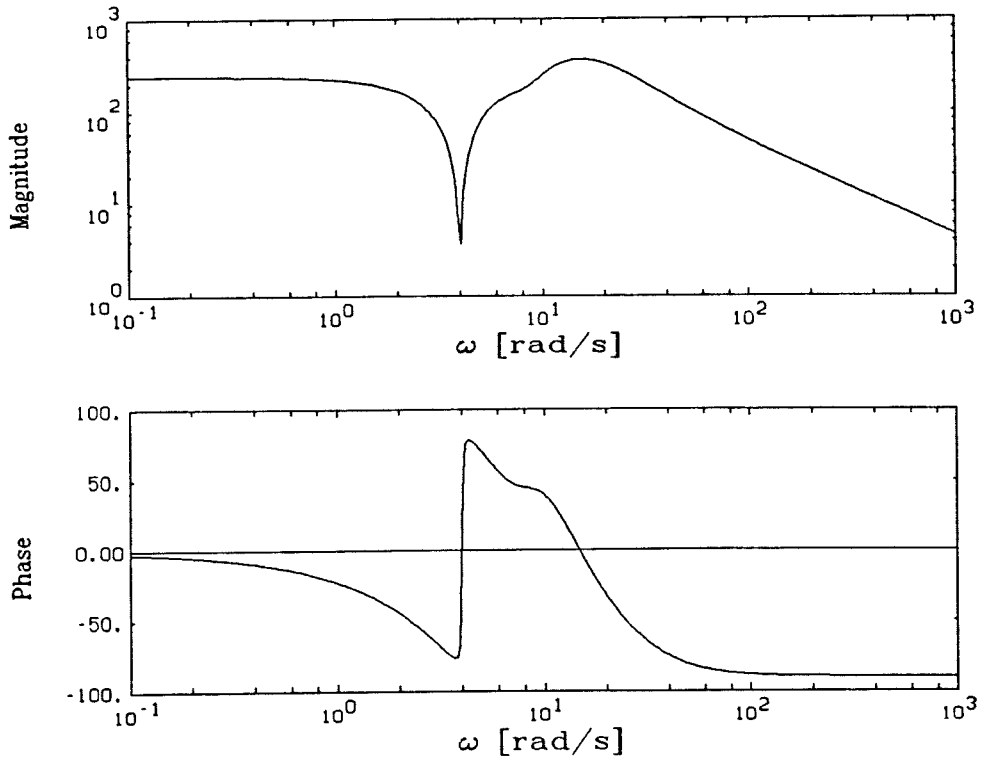


Fig. 5.8 Bode plot for the linear subsystem (5.3) with reduced bandwidth design ($\omega_{cl} = 8$).

$$G(j\omega) \approx \frac{g_1}{j\omega} - \frac{g_2}{\omega^2}$$

Thus (5.10) will be satisfied as $\omega \to \infty$ only if $g_2 < 0$. Let the transfer function G(s) be parameterized as

$$G(s) = \frac{b_1 s^{n-1} + \ldots + b_n}{s^n + a_1 s^{n-1} + \ldots + a_n} \tag{5.12}$$

The series expansion (5.11) then gives

$$\begin{cases} g_1 = b_1 \\ g_2 = b_2 - a_1 b_1 \end{cases}$$

and (5.10) will be satisfied as $\omega \to \infty$ only if

$$a_1 b_1 > b_2 \tag{5.13}$$

In the reduced bandwidth design case ($\omega_{cl} = 8$), a numerical calculation of the transfer function for the linear subsystem (5.3) shows that (5.13) is satisfied.

Thus, if we accept the additional assumptions about the nonlinear friction stated above, the conditions in Lemma 5.2 are satisfied. Hence the closed loop nonlinear system has a globally and asymptotically stable solution $x(t) = 0$. This agrees with the result from the describing function analysis in Section 5.1 and the simulations in Chapter 4.

6. ELIMINATION OF LIMIT CYCLES

6.1 REGULATOR STABILITY CRITERIA

As we have shown in Chapter 5, an unstable regulator is a sufficient condition for the existence of limit cycle oscillations in the flexible servo system. An analytical investigation of how different process and design parameters influence the regulator stability is therefore of great interest, since it may provide valuable clues for avoiding limit cycle oscillations.

Characteristic equation

The regulator designed in Chapter 4 has the following characteristic equation:

$$A_r(s) = \det(sI-A+BL+KC) = 0 \qquad (6.1)$$

The polynomial $A_r(s)$ will be a third order polynomial in s, since the process model is a third order system. Hence

$$A_r(s) = s^3+a_1s^2+a_2s+a_3 \qquad (6.2)$$

Let us assume a control design is made as in Chapter 4. All closed loop poles are placed at a distance ω from the origin and the complex poles have a relative damping ς. The observer poles are placed in the same pattern but at a distance $\alpha\omega$ from the origin ($\alpha > 1$).

The coefficients a_i in $A_r(s)$ will be very complicated functions of the process parameters (J_1, J_2, d_1, d_2, k, etc.) and the control design parameters (ω, ς, α). Closed form expressions can be obtained by using the simplified linear model (3.7) in the control design. The errors introduced in the regulator poles by this simplification are quite small. This is shown in Table 6.1, where the regulator poles corresponding to the nominal design (ω=12, ς=0.7, α=1.5) based on the complete model are shown together with those corresponding to a design based on the simplified model.

Straightforward but tedious calculations, performed with the symbolic manipulation program MACSYMA [13], give the following expressions for the coefficients of the regulator characteristic polynomial $A_r(s)$ based on the simplified servo model:

complete model	simplified model
9.03 + 14.16i	9.35 + 13.93i
9.03 - 14.16i	9.35 - 13.93i
-89.5 + 0.000i	-90.7 + 0.000i

Table 6.1: Regulator poles calculated with complete and simplified servo model. Design parameters: ω=12, ζ=0.7, α=1.5 .

$$a_1 = \omega(2\zeta+1)(\alpha+1) \tag{6.3a}$$

$$a_2 = \frac{1}{J_2^2k^2}\cdot\Big\{-J_1J_2^3\omega^6\alpha^3+J_1J_2^2k\omega^4\left[4\alpha^2\zeta^2+(2\alpha^3+4\alpha^2+2\alpha)\zeta-(\alpha^3+\alpha^2+\alpha)\right] - J_1J_2k^2\omega^2\left[4\alpha\zeta^2+(2\alpha^2+4\alpha+2)\zeta+(\alpha^2+\alpha+1)\right]+k^3(J_2+J_1)\Big\} \tag{6.3b}$$

$$a_3 = \frac{1}{J_2^2k}\cdot\Big\{J_1J_2^2\omega^5\cdot\alpha^2(\alpha+1)(2\zeta+1)-J_1J_2k\omega^3\left[4\alpha(\alpha+1)\zeta(\zeta+1)+\alpha^3+\alpha^2+\alpha+1\right] +k^2\omega(\alpha+1)(2\zeta+1)(J_1+J_2)\Big\} \tag{6.3c}$$

Introducing the dimensionless quantity

$$w = \frac{J_2\omega^2}{k} \tag{6.4}$$

the expressions for a_2 and a_3 can be rewritten as polynomials in w:

$$a_2 = \frac{J_1k}{J_2^2}\cdot\Big\{-\alpha^3w^3+\left[4\alpha^2\zeta^2+(2\alpha^3+4\alpha^2+2\alpha)\zeta-(\alpha^3+\alpha^2+\alpha)\right]w^2 -\left[4\alpha\zeta^2+(2\alpha^2+4\alpha+2)\zeta+(\alpha^2+\alpha+1)\right]w+\left[1+\frac{J_2}{J_1}\right]\Big\} \tag{6.5a}$$

$$a_3 = \frac{J_1k\omega}{J_2^2}\Big\{\alpha^2(\alpha+1)(2\zeta+1)w^2-\left[4\alpha(\alpha+1)\zeta(\zeta+1)+\alpha^3+\alpha^2+\alpha+1\right]w +(\alpha+1)(2\zeta+1)\left[1+\frac{J_2}{J_1}\right]\Big\} \tag{6.5b}$$

Stability criteria

The stability criteria for $A_r(s)$ are

$$\begin{cases} a_1, a_2, a_3 > 0 \\ a_1 a_2 > a_3 \end{cases} \tag{6.6}$$

To evaluate the stability criterion $a_1 a_2 > a_3$, introduce the quantity e defined by

$$e = a_1 a_2 - a_3 \tag{6.7}$$

Insert a_1, a_2 and a_3 from (6.3) into (6.7). Simple but cumbersome calculations (again performed with MACSYMA) using (6.4) give the result

$$e = \frac{J_1 \omega^3}{J_2}\left[c_0 w^2 + c_1 w + c_2\right] \tag{6.8}$$

where

$$c_0 = -\alpha^3(\alpha+1)(2\zeta+1)$$

$$c_1 = \left[8\alpha^2(\alpha+1)\zeta^3 + 4(\alpha^4+4\alpha^3+4\alpha^2+\alpha)\zeta^2 + 4(\alpha^4+2\alpha^3+\alpha)\zeta + \alpha^4+\alpha^3+\alpha^2+\alpha\right]$$

$$c_2 = -\left[8\alpha(\alpha+1)\zeta^3 + 4(\alpha^3+3\alpha^2+3\alpha+1)\zeta^2 + 2(2\alpha^3+3\alpha^2+3\alpha+2)\zeta + \alpha(\alpha+1)\right]$$

Let us now investigate the stability conditions using (6.5) and (6.8). The coefficient a_1 is always strictly positive, and does not affect the regulator stability. The sign of a_2 and e, on the other hand, will be negative when w is large. By calculating and plotting $a_2(w)$, $a_3(w)$ and e(w) for a particular choice of design variables (ζ, α), we can get a more detailed picture of the set of w values for which the regulator is stable. Figure 6.1 shows the normalized and dimensionless coefficients

$$a_{2n}(w) = \frac{J_2^2}{J_1 k} \cdot a_2(w)$$

$$a_{3n}(w) = \frac{J_2^2}{J_1 k \omega} \cdot a_3(w) \tag{6.9}$$

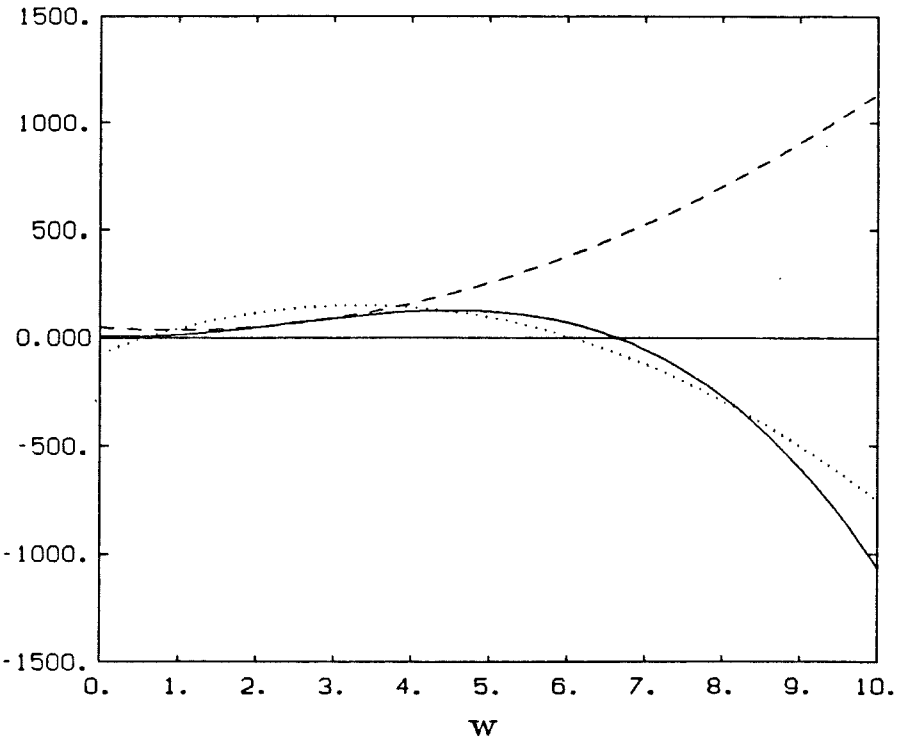


Fig. 6.1 Normalized coefficients of the regulator characteristic polynomial. Nominal design case. Solid line = $a_{2n}(w)$, dashed line = $a_{3n}(w)$ and dotted line = $e_n(w)$.

$$e_n(w) = \frac{J_2}{J_1 \omega^3} \cdot e(w)$$

with nominal design parameters ($\zeta = 0.7$, $\alpha = 1.5$, $J_1 = 22 \cdot 10^{-6}$ kgm^2 and $J_2 = 150 \cdot 10^{-6}$ kgm^2). The normalized coefficients are plotted because they are of the same order of magnitude and thus the sign shifts are more easily observed in the plot.

Bandwidth limit

It can be seen in Figure 6.1 that the stability limit in this case is determined by the condition $e_n(w) \geq 0$. The stable w interval is

$$0.59 < w < 6.05$$

With $J_2 = 150 \cdot 10^{-6}$ kgm^2 and $k = 2.4 \cdot 10^{-3}$ Nm/rad, the upper natural frequency limit corresponding to $w_{max} = 6.05$ can be calculated from (6.4):

$$\omega_{max}^2 = \frac{k w_{max}}{J_2} \quad \Rightarrow \quad \omega_{max} = 9.84 \text{ rad/s}$$

This is a good approximation of the true natural frequency limit. Numerical calculation of the regulator poles for different ω values using the complete servo model show that the stability limit is ω_{max} = 9.90 rad/s. It also agrees well with the results in Chapter 4 and 5, where it was found that a design with ω=12 gives an unstable regulator and limit cycle oscillations appear, whereas ω=8 gives a stable regulator and no limit cycles (cf. Figure 4.2 and 4.4).

If the performance specifications should require a certain minimum closed loop natural frequency ω, the corresponding maximum allowable inertial load J_2 (or, alternatively, the minimum spring constant k) can be calculated from the w stability limit and (6.4). The influence of the control design on the achievable bandwidth can be investigated by plotting the normalized coefficients (6.9) for different ζ and α values. By increasing the relative damping ζ of the closed loop system, the achievable bandwidth with respect to regulator stability is increased. A fast observer (increased α), on the other hand, has the opposite effect.

Note that the sign of a_2 and a_3 (cf. eqn 6.5) depends not only on w, but also on the inertia ratio J_2/J_1. With reasonable inertia ratios this dependance is small, however, since J_2/J_1 only appears in the constant (w-independent) term. A numerical example will illustrate this point:

EXAMPLE 6.1

Let $\hat{a}_2(w)$ denote approximate a_2 values obtained from (6.5a) with a constant inertia ratio J_2/J_1 and varying w. With nominal parameters (ζ=0.7, α=1.5, $J_1=22\cdot10^{-6}$ kgm^2 and $J_2=150\cdot10^{-6}$ kgm^2) the equation

$$\hat{a}_2(w) = 0$$

has one real solution, w = 6.62 (cf. Figure 6.1). Using (6.4) with ω = 12 and k = $2.4\cdot10^{-3}$ Nm/rad, this solution corresponds to $J_2 = 110.4\cdot10^{-6}$ kgm^2.

An exact evaluation of a_2 shows that

$$J_2 = 110\cdot10^{-6} \Rightarrow a_2 = 6.05$$
$$J_2 = 111\cdot10^{-6} \Rightarrow a_2 = -28.45$$

Hence the solution to $\hat{a}_2(w) = 0$ with the assumption that J_2/J_1 is constant can be used with good accuracy to predict the J_2 value for which a_2 changes sign. □

Conclusions

It has been shown that the stability of the regulator depends on the dimensionless quantity w. For large w values, the regulator will be unstable and limit cycles will appear. By the definition of w, we see that regulator instability can be caused by a too large closed loop natural frequency (ω) or inertial load (J_2), or a too weak coupling between the drive motor and the load (small k). Since w is a quadratic function of ω, we may conclude that decreasing ω is a more efficient way to stabilize the regulator than modification of J_2 or k. Furthermore, ω is a control design parameter and as such it may be the only parameter at our disposal to manipulate w. The requirement that the regulator should be stable thus imposes constraints on the achievable bandwidth.

Although an unstable regulator is a sufficient condition for the appearance of limit cycles, we have no proof that by stabilizing the regulator we will eliminate the limit cycles. In all cases that have been tested so far, however, limit cycle appearance has coincided with an unstable regulator. It is therefore conjectured that regulator instability is also a necessary condition for limit cycles to appear.

6.2 NON-COLOCATED SPEED SENSOR

An interesting observation is that the limit cycles can be eliminated by using a non-colocated speed sensor, i.e. measuring the speed of the inertial load instead of the drive motor speed. This is demonstrated in Figure 6.2, which shows the closed loop step response with nominal control design (ω=12, ζ=0.7, α=1.5), and with $y_2 = k_{\omega 2}\omega_2$ as input to the regulator. As we have seen in Chapter 4, this control design will result in limit cycle oscillations if ω_1 is measured (cf. Figure 4.2).

Describing function analysis

The absence of limit cycles when ω_2 is measured can be explained with the describing function method. Decompose the system into one linear and one nonlinear subsystem in a fashion similar to Figure 5.2. The servo model can in this case be written as

$$\begin{cases} \dot{x} = Ax + B(M_f+u) \\ y_1 = C_1x \\ y_2 = C_2x \end{cases} \qquad (6.10)$$

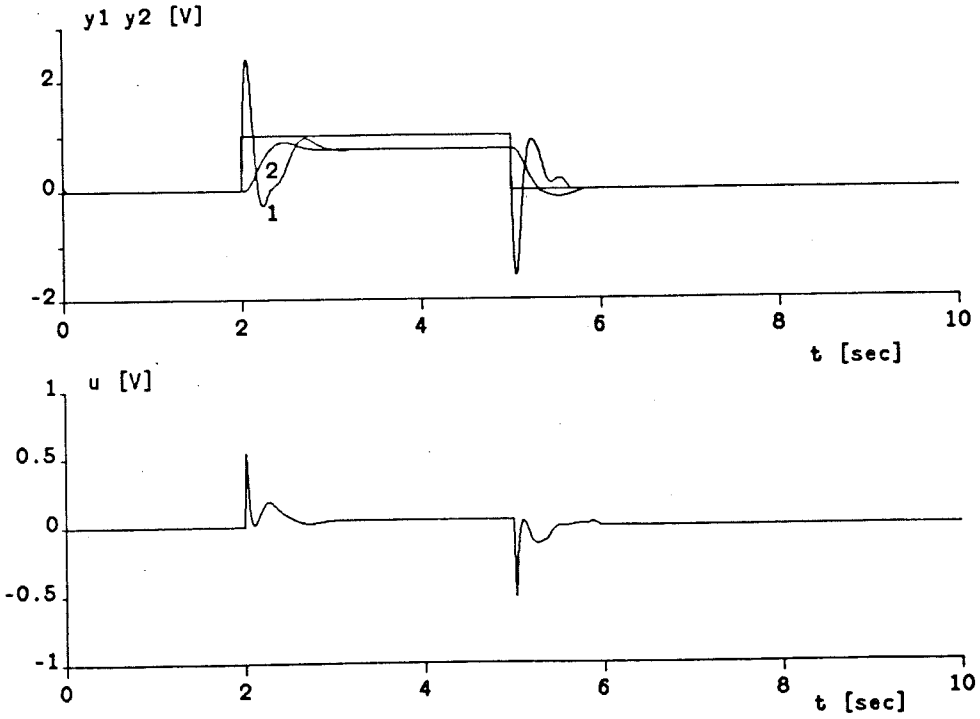


Fig. 6.2 Closed loop step response with nonlinear model. Non-colocated speed sensor (y_2 measured) and nominal control design (ω=12, ζ=0.7, α=1.5).

where A is defined as in (3.1) and

$$B = \left[\frac{1}{J_1}\ 0\ 0\right]^T$$

$$C_1 = \left[k_{\omega 1}\ 0\ 0\right]\ ; \quad C_2 = \left[0\ k_{\omega 2}\ 0\right]$$

With zero speed reference, the regulator equations are

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K(y_2 - C_2\hat{x}) \\ u = -L\hat{x} \end{cases} \tag{6.11}$$

The linear subsystem will be described by the state space equation

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BL \\ KC_2 & A-BL-KC_2 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} M_f$$

$$y = \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \tag{6.12}$$

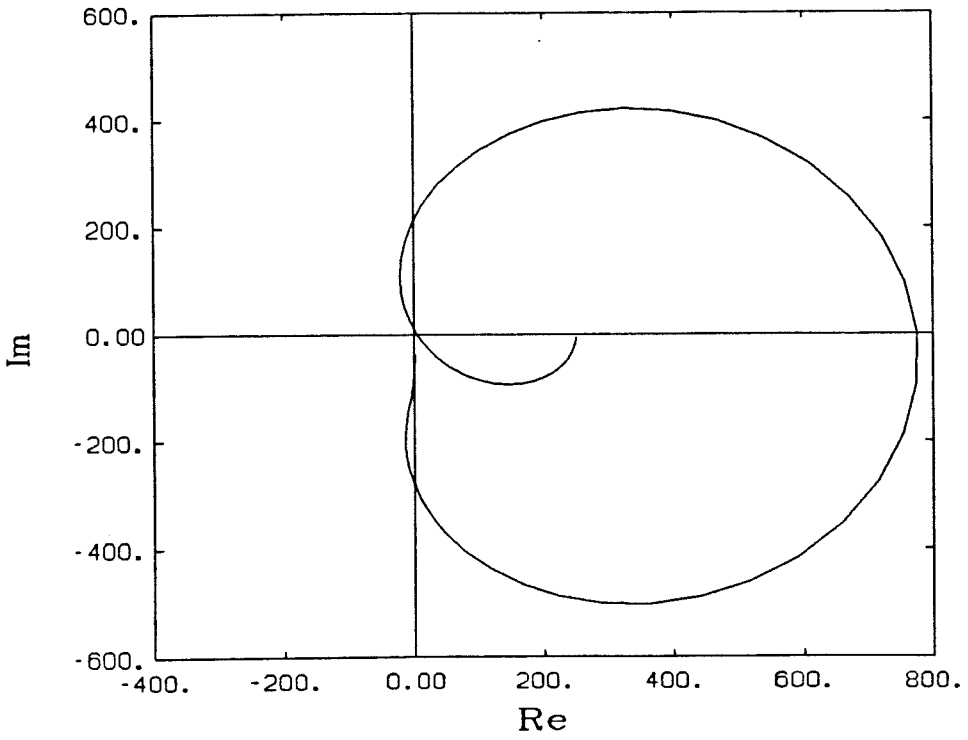


Fig. 6.3 Nyquist curve for linear subsystem (y_2 measured). Nominal control design (ω=12, ζ=0.7, α=1.5).

The output of the linear subsystem is $y_1 = k_{\omega 1}\omega_1$, since the nonlinearity corresponds to the Coulomb friction on shaft 1. The Nyquist curve for the linear subsystem never intersects the negative real axis (see Figure 6.3). Therefore, by the describing function stability criterion described in Chapter 5, we have a stable system and no limit cycles.

Regulator stability

Just as in the colocated speed sensor case (cf. Section 5.2), it can be shown that an unstable regulator will give rise to limit cycles. The linear subsystem can now be described with a block diagram according to Figure 6.4. (Cf. Figure 5.2 and 5.5). With

$$G_1(s) = C_1(sI-A)^{-1}B = \frac{B_1(s)}{A_o(s)}$$

$$G_2(s) = C_2(sI-A)^{-1}B = \frac{B_2(s)}{A_o(s)}$$

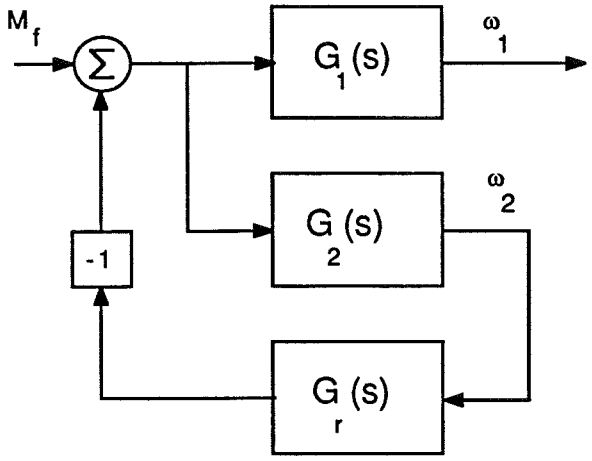


Fig. 6.4 Block diagram for linear subsystem with non-colocated speed sensor.

$$G_r(s) = L(sI-A+Bl+KC_2)^{-1}K = \frac{B_r(s)}{A_r(s)}$$

the transfer function for the closed loop linear subsystem is

$$G = \frac{B_1 A_r}{A_o A_r + B_r B_2}$$

Note that also in this case the regulator poles appear as zeros in the closed loop transfer function. By applying the same type of arguments that we used in Lemma 5.1, it is easy to prove that an unstable regulator will cause the closed loop Nyquist curve $G(i\omega)$ to intersect the negative real axis. Hence, by the describing function stability criterion, limit cycles will appear whenever the regulator goes unstable.

Stability criteria

It is interesting to note that in this case the regulator does not go unstable when the closed loop bandwith is increased. The regulator has the characteristic equation

$$A_r(s) = \det(sI-A+BL+KC_2) = 0$$

Using the same notation as in Section 6.1,

$$A_r(s) = s^3+a_1s^2+a_2s+a_3$$

the regulator stability can be investigated with the criterion (6.6). With the dimensionless quantity (cf. 6.4)

$$w = \frac{J_2\omega^2}{k}$$

we get

$$a_1 = \omega(2\zeta+1)(\alpha+1)$$

$$a_2 = \frac{k}{J_2}\cdot\left\{w\left[4\alpha\zeta^2+(2\alpha^2+4\alpha+2)\zeta+(\alpha^2+\alpha+1)\right] - \left[1+\frac{J_2}{J_1}\right]\right\}$$

$$a_3 = \frac{\omega k}{J_2}\left\{w\left[4\alpha(\alpha+1)\zeta(\zeta+1)+\alpha^3+\alpha^2+\alpha+1\right] - (\alpha+1)(2\zeta+1)\left[1+\frac{J_2}{J_1}\right]\right\}$$

and

$$e = a_1a_2 - a_3 =$$

$$= \omega^3\left[8\alpha(\alpha+1)\zeta^3+4(\alpha^3+3\alpha^2+3\alpha+1)\zeta^2+2(2\alpha^3+3\alpha^2+3\alpha+2)\zeta+\alpha(\alpha+1)\right]$$

Both a_1 and e are strictly positive, so the regulator stability will depend only on the signs of a_2 and a_3. These coefficients are linear functions in w, and they will be negative for small w values. Figure 6.5 shows the normalized coefficients

$$a_{2n}(w) = \frac{J_2}{k}\cdot a_2(w) \;;\quad a_{3n}(w) = \frac{J_2}{\omega k}\cdot a_3(w)$$

calculated with nominal design parameters (ζ=0.7, α=1.5, $J_1=22\cdot10^{-6}$ kgm^2 and $J_2=150\cdot10^{-6}$ kgm^2). The stable w interval is

$$w > w_{min} \approx 1.9$$

which corresponds to (cf. eqn 6.4) $\omega_{cl} > 5.5$ rad/s.

We now have the interesting result that with a non-colocated speed sensor there exists a lower bandwidth limit below which the regulator is unstable. There is no upper limit, however. This agrees with the nominal design, ω_{cl} = 12 rad/s, which produces a stable regulator. The corresponding Nyquist curve (cf. Figure 6.3) does not intersect the negative real axis, and no limit cycles appear in this case.

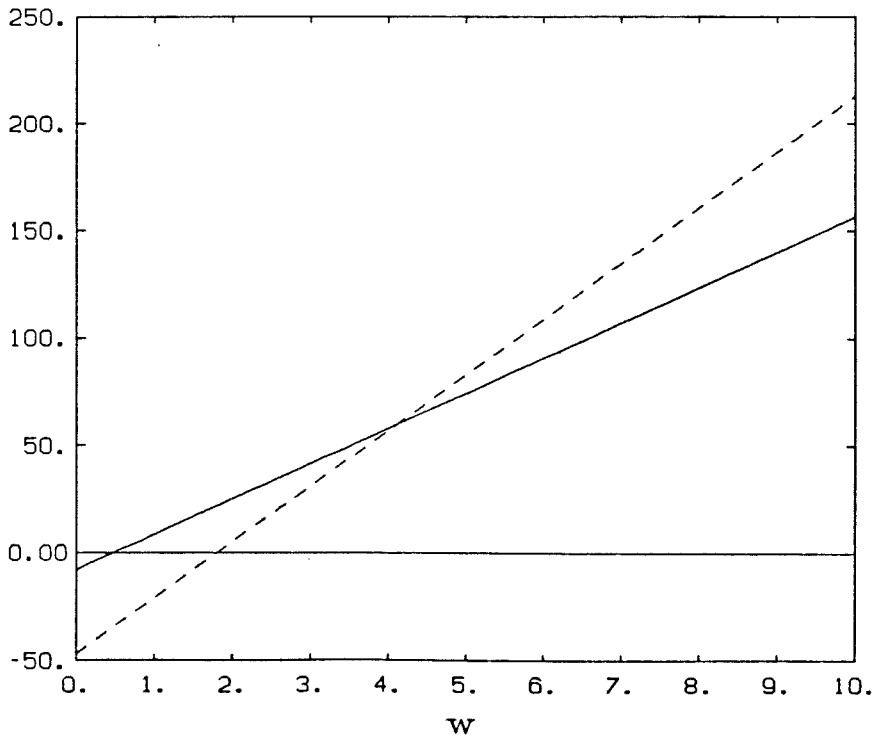


Fig. 6.5 Normalized coefficients of the regulator characteristic polynomial with non-colocated speed sensor and nominal design. Solid line = $a_{2n}(w)$, dashed line = $a_{3n}(w)$.

Conclusions

No limit cycles appear in the nominal design case with a non-colocated speed sensor. This can be explained with the describing function method. We have shown that an unstable regulator is a sufficient condition for the existence of limit cycles. With a non-colocated sensor, the regulator goes unstable for low bandwidths instead of high. No limit cycles have been observed with stable regulators. Our conjecture that regulator instablility is a sufficient and necessary condition for the appearance of limit cycles is supported by these observations.

The conclusion is that a non-colocated speed sensor may be one way to achieve higher bandwidths without the appearance of limit cycles. Consequently this sensor arrangement should be preferred in cases where fast servo response is required. One could of course also consider using two speed sensors, one for each speed signal. Unfortunately, in many practical applications this is not feasible since it is often difficult to measure the inertial load speed.

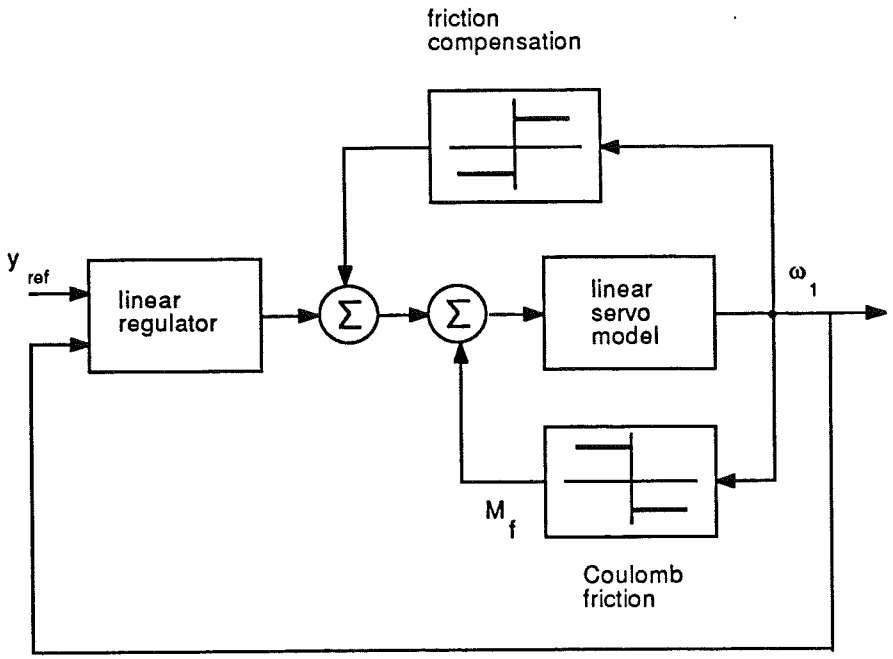


Fig. 6.6 Block diagram of servo system with friction compensation.

6.3 FRICTION COMPENSATION

One attractive way to eliminate the limit cycle problem is to "linearize" the system by introducing an additional nonlinearity into the system, thereby cancelling the influence of the existing nonlinearity. This approach has been reported succesful in several similar applications, e.g. [1], [4], [12]. It is done by adding an extra term, corresponding to the estimated Coulomb friction torque, to the control signal. See Figure 6.6. The problem is that the exact nonlinear function describing the Coulomb friction is unknown, and if it were known it would probably be difficult to implement. A simple ideal relay friction model has been used succesfully to predict and analyse the limit cycle phenomenon. If the friction compensation is implemented as a relay function, the discontinuity at zero speed may cause problems such as chattering, however. Hopefully, a modified relay function in the friction compensation will be sufficient to eliminate the limit cycles.

Two alternative approximations of an ideal relay function have been tested, a high gain with saturation and a relay with deadzone. The high gain model is

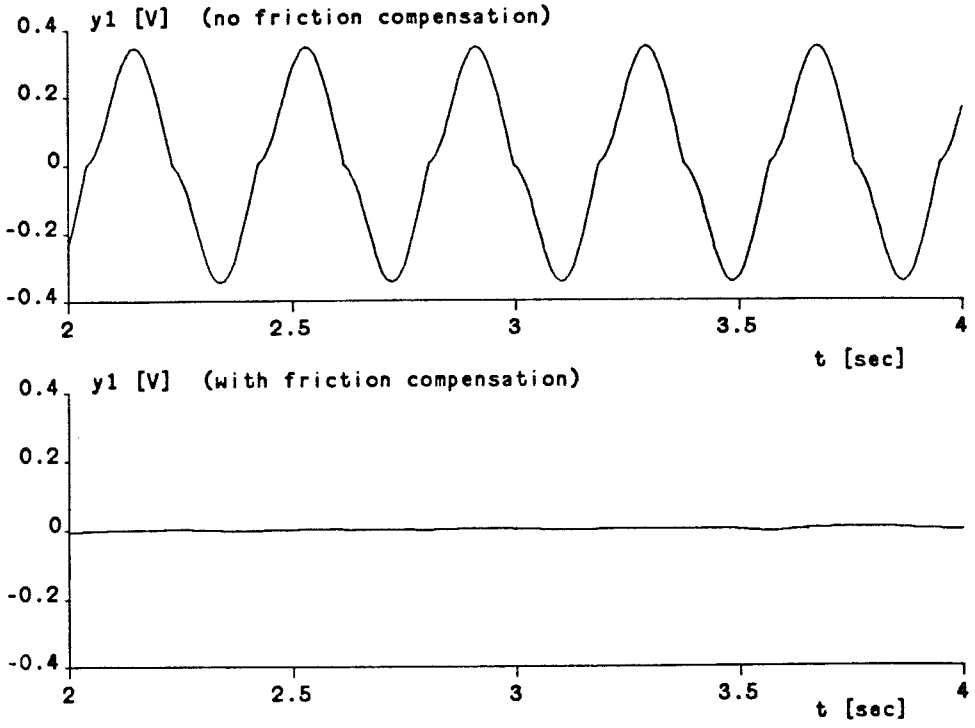


Fig. 6.7 Friction compensation (high gain with saturation). Nominal control design (ω_{cl}=12, ζ=0.7, α=1.5).

$$M_f = \begin{cases} F \cdot \omega_1/\varepsilon & ; \ |\omega_1| < \varepsilon \\ F \cdot \text{sign}(\omega_1) & ; \ |\omega_1| \geq \varepsilon \end{cases} \tag{6.13}$$

where F is the estimated friction torque on shaft 1, and ε is a small number defining the linear range of the friction model. The relay with deadzone is modelled as

$$M_f = \begin{cases} 0 & ; \ |\omega_1| < \varepsilon \\ F \cdot \text{sign}(\omega_1) & ; \ |\omega_1| \geq \varepsilon \end{cases} \tag{6.14}$$

In this case ε is the width of the deadzone. The measured speed signal, $y_1 = k_{\omega 1}\omega_1$, may be distorted by measurement noise and ripple from the tacho signal. Therefore the estimated speed signal from the observer in the linear regulator is used as input to the friction compensation.

Figures 6.7 and 6.8 illustrate the effect of friction compensation using the models (6.13) and (6.14), respectively. In both cases the estimated friction torque is $F = 5 \cdot 10^{-4}$ Nm (the same value as in the nonlinear servo model, i.e. the "true" value), and the parameter ε is 0.001. As can be seen in the figures, the limit

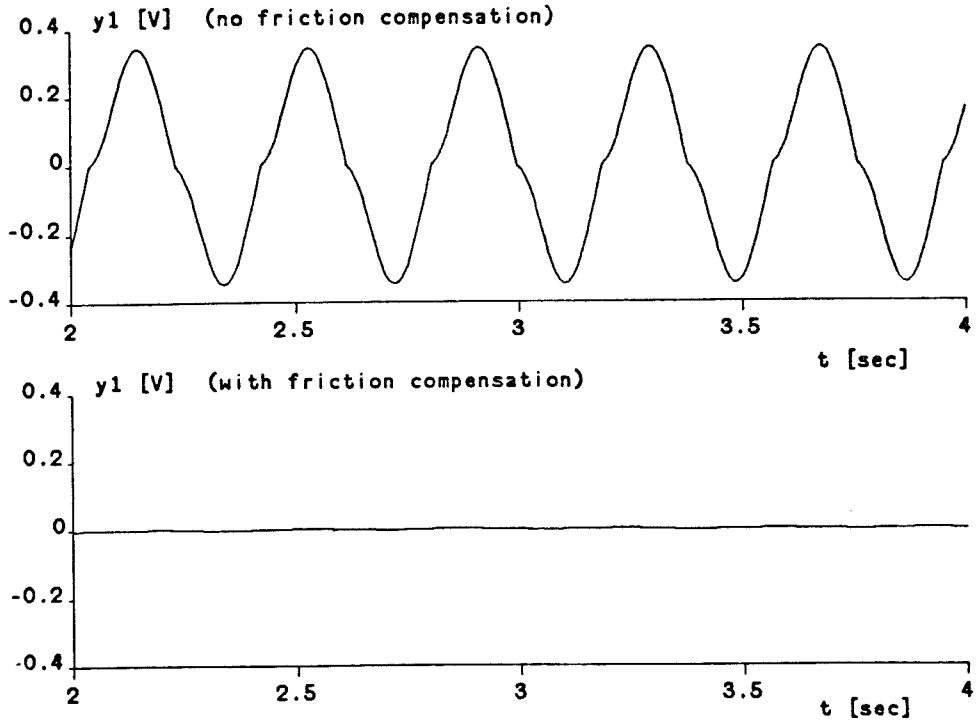


Fig. 6.8 Friction compensation (relay with deadzone). Nominal control design (ω_{cl}=12, ζ=0.7, α=1.5).

cycles in the output virtually disappear regardless of which method we use to approximate the ideal relay function. If ϵ is chosen too large or F has an incorrect value, the performance of the friction compensation will be degraded and the limit cycles will reappear, typically with a reduced amplitude.

Digital friction compensation

The discrete time linear control algorithm (4.3) can be augmented with a discrete time friction compensation term calculated from either (6.13) or (6.14). It is a nontrivial task to simulate the digital friction compensation, however. The default integration routine in SIMNON is Hamming's modified predictor corrector method (HAMPC) [6]. This method was used first, but it was found that the results depended critically on maximum step size and the error bound. When the classical fourth order Runge Kutta method with variable step size (RK) was used, the maximum step size and error bound did not have any significant influence on the results. The RK algorithm was therefore used in this particular case. Figure 6.9 shows a the effect of a nonlinear friction compensation based on (6.13) in the discrete time case. The limit cycle amplitude is reduced to an almost negligible level.

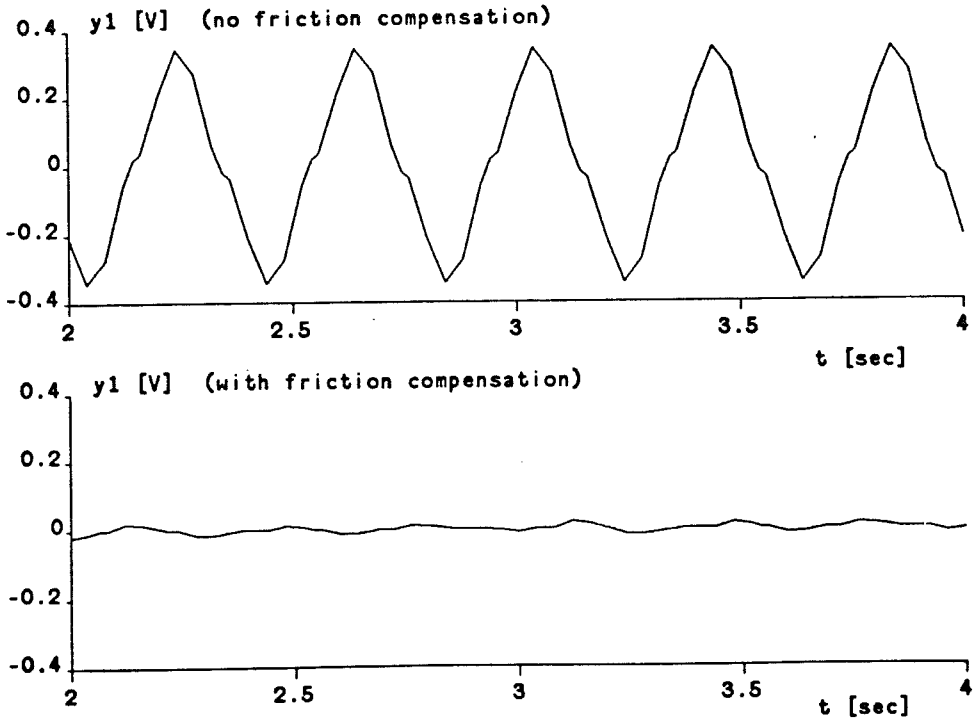


Fig. 6.9 Digital control and friction compensation (high gain with saturation). Nominal control design corresponding to ω_{cl}=12, ζ=0.7, α=1.5, and a sampling interval h=40 ms. Integration algorithm: RK with default maximum step size and error bound (ERROR=0.001).

7. EXPERIMENTAL RESULTS

7.1 IMPLEMENTATION OF THE LINEAR CONTROL LAW

Originally it was intended to implement the digital control law described in Section 4.2. After some experimentation, it was found that the required sampling rate was difficult to achieve with the available equipment. Furthermore, the theoretical analysis of the limit cycle phenomenon is based on a continuous time model and regulator (cf. Chapter 5 and 6). It was therefore decided to verify the theoretical results and simulations by implementing the continuous time control law, described by (4.1) and (4.2), with analog hardware.

Amplitude scaling

Before designing the circuits, the linear state space control law must be transformed into a form suited for analog implementation. With nominal parameter values and control design (cf. Chapter 3 and 4) we have the following state feedback and observer equations:

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \\ u = l_r y_r - L\hat{x} \end{cases} \tag{7.1}$$

where

$$A = \begin{bmatrix} -0.45 & 0 & 109 \\ 0 & -0.07 & -16.0 \\ -1 & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1136 \\ 0 \\ 0 \end{bmatrix} \qquad K = \begin{bmatrix} 427 \\ 467 \\ 59.6 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.1 & 0 & 0 \end{bmatrix}$$

$$L = 10^{-2}\begin{bmatrix} 2.49 & 6.86 & -19.2 \end{bmatrix} \qquad l_r = 0.95$$

The A matrix is badly conditioned in itself, and the combination of very large gains (in the B and K matrices) with very small gains (in the L matrix) is also bad from a numerical point of view. In an analog implementation, it is desirable to have all gains within the range 0.1 - 100 if possible. We must also make sure that all signals (e.g. state variables, inputs, outputs etc.) stay within the normal

operating range of the operational amplifiers. With a ±15 V power supply, restricting the signal levels to the interval ±10 V provides an adequate safety margin (20-30%) to saturation.

This can be achieved with amplitude scaling of the state variables. During normal operation, the states are assumed have the following maximum values:

$$|x_1| < 100 \text{ rad/s}$$
$$|x_2| < 100 \text{ rad/s}$$
$$|x_3| < 5 \text{ rad}$$

Amplitude scaling of the state vector x can be expressed as a coordinate transformation z = Tx where T is a diagonal matrix and z is the new (scaled) state vector. Let the maximum x_i value correspond to z_i = 10 V. This implies that

$$T = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 2.0 \end{bmatrix}$$

With z representing the estimated states in the observer, the transformed realisation of the control law is

$$\begin{cases} \dot{z} = Az + Bu + K(y - Cz) \\ u = l_r y_r - Lz \end{cases} \qquad (7.2)$$

where

$$A = \begin{bmatrix} -0.45 & 0 & 5.45 \\ 0 & -0.07 & -0.80 \\ -20 & 20 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 113.6 \\ 0 \\ 0 \end{bmatrix} \quad K = \begin{bmatrix} 42.7 \\ 46.7 \\ 119 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.0 & 0 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.249 & 0.685 & -0.0962 \end{bmatrix} ; \qquad l_r = 0.95$$

More sophisticated methods are available for transformation of state space models into well-conditioned, or balanced, realisations [5]. These methods generally have the disadvantage that they destroy "exact zeros" in the matrices. In our case this means a loss of structure in the A matrix which implies that more components are required in the implementation. Furthermore, balancing methods do not guarantee that the state values stay within the required range. Amplitude scaling

was therefore considered to be a more appropriate method for our problem.

Controller stability and windup protection

In state space control design, the feedback gain L and observer gain K are chosen such that the closed loop system and the observer error have specified dynamics. We have no guarantee at all, however, that the resulting controller will be stable. This is not just a state space design problem, the same problem arises when using the input-output approach. If we eliminate u from the observer equation, and collect all terms on the right hand side containing the state vector z, the controller equations can be written as:

$$\begin{cases} \dot{z} = (A-BL-KC)z + Bl_r y_r + Ky \\ u = l_r y_r - Lz \end{cases}$$

The controller is unstable if the matrix (A-BL-KC) has eigenvalues in the right half of the plane. As we have seen in Section 5.2, this is in fact the case with our nominal design. An unstable controller implies that extra precautions must be taken in the implementation to avoid windup problems during startup, or if the loop is at any time broken. One way to do this is to distinguish between the process input u_{in} and the controller output u_{out} and write the controller equations as

$$\begin{cases} \dot{z} = (A-KC)z + Bu_{in} + Ky \\ u_{out} = l_r y_r - Lz \end{cases}$$

This is a standard method to avoid windup in the controller if the process input saturates or otherwise differs from the controller output signal [18]. If we break the loop, i.e. $u_{in} \neq u_{out}$, the observer part of the controller is a stable system and the state estimates will follow the true states as long as the observer is fed with the same input signal as the process. Thus we will get a (comparatively) smooth transient when the loop is closed since the calculated control signal is based on correct state estimates. The start-up or "connect" procedure for the analog controller should be:

1) Power on (with process disconnected from controller).
2) Connect process input u to u_{in} of controller.
3) Connect controller output u_{out} to process input u.

Shut down or "disconnect" is done by reversing this procedure.

Implementation

The implementation structure for the observer and state feedback control law was chosen as

$$\begin{cases} \dot{z} = Az + Bu_{in} + K(y-Cz) \\ u_{out} = l_r y_r - Lz \end{cases} \tag{7.3}$$

This structure does not minimize the required number of components, but it has several other nice properties which make it suitable for a prototype controller. It can be implemented with a one-to-one correspondence between each matrix element and a single component (resistor or capacitor) value, which means that independent changes of model parameters (A, B, C) and controller gains (L, l_r, K) can be made easily. It also performs an explicit calculation of the estimated process output $\hat{y} = Cz$ and the estimated output error $\tilde{y} = y-\hat{y}$. The output estimate $\hat{y}$ is a filtered value of the measured output y (which is distorted by tachometer ripple), and can be used in the friction compensation circuit. The output estimate error $\tilde{y}$ may be of interest for diagnostic purposes.

The standard analog integrator circuit includes a change of sign. The integrator outputs will thus correspond to the state estimates z if the integrator inputs are chosen as

$$-\dot{z} = -Az-Bu_{in}-K(y-Cz)$$

With the observer states as integrator outputs, and positive state feedback gains (cf. eqn 7.1), the control signal can be computed in one single inverting amplifier stage. Only one sign shift is needed for the reference signal. Using the structure above and the fact that several matrix elements are zero, the controller equations will be

$$\begin{cases} -\dot{z}_1 = -a_{11}z_1 - a_{13}z_3 - b_1u_{in} - k_1(y-c_1z_1) \\ -\dot{z}_2 = -a_{22}z_2 - a_{23}z_3 - k_2(y-c_1z_1) \\ -\dot{z}_3 = -a_{31}z_1 - a_{32}z_2 - k_3(y-c_1z_1) \\ u_{out} = l_r y_r - l_1z_1 - l_2z_2 - l_3z_3 \end{cases} \tag{7.4}$$

Conventional analog circuits based on operational amplifiers were used in the implementation. The circuit design required 11 operational amplifiers, 30 resistors and 3 capacitors. A block diagram and a complete circuit diagram for the analog linear controller is given in appendix D.

Non-colocated speed sensor

Only minor modifications of the prototype controller are required to implement the linear analog controller with a non-colocated speed sensor. These modifications involve the observer circuits only. The computation of the estimated process output $\hat{y} = Cz$ must of course be modified, and the observer gain vector K is different. An extra inverter must also be included, since one of the observer gains (k_3) is negative in the nominal design case. The computation of the control signal

$$u_{out} = l_r y_r - Lz$$

is independent of the selected sensor location, provided that the state estimates are correct. A complete circuit diagram can be found in appendix D.

7.2 EXPERIMENTAL RESULTS WITH THE LINEAR CONTROL LAW

The behaviour of the closed loop system was tested with a step reference signal (step amplitude = 1 Volt = 10 rad/s). The drive motor speed y_1, the inertial load speed y_2 and the control signal u were logged with a special purpose data acquisition program with a sampling interval of 10 ms. A high frequency disturbance can be seen on the y_1 signal. It is caused by tachometer ripple, and the ripple frequency is proportional to the drive motor speed y_1. The normal steady state speed, y_1 = 1 V, corresponds to a ripple frequency of 100 rad/s. The maximum speed value in the experiments, y_1 = 3 V, gives a maximum ripple frequency of 300 rad/s or approximately 50 Hz. This maximum frequency appears only in transients.

Nominal design

Figure 7.1 shows the closed loop step response with nominal linear control design (ω_{cl} = 12 rad/s). The transient behaviour agrees very well with the simulation of the nonlinear model (cf. Figure 4.2), and a a limit cycle oscillation can be observed when the speed reference is zero. The limit cycle frequency coincides with the predicted value 2.5 Hz. The amplitude of the oscillation is approximately

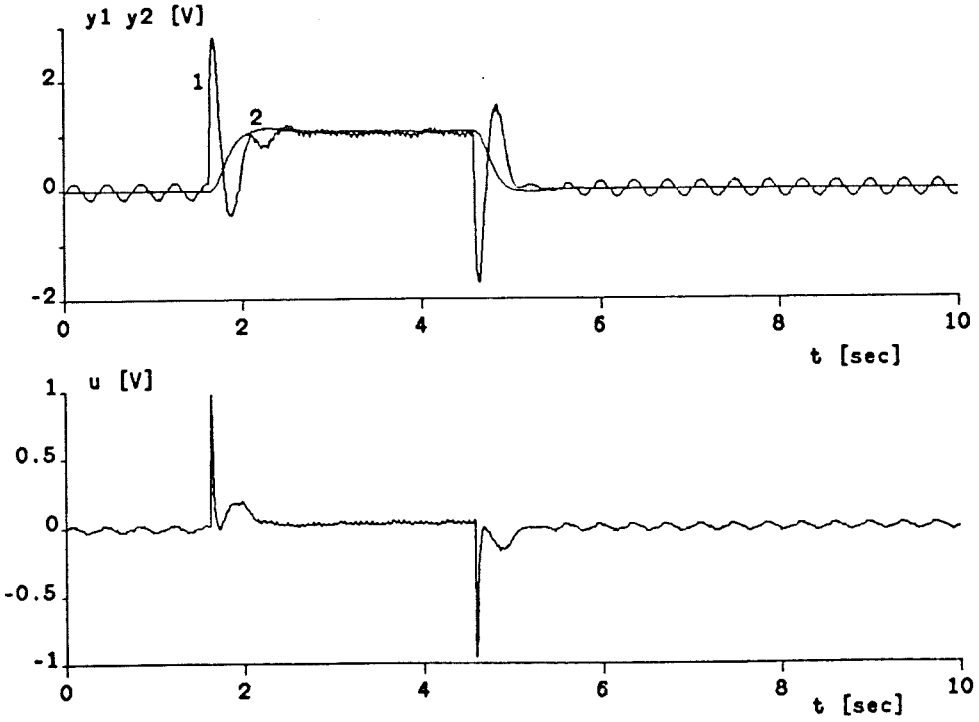


Fig. 7.1 Analog linear controller: Closed loop step response with y_1 measured and nominal design (ω_{cl}=12 rad/s).

0.2 V. This is smaller than the predicted value 0.3 V (cf. Section 5.1). The amplitude discrepancy between the simulations and the experiments can be explained by an incorrect value of the Coulomb friction torque. It follows from equation (5.2) that an incorrect value of the friction torque gives an error in the predicted amplitude, but not in the frequency.

Reduced bandwidth

The closed loop step response with reduced bandwidth design (ω_{cl} = 8 rad/s) is shown in Figure 7.2. The limit cycle oscillations have now disappeared just as predicted by the analysis in Chapter 5 and 6. The transient behaviour agrees well with the simulations (cf. Figure 4.4). Note that the steady state error is much smaller in the experiment than in the simulated step response. This can be explained by errors in the estimated Coulomb friction torques. In steady state with non-zero speed, the Coulomb friction may be regarded as a load disturbance on the control signal. With a low bandwidth design the controller gains are small, and a load disturbance on the control signal will have a large influence on the output steady state error. An incorrect Coulomb friction torque will therefore have a significant influence on the steady state error in the simulated step

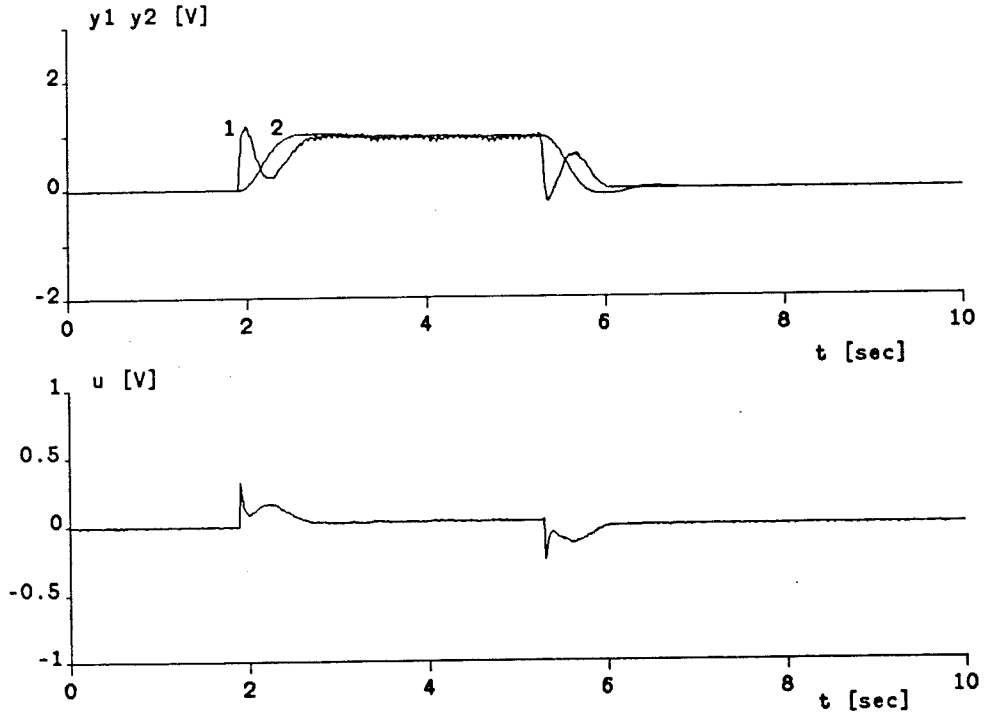


Fig. 7.2 Analog linear controller: Closed loop step response with y_1 measured and reduced bandwidth (ω_{cl}=8 rad/s).

response.

As mentioned in Section 4.1, steady state errors can of course be eliminated by introducing integral action in the regulator. This does not change the basic issues that we want to study. The regulator then becomes more complex, however, and the algebraic calculations will be much more involved.

Non-colocated speed sensor

Figure 7.3 shows the closed loop step response with nominal design (ω_{cl} = 12 rad/s) and a non-colocated speed sensor, i.e. direct measurement of the inertial load speed. As predicted in Chapter 6 no limit cycles appear in this case. The dynamic response agrees well with the simulations (cf. Figure 6.2). Note that the variation in the drive motor speed y_1 due to disturbances has increased compared with the colocated speed sensor case. This different behaviour can be explained as follows. When y_1 is measured, the controller senses the disturbances and tries to counteract them to keep y_1 at a constant value. With a non-colocated speed sensor, these disturbances are filtered by the elastic coupling between the drive motor and the inertial load. Therefore the disturbances are not noticed by the controller and consequently not compensated. This is not

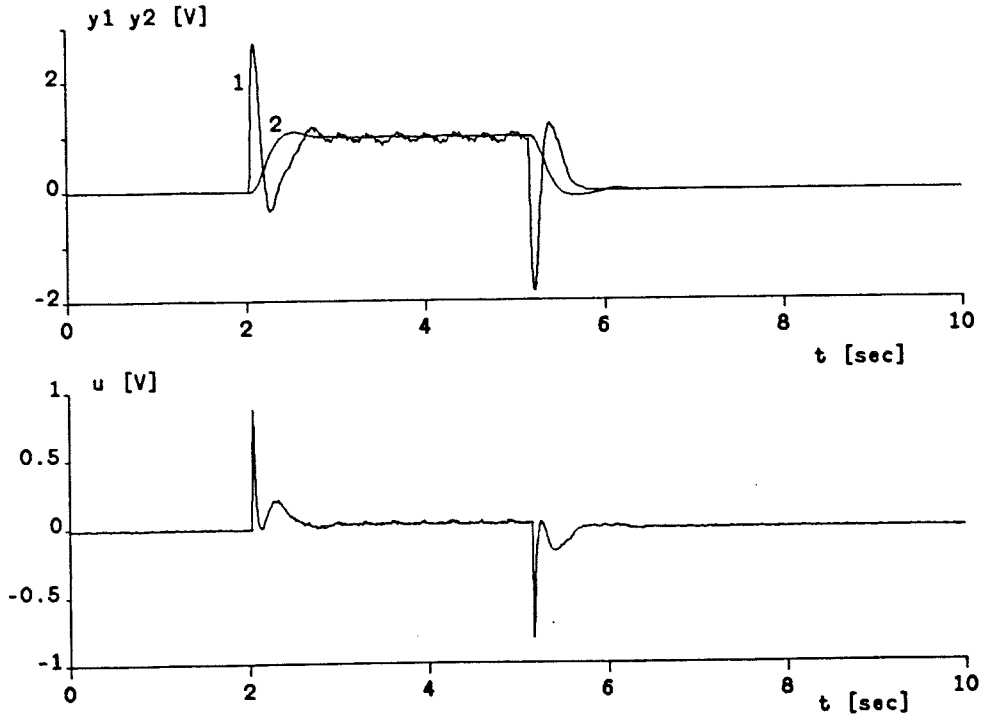


Fig. 7.3 Analog linear controller: Closed loop step response with non-colocated speed sensor. Nominal design (ω_{cl}=12 rad/s).

important, however, since our main objective is to control the inertial load speed y_2. Small variations in the drive motor speed can be accepted as long as they do not propagate to the inertial load.

Conclusions

The experimental results with linear state feedback and reconstruction agree well with the simulations and theoretical calculations. Explanations have been given in those cases where differences have been noticed. We may therefore conclude that the flexible servo model presented in Chapter 3 provides a description of the system dynamics sufficiently accurate to be used for analysis and design of linear control laws.

7.3 ANALOG FRICTION COMPENSATION

The simulations in Chapter 6 indicate that the limit cycle oscillations can be practically eliminated by nonlinear friction compensation, i.e. by adding a term corresponding to the Coulomb friction torque to the control signal from the linear controller (cf. Section 6.3). Two approximations of the ideal relay friction model were tested in the simulations, a relay with a small deadzone and a high gain with

saturation. The performance of the two approximations was found to be quite similar. An analog friction compensation circuit has been built to test these results on the prototype flexible servo system.

Implementation

A relay with a small deadzone is easier to implement with analog circuits than a high gain amplifier with a sharp saturation. The friction compensation circuit used in the experiments is described with the following equation (cf. equation 6.14):

$$u_f = \begin{cases} F_+ \; ; & \hat{y}_1 > \varepsilon \\ 0 \; ; & |\hat{y}_1| < \varepsilon \\ F_- \; ; & \hat{y}_1 < -\varepsilon \end{cases} \qquad (7.4)$$

In this equation u_f is the output voltage from the friction compensation, $\hat{y}_1$ is the estimated angular velocity of the drive motor (calculated in the observer of the linear controller) and ε is the relay deadzone. F_+ and F_- are voltage levels corresponding to the Coulomb friction torque in the positive and negative direction of rotation, respectively. These two levels where made individually adjustable to enable compensation of unsymmetric Coulomb friction. The deadzone ε was also made adjustable. A circuit diagram of the analog friction compensation is given in appendix D.

Experimental results

The deadzone ε was adjusted to a minimum while still maintaining reliable cicrcuit performance, and the parameters F_+ and F_- were trimmed to achieve minimum limit cycle amplitude. These experiments showed that the Coulomb friction was unsymmetric. The best performance was achieved with the following parameter values:

$F_+ = 17$ mV
$F_- = 9$ mV
$\varepsilon = 20$ mV.

The friction torque M_f corresponding to u_f can be calculated from (cf. Chapter 3)

$$M_f = k_m \cdot k_i \cdot u_f$$

With the parameters given above this yields $M_{f+} = 4.3 \cdot 10^{-4}$ Nm and

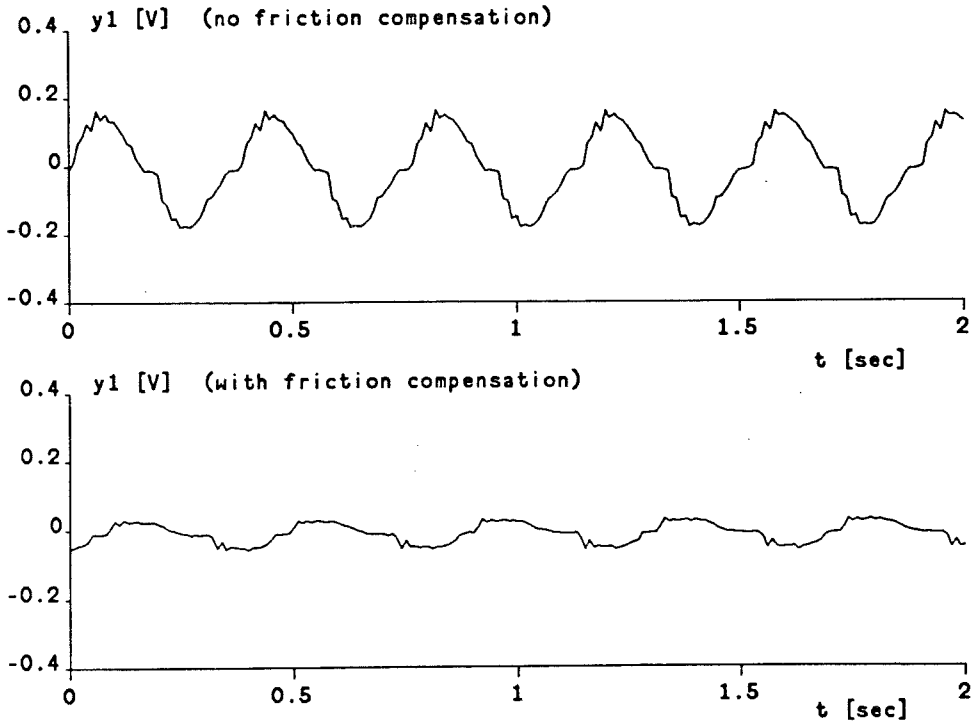


Fig. 7.4 Limit cycles with analog linear controller (nominal design, ω_{cl}=12 rad/s), and linear controller with friction compensation (relay with deadzone). F_+=17 mV, F_-=9 mV, $\epsilon \approx 20$ mV.

$M_{f-} = 2.3 \cdot 10^{-4}$ Nm. A comparison with the earlier estimated values in Section 3.2 shows that the values given above are significantly smaller (particularly M_{f-}). This agrees with the fact that the amplitude of the observed limit cycles is smaller than the predicted value, since the amplitude predicted by the describing function method is proportional to the Coulomb friction torque (cf. Chapter 5). We may thus conclude that either the friction torque has changed or the previous estimates were too high.

Figure 7.4 illustrates the effect of the analog friction compensation on the limit cycle oscillations (note that the figure shows two different experiments, hence the two curves are not synchronized in time). The limit cycle amplitude is reduced approximately by a factor of four, but the oscillations do not disappear. It may be argued that it would be better to feed the friction compensation circuit with the measured y_1 signal instead of the reconstructed value $\hat{y}_1$. This has also been tested, however, and it was found to give a deteriorated performance. This may be due to the ripple on the y_1 tacho signal, which causes the friction compensation circuit to miss the zero crossings and thus change the compensation torque at incorrect times.

A relay without deadzone has also been tested, implemented as an analog zero crossing detector. The results were similar to those in Figure 7.4, with only a slightly different limit cycle curve shape. We may thus conclude that the deadzone size used in the experiments (ϵ = 20 mV) is not too large, and that this is not the reason for the bad performance of the friction compensation.

Conclusions

The friction compensation schemes that have been used in these experiments seems to be too simple to enable elimination of the limit cycle oscillations. A significant reduction of the oscillation amplitude can be achieved, however. It appears that it is not the selected approximation of the relay function that is the critical part, but that the relay Coulomb friction model is too simple.

The general impression from the experiments is that friction compensation is very sensitive to parameter variations. This is not surprising, since it is well known that static friction torques vary due to temperature changes, mechanical wear etc. Friction compensation is also sensitive to disturbances, since the friction torque is a discontinuous function of the speed. It is important to be able to detect the zero crossings of the measured speed with great accuracy, and minor disturbances at or near the zero crossings may have a large negative effect on the performance.

The conclusion that can be drawn from these experiments is that nonlinear friction effects in servo systems with resonant modes cannot be compensated with the simple friction compensation schemes presented here. In order to eliminate the limit cycle oscillations induced by Coulomb friction, more sophisticated friction models and compensation schemes will have to be employed. This, however, brings us beyond the scope of this thesis.

8. CONCLUSIONS

Active damping of resonant modes

It has been demonstrated that a control design based on a linear model and pole placement with state feedback is an adequate method to provide active damping of the resonant modes in a flexible servo system. Well damped closed loop poles gives a well-damped inertial load speed step response. The step response of the drive motor speed, however, will exhibit a significant overshoot. This overshoot grows when the closed loop bandwidth is increased, and thus introduces large torsion torques in the connecting shaft during transients. The achievable bandwidth is therefore limited by the maximum torque that can be transferred by the shaft.

Limit cycles and their relation to regulator stability

Further performance constraints are introduced by the nonlinear friction terms. Limit cycle oscillations, caused by the Coulomb friction in the drive motor, appear at zero speed reference if the desired closed loop bandwidth is pushed too far. The existence of these oscillations, as well as their amplitude and frequency, can be predicted by the describing function method in combination with a simple nonlinear friction model. It has been proved that an unstable regulator is a sufficient condition for the appearance of these limit cycles. Unfortunately we have no proof that this is also a necessary condition. Thus we cannot guarantee that limit cycles can be avoided by requiring a stable regulator. In all cases that have been tested so far, however, limit cycle appearance has coincided with an unstable regulator. It is therefore conjectured that regulator instability is also a necessary condition for limit cycles to appear. This, in combination with the inherent windup problem, leads to the conclusion that unstable regulators if possible should be avoided in control design for flexible servo systems.

Bandwidth constraints

It has been shown that the stability of the regulator mainly depends on a dimensionless quantity w defined by

$$w = \frac{J_2\omega^2}{k}$$

where J_2 is the load inertia, ω is the closed loop natural frequency and k is the shaft spring constant. When the drive motor speed is measured, the regulator will be unstable for large w values and hence limit cycles will appear. Analytic expressions have been derived from which approximate values of the w stability limit can be calculated with good accuracy.

Non-colocated speed sensor

No limit cycles appear in the nominal design case with a non-colocated speed sensor, i.e. when the inertial load speed is measured and used as input to the regulator. This can be explained with the describing function method. It also agrees with the conjecture that regulator instability is a sufficient and necessary condition for limit cycles, since the corresponding regulator is stable. With a non-colocated speed sensor, the regulator does not go unstable when the closed loop bandwidth is increased. The conclusion is that with this sensor location the limit cycle problem may be avoided in applications where fast response is required.

Friction compensation

Friction compensation schemes, based on different approximations of the ideal relay friction model, have been tested. Simulations indicate that a continuous time friction compensation works well, i.e. it reduces the limit cycle amplitude to a negligible level.

Implementation and experimental results

A continuous time linear state feedback controller has been built and tested on the prototype flexible servo system. Most experimental results agree well with the simulations. With a low closed loop bandwidth, the linear controller works well and gives a well damped inertial load step response. Limit cycle oscillations appear as predicted when the bandwidth is increased. The observed amplitude and frequency agree with the calculated values with reasonable accuracy, considering the approximate nature of the describing function method. The continuous time friction compensation scheme, however, does not work as well in practice as in the simulations. The limit cycle amplitude is reduced, but the oscillations are not eliminated. It is believed that this depends on the friction models being too simple. The use of unstable regulators requires extra precautions in the implementation to avoid windup problems during startup or if the loop is at any time broken. These problems can be handled with normal anti-windup methods.

The experimental results show that simple models can be used to analyse and predict the behavior of flexible servo systems. They give good insight into qualitative properties and also, in most cases, quite accurate quantitative results. Nonlinear friction compensation seems to be a difficult problem, however.

Suggestions for further work

Many conventional servo control systems consist of two cascaded loops, one outer position loop and one inner speed loop. This work can be applied directly to the inner loop of such systems. It would of course be of great practical interest to apply these methods to the position control problem, and also to investigate the implications of integral action in the regulator. Another possible extension is to study more complicated flexible systems, e.g. systems with multiple inertias in combination with nonlinear friction. The problem complexity will then increase rapidly, however.

Friction compensation probably requires more detailed friction models than the simple relay-type functions used here. Another problem is that the friction torque also is time-varying due to temperature changes, mechanical wear etc. This naturally suggests the use of adaptive friction compensation, see [1], [3] and [4].

In robotic applications, one major problem is the inertia variations due to different arm configurations. Active damping of the resonant modes with pole placement requires that the resonance frequency is known with reasonable accuracy. Since the resonance frequency depends on the load inertia, a varying inertia will give degraded performance. Robotic applications therefore require that inertia variations are taken into account in the control design.

9. REFERENCES

[1] Anwar, G., Tomizuka, M., Horowitz, R. and Kubo, T.: "Experimental Study on Discrete Time Adaptive Control of an Industrial Robot Arm". Proceedings of the 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing, July 1986. Lund Institute of Technology, Lund, Sweden (1986).

[2] Atherton, D.P.: "Nonlinear Control Engineering". Van Nostrand Reinhold (1975). (Student edition 1982).

[3] Canudas de Wit, C.: "Adaptive Control for Partially Known Systems: Theory and Applications" (Thesis). Laboratoire d'Automatique de Grenoble, GRECO(60) "Systemes Adaptatifs" E.N.S.I.E.G., B.P.46, 38402 Saint Martin d'Heres, France (1987).

[4] Canudas de Wit, C., Åström, K.J. and Braun, K.: "Adaptive Friction Compensation in DC Motor Drives". IEEE Conference on Robotics and Automation, San Francisco, USA (1986). To appear in IEEE transactions on Robotics and Automation.

[5] Ctrlc-C User's Guide. Systems Control Technology, Inc., Palo Alto, CA, USA (1986).

[6] Elmqvist, H.: "SIMNON - User's Manual". Report no. 7502, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden (1975).

[7] Elmqvist, H., Åström, K.J., and Schönthal, T.S.: "SIMNON - User's Guide for MS-DOS Computers". Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden (1986).

[8] Franklin, G.F., Powell, J.D. and Emami-Naeini, A.: "Feedback Control of Dynamic Systems". Addison-Wesley (1986).

[9] Gibson, J.: "Nonlinear Automatic Control". McGraw-Hill (1963).

[10] Horowitz, I.M.: "Synthesis of Feedback Systems". Academic Press, New York (1963).

[11] Levinson, N. and Redheffer, R.: "Complex Variables". Holden-Day, Oakland (1970).

[12] Lundh, M.: "Adaptive Friction and Load Compensation Based on Shaft Angle Measurement in DC-Motor Servos". Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden. LUTFD2/(TFRT-7331)/1-10/(1986).

[13] MACSYMA Reference Manual. Symbolics Inc., Cambridge, Mass., USA (1983).

[14] Sidman, M.D. and Franklin, G.F.: "Adaptive Control of a Flexible Structure". Proceedings of the 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing, July 1986. Lund Institute of Technology, Lund, Sweden (1986).

[15] Siljak, D.: "Nonlinear Systems - the Parameter Analysis and Design". John Wiley & Sons, Inc. (1969).

[16] Tsypkin, Y.Z.: "Relay Control Systems". Cambridge University Press (1984).

[17] Vidyasagar, M.: "Nonlinear Systems Analysis". Prentice-Hall (1978).

[18] Åström, K.J. and Wittenmark, B.: "Computer Controlled Systems - Theory and Design". Prentice-Hall (1984).

Appendix A

Most of the numerical calculations in this thesis were done with Ctrl-C, a product of Systems Control Technology, Inc. Ctrl-C is an interactive program with commands for control system analysis and design, matrix analysis, digital signal processing, and graphics. It is particularly useful for control problems on state space form, due to the powerful matrix computation facilities. For further details, see [5]. This appendix contains a collection of Ctrl-C programs used in the control system analysis and design.

Contents:

```
// SERVOPAR.CTR
//
// Flexible servo parameters (wire # 3)
//
J1 = 22e-6
J2 = 150e-6
k  = 2.4e-3
d  = 0.0
d1 = 1.0e-5
d2 = 1.0e-5
km = 0.1
ki = 0.25
kw1 = 0.1
kw2 = 0.1
```

```
// SERVOSS.CTR
//
// Continuous state space model of flexible servo speed loop.
//
A = [-(d1+d)/J1     d/J1     k/J1
      d/J2      -(d2+d)/J2 -k/J2
       -1           1        0     ]
B = [ki*km/J1 0 0]'
C = [kw1 0 0]
```

```
// CD.CTR
//
// Ctrl-c procedure for design of continuous state feedback
// control for flexible servo.
// Enter servo model (A,B,C) and design parameters (wcl, z and alfa)
// before using this procedure.
//
// External function references:
//   CPD3    calculates desired poles
//   CSFP    converts parameters to SIMNON file
//
// Author: Anders Wallenborg
//
P = cpd3(z,wcl,wcl)           // closed loop poles
L = place(A,B,P)
P = cpd3(z,alfa*wcl,alfa*wcl) // observer poles
Ko = place(A',C',P)'
clear p
gain = C*((-A+B*L)\B)         // closed loop steady state gain
lr = 1/gain
csfp(A,B,C,Ko,L,lr,'reg')
```

```
// CDS.CTR
//
// Ctrl-c procedure for design of continuous state feedback
// control for flexible servo. Scaling of state variables included.
// Enter servo model (A,B,C) and design parameters (wcl, z, alfa)
// before using this procedure.
//
// External function references:
//   CPD3    calculates desired poles
//   CSFP    converts parameters to SIMNON file
//   TRSS    transformation of state space description
//
// Author: Anders Wallenborg
//
T = [0.1 0 0 ;
     0 0.1 0 ;
     0  0  2 ]
[Az,Bz,Cz] = trss(A,B,C,T)
P = cpd3(z,wcl,wcl)            // closed loop poles
L = place(Az,Bz,P)
P = cpd3(z,alfa*wcl,alfa*wcl) // observer poles
Ko = place(Az',Cz',P)'
gain = Cz*((-Az+Bz*L)\Bz)      // closed loop steady state gain
lr = 1/gain
clear p gain
//
csfp(Az,Bz,Cz,Ko,L,lr,'reg')
```

```
// DD.CTR
//
// Ctrl-c procedure for design of digital state feedback
// control and observer using latest measured value.
//
// Enter servo parameters (A, B, C), sample interval (h), and
// design parameters (wcl, z, alfa) before using this procedure.
//
// External function references:
//   DPD3    calculates desired discrete poles
//   DSFP    generates SIMNON parameter file
//
// Author: Anders Wallenborg
//
[F,G] = c2d(A,B,h)
P = dpd3(z,wcl,wcl,h)              // closed loop poles
L = place(F,G,P)
P = dpd3(z,alfa*wcl,alfa*wcl,h)    // observer poles
Kobs = place(F',(C*F)',P)'
clear p
gain = C*((eye(3)-F+G*L)\G)        // closed loop steady state gain
lr = 1/gain
dsfp(F,G,C,Kobs,L,lr,h,'reg')
```

```
// F1W1.CTR
//
// Ctrl-c procedure for generating closed loop system matrices Acl, Bcl & Ccl
// for the flexible servo controlled with state feedback & reconstruction.
// (Motor speed (w1) measured, static friction on motor).
//
// Note: Enter servo parameters & control design parameters
//       (wcl, a, alfa) before using this procedure.
//
// External functions used:
//   CPD3   calculates desired poles
//
// Author: Anders Wallenborg
//
A = [-(d1+d)/J1      d/J1   k/J1
      d/J2      -(d2+d)/J2 -k/J2
       -1           1       0     ];
B = [1/J1 0 0]';                        // Note: input = motor torque
C = [kw1 0 0];
//
p = cpd3(z,wcl,wcl);                    // desired closed loop poles
L = place(A,B,P);                       // state feedback design
//
p = cpd3(z,alfa*wcl,alfa*wcl);          // desired observer poles
Ko = place(A',C',P)';                   // observer design
//
// Closed loop system with state feedback and observer
//
Acl = [ A         -B*L
        Ko*C  A-B*L-Ko*C ];
Bcl = [B' 0 0 0]';
Ccl = [C 0 0 0];


// F1W2.CTR
//
// Ctrl-c procedure for generating the closed loop system matrices for
// the flexible servo controlled with state feedback & reconstruction.
// Load speed (y2) measured, static friction on motor.
//
// Note: Enter servo parameters & regulator design parameters
//       (wcl, z, alfa) before using this procedure
//
// External functions used:
//    CPD3   calculates desired poles
//
// Author: Anders Wallenborg
//
A = [-(d1+d)/J1      d/J1   k/J1
      d/J2      -(d2+d)/J2 -k/J2
       -1           1       0     ]
B = [1/J1 0 0]'                         // Note: input = motor torque
C1 = [kw1 0 0]
C2 = [0 kw2 0]
//
```

```
P = cpd3(z,wcl,wcl)                  // desired closed loop poles
L = place(A,B,P)
P = cpd3(z,alfa*wcl,alfa*wcl)        // desired observer poles
Ko = place(A',C2',P)'
//
// Closed loop system with state feedback and observer
//
Acl = [ A         -B*L
        Ko*C2  A-B*L-Ko*C2 ]
Bcl = [B' 0 0 0]'
Ccl = [C1 0 0 0]                     // closed loop output = motor speed


// [p] = CPD3(z,w,a)
//
// Calculates a vector p with eigenvalues corresponding
// to the char. equation
//
//         (s+a)(s*s+2zws+w*w)=0
//
// Author: Anders Wallenborg
//
p1 = -a;
p2 = -z*w + w*sqrt(1-z*z)*sqrt(-1);
p3 = -z*w - w*sqrt(1-z*z)*sqrt(-1);
p = [p1;p2;p3];


[] = CSFP(A,B,C,K,L,lr,syst);
//
// Conversion of state feedback controller parameters to a
// SIMNON parameter file with the system name 'syst'.
//
//   A = A matrix
//   B = B matrix
//   C = C matrix
//   K = observer gain vector
//   L = state feedback gain vector
//   lr= reference signal gain
//   syst = SIMNON system name (should be entered within single quotes)
//
//   Output file name: CSFPAR.T
//   Maximum system order = 5
//
// Author: Anders Wallenborg
//
[m,n] = size(a);              // n = system order
syst=['[',syst,']'];
astr=['a11:';'a12:';'a13:';'a14:';'a15:';
      'a21:';'a22:';'a23:';'a24:';'a25:';
      'a31:';'a32:';'a33:';'a34:';'a35:';
      'a41:';'a42:';'a43:';'a44:';'a45:';
      'a51:';'a52:';'a53:';'a54:';'a55:'];
astr = [astr(1:n,:);
```

```
        astr(6:5+n,:);
        astr(11:10+n,:);
        astr(16:15+n,:);
        astr(21:20+n,:)];
bstr = ['b1:  ';'b2:  ';'b3:  ';'b4:  ';'b5:  '];
cstr = ['c1:  ';'c2:  ';'c3:  ';'c4:  ';'c5:  '];
kstr = ['k1:  ';'k2:  ';'k3:  ';'k4:  ';'k5:  '];
lstr = ['l1:  ';'l2:  ';'l3:  ';'l4:  ';'l5:  '];
lrstr = 'lr:  ';
//
for i=1:n, ...
   bpar(i,:) = str(b(i)); , ...
   cpar(i,:) = str(c(i)); , ...
   kpar(i,:) = str(k(i)); , ...
   lpar(i,:) = str(l(i)); , ...
   for j=1:n, ...
      apar((i-1)*n+j,:) = str(a(i,j)); , ...
   end, ...
end;
lrpar = str(lr);
//
diary >csfpar.t ;
disp(syst);
disp([astr(1:n*n,:) apar;
      bstr(1:n,:)   bpar;
      cstr(1:n,:)   cpar;
      kstr(1:n,:)   kpar;
      lstr(1:n,:)   lpar;
      lrstr         lrpar ])
diary -off;


// [p] = DPD3(z,w,alfa,h)
//
// Generates a vector p with the poles of a discrete system
// corresponding to a continuous system with char. equation
//
//   (s+alfa)(s*s+2zws+w*w)=0
//
// and sample interval = h
//
// Author: Anders


p(2) = -z*w + i*w*sqrt(1-z*z);
p(3) = -z*w - i*w*sqrt(1-z*z);
for j = 1:3, ...
   p(j) = exp(p(j)*h); ...
end;
```

```
[] = DSFP(F,G,C,K,L,lr,h,syst);
//
// Conversion of discrete state feedback controller parameters to a
// SIMNON parameter file with the system name 'syst'.
//
//   F = Fi matrix
//   G = Gamma matrix
//   C = C matrix
//   K = observer gain vector
//   L = state feedback gain vector
//   lr= reference signal gain
//   h = sample interval (sec)
//   syst = SIMNON system name (should be entered within single quotes)
//
//   Output file name: DSFPAR.T
//   Maximum system order = 5
//
// Author: Anders Wallenborg
//
[m,n] = size(f);                    //  n = system order
syst=['[',syst,']'];
fstr=['f11:';'f12:';'f13:';'f14:';'f15:';
      'f21:';'f22:';'f23:';'f24:';'f25:';
      'f31:';'f32:';'f33:';'f34:';'f35:';
      'f41:';'f42:';'f43:';'f44:';'f45:';
      'f51:';'f52:';'f53:';'f54:';'f55:'];
fstr = [fstr(1:n,:);
        fstr(6:5+n,:);
        fstr(11:10+n,:);
        fstr(16:15+n,:);
        fstr(21:20+n,:)];
gstr = ['g1: ';'g2: ';'g3: ';'g4: ';'g5: '];
cstr = ['c1: ';'c2: ';'c3: ';'c4: ';'c5: '];
kstr = ['k1: ';'k2: ';'k3: ';'k4: ';'k5: '];
lstr = ['l1: ';'l2: ';'l3: ';'l4: ';'l5: '];
gstr = gstr(1:n,:);
csrt = cstr(1:n,:);
kstr = kstr(1:n,:);
lstr = lstr(1:n,:);
lrstr = 'lr: ';
hstr = 'h:  ';
//
for i=1:n, ...
   gpar(i,:) = str(g(i));, ...
   cpar(i,:) = str(c(i));, ...
   kpar(i,:) = str(k(i));, ...
   lpar(i,:) = str(l(i));, ...
   for j=1:n, fpar((i-1)*n+j,:) = str(f(i,j));, end, ...
end;
lrpar = str(lr);
hpar = str(h);
//
diary >dsfpar.t ;
disp(syst);
disp([hstr          hpar;
      fstr(1:n*n,:) fpar;
```

```
      gstr(1:n,:)   gpar;
      cstr(1:n,:)   cpar;
      kstr(1:n,:)   kpar;
      lstr(1:n,:)   lpar;
      lrstr         lrpar ])
diary -off;
```

```
// [w] = LGSPACE(lgw1,lgw2,n)
//
// Creates a vector of n logaritmically spaced vector
// elements between 10**(lgw1) and 10**(lgw2).
//
// Author: Anders Wallenborg
//
ln10 = log(10);
w = exp((lgw1:((lgw2-lgw1)/(n-1)):lgw2)*ln10);
```

```
// [fr] = NYPL(a,b,c,d,lgw1,lgw2,n)
//
// Nyquist plot for system on state space form
//
//   a,b,c,d = system matrices
//   lgw1 = log(start frequency)
//   lgw2 = log(stop frequency)
//   n    = number of frequency points
//
// Author: Anders Wallenborg
//
w = lgspace(lgw1,lgw2,n);
[Re,Im] = nyqu(a,b,c,d,1,w);
fr = [w' Re' Im'];
plot(Re,Im);
xlabel('Re',' 1');
ylabel('Im',' 1');
```

```
// [a2n,a3n,en,w] = RCP1(w1,dw,w2,z,a,j1,j2)
//
// Normalized coefficients in the char. polynomial of the state feedback
// observer/controller for the flexible servo with colocated speed
// sensor (simplified model).
// Coefficients are calculated and plotted as a function of the
// dimensionless parameter w.
//
// Author: Anders Wallenborg
//
w = w1;
x = w1;
a2n = -( (a**3)*x**3 -(4*(a**2)*z**2+2*a*(a+1)**2*z+a**3+a**2+a)*x**2 ...
       +(4*a*z**2+2*(a+1)**2*z+a**2+a+1)*x -(1+J2/J1) );
a3n = ( a**2*(2*z+1)*(a+1)*x**2 -(4*a*(a+1)*(z**2+z)+a**3+a**2+a+1)*x ...
```

```
      +(a+1)*(2*z+1)*(1+J2/J1) );
en = -( a**3*(a+1)*(2*z+1)*x**2 ...
      -(8*a**2*(a+1)*z**3+4*(a**4+4*a**3+4*a**2+a)*z**2 ...
        +4*(a**4+2*a**3+2*a**2+a)*z+(a**4+a**3+a**2+a))*x ...
       +8*a*(a+1)*z**3+4*(a**3+3*a**2+3*a+1)*z**2 ...
       +2*(2*a**3+3*a**2+3*a+2)*z+a*(a+1) );
//
for x = w1+dw:dw:w2, ...
   w = [w;x]; ...
   a2 = -( (a**3)*x**3 -(4*(a**2)*z**2+2*a*(a+1)**2*z+a**3+a**2+a)*x**2 ...
          +(4*a*z**2+2*(a+1)**2*z+a**2+a+1)*x -(1+J2/J1) ); ...
   a2n = [a2n;a2]; ...
   a3 = ( a**2*(2*z+1)*(a+1)*x**2 -(4*a*(a+1)*(z**2+z)+a**3+a**2+a+1)*x ...
         +(a+1)*(2*z+1)*(1+J2/J1) ); ...
   a3n = [a3n;a3]; ...
   e = -( a**3*(a+1)*(2*z+1)*x**2 ...
          -(8*a**2*(a+1)*z**3+4*(a**4+4*a**3+4*a**2+a)*z**2 ...
            +4*(a**4+2*a**3+2*a**2+a)*z+(a**4+a**3+a**2+a))*x ...
          +8*a*(a+1)*z**3+4*(a**3+3*a**2+3*a+1)*z**2 ...
          +2*(2*a**3+3*a**2+3*a+2)*z+a*(a+1) ); ...
    en = [en;e]; ...
end;
//
eras;
plot(w,a2n,'solid',w,a3n,'dashed',w,en,'dotted');
xlabel('w');
ylabel('a2 (solid), a3 (dash), e (dot)');



// [a2n,a3n,w] = RCP2(w1,dw,w2,z,a,j1,j2)
//
// Normalized coefficients in the char. polynomial of the state feedback
// observer/controller for the flexible servo with non-colocated speed
// sensor (simplified model).
// Coefficients are calculated and plotted as a function of the
// dimensionless parameter w.
//
// Author: Anders Wallenborg
//
w = w1;
x = w1;
a2n = ( 4*a*z**2 + (2*a**2+4*a+2)*z + (a**2+a+1) )*x - (1+j2/j1);
a3n = ( (4*a**2+4*a)*z**2 + (4*a**2+4*a)*z + (a**3+a**2+a+1) )*x ...
      - ((2*a+2)*j2/j1+(2*a+2))*z - ((a+1)+(a+1)*j2/j1);
//
for x = w1+dw:dw:w2, ...
   w = [w;x]; ...
   a2 = ( 4*a*z**2 + (2*a**2+4*a+2)*z + (a**2+a+1) )*x - (1+j2/j1); ...
   a2n = [a2n;a2]; ...
   a3 = ( (4*a**2+4*a)*z**2 + (4*a**2+4*a)*z + (a**3+a**2+a+1) )*x ...
        - ((2*a+2)*j2/j1+(2*a+2))*z - ((a+1)+(a+1)*j2/j1); ...
   a3n = [a3n;a3]; ...
end;
//
eras;
```

```
plot(w,a2n,'solid',w,a3n,'dashed');
xlabel('w');
ylabel('a2 (solid)  a3 (dashed)');


// [Az,Bz,Cz] = TRSS(A,B,C,T)
//
// Transformation of state space system description  S=(A,B,C)
// to new coordinates z=Tx
//
// Author: Anders Wallenborg
//
Az = A/T;
Az = T*Az;
Bz = T*B;
Cz = C/T;
```

Appendix B

The simulations in this thesis were done with SIMNON, a product of the Department of Automatic Control, Lund Institute of Technology. SIMNON is an interactive simulation program for nonlinear continuous and discrete time systems. For further details, see [6] and [7].
This appendix contains a collection of SIMNON system descriptions, parameter files and simulation macros.

Contents:

```
CONTINUOUS SYSTEM SERVO

" Flexible servo system with two moments of inertia
" (linear model).
" File name: SERVO.T
" Author: Anders Wallenborg

INPUT  u        " motor drive amplifier input [V]
OUTPUT y1       " shaft 1 tacho signal [V]
OUTPUT y2       " shaft 2 tacho signal [V]
STATE  w1       " shaft 1 angular velocity [rad/s]
STATE  w2       " shaft 2     - " -        [rad/s]
STATE  fi       " position difference (shaft 2 - shaft 1) [rad]
DER    dw1 dw2 dfi

" system dynamics ------------------------------------------------

Mi = ki*km*u                                    " motor torque
dw1 = (-(d1+d)*w1 + d*w2 + k*fi + Mi)/J1
dw2 = (d*w1 - (d2+d)*w2 - k*fi)/J2
dfi = -w1 +w2
y1 = kw1*w1
y2 = kw2*w2

" parameter values -----------------------------------------------

ki : 0.25       " drive amp gain [A/V]
km : 0.1        " motor current to torque gain [Nm/A]
J1 : 22.0E-6    " moment of inertia of motor [kgm2]
J2 : 150.0E-6   " shaft 2 inertia [kgm2]
k  : 2.4E-3     " transmission rigidity [Nm/rad]
d  : 0.0        " damping coefficient of transmission [Nm/rad/s]
d1 : 1.0E-5     " viscous friction coefficient of shaft 1 [Nm/rad/s]
d2 : 1.0E-5     "       - " -                 of shaft 2 [Nm/rad/s]
kw1: 0.1        " shaft 1 tacho gain [V/rad/s] (including ext. amplifier)
kw2: 0.1        " shaft 2 tacho gain [V/rad/s]

END
```

```
CONTINUOUS SYSTEM servo

" Flexible servo system with two moments of inertia.
" Nonlinear model with static and Coulomb friction.
" File name: SERVONL.T
" Author: Anders Wallenborg

INPUT  u        " motor drive amplifier input [V]
OUTPUT y1       " shaft 1 tachometer signal [V]
OUTPUT y2       " shaft 2 tachometer signal [V]
STATE  w1       " shaft 1 angular velocity [rad/s]
STATE  w2       " shaft 2     - " -        [rad/s]
STATE  fi       " position difference (shaft 2 - shaft 1) [rad]
DER    dw1 dw2 dfi

" system dynamics ------------------------------------------------------

Mi = ki*km*u                                        " input torque from motor
M1 = k*fi -d1*w1 -d*(w1-w2) +Mi                     " net linear torque
M2 = -k*fi -d2*w2 +d*(w1-w2)
Mf10 = if abs(M1)>F1 then F1*sign(M1) else M1       " Coulomb friction
Mf20 = if abs(M2)>F2 then F2*sign(M2) else M2
Mf1 = if abs(w1)<eps then Mf10 else F1*sign(w1)
Mf2 = if abs(w2)<eps then Mf20 else F2*sign(w2)
dw1 = (M1-Mf1)/J1
dw2 = (M2-Mf2)/J2
dfi = -w1 +w2
y1 = kw1*w1
y2 = kw2*w2

" parameter values -----------------------------------------------------

pi : 3.141593
eps: 0.001
ki : 0.25       " drive amp gain [A/V]
km : 0.1        " motor current to torque gain [Nm/A]
J1 : 22.0E-6    " moment of inertia of motor [kgm2]
J2 : 150.0E-6   " shaft 2 inertia [kgm2]
k  : 2.4E-3     " transmission rigidity [Nm/rad]
d  : 0.0        " damping coefficient of transmission [Nm/rad/s]
d1 : 1.0E-5     " viscous friction coefficient shaft 1 [Nm/rad/s]
d2 : 1.0E-5     "       - " -                  shaft 2 [Nm/rad/s]
kw1: 0.1        " shaft 1 tacho gain [V/rad/s] (including ext. amplifier)
kw2: 0.1        " shaft 2 tacho gain [V/rad/s]
F1 : 5.E-4      " static friction on shaft 1 [Nm]
F2 : 5.E-4      " static friction on shaft 2 [Nm]

END
```

```
CONTINUOUS SYSTEM reg

" State feedback controller with full state reconstruction and
" friction compensation.
" The friction compensation scheme is controlled with the switch 'fc':
"
"   fc = 0  No friction compensation (linear controller)
"        1  High gain with saturation
"        2  Relay with dead zone
"
" File name: CSFNL.T
" Author: Anders Wallenborg

INPUT yr                " reference signal
INPUT y                 " measured signal
OUTPUT u                " control output
STATE x1e               " estimate of w1
STATE x2e               "   - " -     w2
STATE x3e               " estimated position difference
DER dx1e dx2e dx3e

" linear controller ------------------------------------------------

dx1e = a11*x1e + a12*x2e + a13*x3e + b1*u0 + k1*(y-ye)
dx2e = a21*x1e + a22*x2e + a23*x3e + b2*u0 + k2*(y-ye)
dx3e = a31*x1e + a32*x2e + a33*x3e + b3*u0 + k3*(y-ye)
ye = c1*x1e + c2*x2e + c3*x3e
v = lr*yr -l1*x1e -l2*x2e -l3*x3e
u0 = if v<umin then umin else if v<umax then v else umax

" friction compensation ---------------------------------------------

Mf1 = if abs(ye)<eps then F*ye/eps else F*sign(ye)
Mf2 = if abs(ye)<eps then 0 else F*sign(ye)
Mf = if not fc then 0 else if fc<1.5 then Mf1 else Mf2
uf = Mf/(km*ki)

" output control signal ---------------------------------------------

u = u0 + uf

" parameters --------------------------------------------------------

umax : 8.0      " maximum control output
umin :-8.0      " minimum     - " -
fc : 0.0        " friction compensation switch (0=off)
eps: 0.001
ki : 0.25       " drive amp gain [A/V]
km : 0.1        " motor current to torque gain [Nm/A]
F  : 5.E-4      " static friction on shaft 1 [Nm]
lr : 0.0        " reference signal gain
l1 : 0.0        " state feedback gain vector
l2 : 0.0
l3 : 0.0
k1 : 0.0        " observer gain vector
k2 : 0.0
```

```
k3 :  0.0
a11:  0.0          " A matrix elements
a12:  0.0
a13:  0.0
a21:  0.0
a22:  0.0
a23:  0.0
a31:  0.0
a32:  0.0
a33:  0.0
b1 :  0.0          " B matrix elements
b2 :  0.0
b3 :  0.0
c1 :  0.0          " C matrix elements
c2 :  0.0
c3 :  0.0

END
```

```
 [REG]

" Continuous regulator parameters
" Nominal control design (y1 measured): wcl=12, z=0.7, alfa=1.5
" File name: CSFPAR1.T

 A11: -0.4545
 A12:   0.0000E+0
 A13:   109.1
 A21:   0.0000E+0
 A22: -6.6667E-02
 A23:  -16.00
 A31:  -1.000
 A32:   1.000
 A33:   0.0000E+0
 B1:    1136.
 B2:    0.0000E+0
 B3:    0.0000E+0
 C1:   0.1000
 C2:    0.0000E+0
 C3:    0.0000E+0
 K1:    426.8
 K2:    466.7
 K3:    59.55
 L1:   2.4885E-02
 L2:   6.8553E-02
 L3:  -0.1924
 LR:   0.9504
```

```
[REG]

" Continuous regulator parameters
" Reduced bandwidth design (y1 measured): wcl=8, z=0.7, alfa=1.5
" File name: CSFPAR2.T

A11: -0.4545
A12:   0.0000E+0
A13:   109.1
A21:   0.0000E+0
A22: -6.6667E-02
A23:  -16.00
A31:  -1.000
A32:   1.000
A33:   0.0000E+0
B1:    1136.
B2:    0.0000E+0
B3:    0.0000E+0
C1:   0.1000
C2:    0.0000E+0
C3:    0.0000E+0
K1:    282.8
K2:    114.3
K3:    20.04
L1:   1.6437E-02
L2:   1.0823E-02
L3:  -2.3966E-02
LR:   0.2816
```

```
[REG]

" Continuous regulator parameters
" Nominal design (y2 measured): wcl=12, z=0.7, alfa=1.5
" File name: CSFPAR3.T

A11: -0.4545
A12:   0.0000E+0
A13:   109.1
A21:   0.0000E+0
A22: -6.6667E-02
A23:  -16.00
A31:  -1.000
A32:   1.000
A33:   0.0000E+0
B1:    1136.
B2:    0.0000E+0
B3:    0.0000E+0
C1:    0.0000E+0
C2:   0.1000
C3:    0.0000E+0
K1:    546.1
K2:    426.8
```

```
K3:   -395.7
L1:   2.4885E-02
L2:   6.8553E-02
L3:  -0.1924
LR:   0.9504


DISCRETE SYSTEM reg

" Discrete time state feedback controller with full state
" reconstruction (using latest measurement).
" Nonlinear friction compensation included.
" The friction compensation scheme is controlled with the switch 'fc':
"
" fc = 0    No friction compensation
"      1    High gain with saturation
"      2    Relay with deadzone
"
" File name: DSFNL.T
" Author: Anders Wallenborg

INPUT yr                " reference signal
INPUT y                 " measured signal
OUTPUT u                " control output
STATE x1o  x2o  x3o     " observer states [xe(k|k-1)]
NEW nx1o nx2o nx3o
TIME t
TSAMP ts

" linear controller ------------------------------------------------------

yo = c1*x1o + c2*x2o + c3*x3o
x1e = x1o + k1*(y-yo)
x2e = x2o + k2*(y-yo)
x3e = x3o + k3*(y-yo)
v = lr*yr -l1*x1e -l2*x2e -l3*x3e
u0 = if v>umax then umax else if v<umin then umin else v
nx1o = f11*x1e + f12*x2e + f13*x3e + g1*u0
nx2o = f21*x1e + f22*x2e + f23*x3e + g2*u0
nx3o = f31*x1e + f32*x2e + f33*x3e + g3*u0
ts = t+h

" friction compensation --------------------------------------------------

ye = c1*x1e+c2*x2e+c3*x3e
Mf1 = if abs(ye)<eps then F*ye/eps else F*sign(ye)
Mf2 = if abs(ye)<eps then 0 else F*sign(ye)
Mf = if fc<1.5 then Mf1 else Mf2
uf = if not fc then 0 else Mf/(km*ki)

" output control signal --------------------------------------------------

u = u0 + uf

" parameters -------------------------------------------------------------
```

```
umax :  8.0      " maximum control output [V]
umin : -8.0      " minimum     - " -      [V]
fc :  0.0        " friction compensation switch
eps:  0.001
F  :  5.E-4      " static friction on shaft 1 [Nm]
ki :  0.25       " drive amp gain [A/V]
km :  0.1        " motor current to torque gain [Nm/A]
h  :  1.0        " sample interval [sec]
lr :  0.0        " reference signal gain
l1 :  0.0        " state feedback gain vector
l2 :  0.0
l3 :  0.0
k1 :  0.0        " observer gain vector
k2 :  0.0
k3 :  0.0
f11:  0.0        " Fi matrix elements
f12:  0.0
f13:  0.0
f21:  0.0
f22:  0.0
f23:  0.0
f31:  0.0
f32:  0.0
f33:  0.0
g1 :  0.0        " Gamma matrix elements
g2 :  0.0
g3 :  0.0
c1 :  0.0        " C matrix elements
c2 :  0.0
c3 :  0.0

END


 [REG]

" Nominal discrete time control parameters (y1 measured)
" (wcl=12, z=0.7, alfa=1.5, h=0.040)
" Observer using latest measured value
" File name: DSFPAR1.T

 H:    4.0000E-02
 F11:  0.8972
 F12:  8.5234E-02
 F13:   4.181
 F21:  1.2501E-02
 F22:  0.9848
 F23: -0.6181
 F31: -3.8329E-02
 F32:  3.8632E-02
 F33:  0.9021
 G1:    43.75
 G2:   0.1910
 G3:  -0.8886
 C1:   0.1000
```

```
C2:   0.0000E+00
C3:   0.0000E+00
K1:    8.186
K2:    8.155
K3:    1.056
L1:   1.6797E-02
L2:   3.7445E-02
L3:  -8.6916E-02
LR:   0.5540


CONNECTING SYSTEM stepc

" Servo speed loop with step reference input
" Author: Anders Wallenborg

TIME t

y[reg] = if my1 then y1[servo] else y2[servo]
yr[reg] = if t<t1 then 0 else if t<t2 then vref else 0
u[servo] = u[reg]

t1 : 2.0
t2 : 5.0
vref : 1.0          " speed reference
my1 : 1.0           " measurement switch (1=shaft 1, 0=shaft 2)

END


MACRO stepr

" Simulation of speed loop step response.
" Author: Anders Wallenborg

STORE u[reg]
PLOT y1 y2 yr[reg]
SPLIT 2 1
AXES h 0 10 v -2 3
SIMU 0 10 / stepr 0.02
TEXT ' y1 y2 [V]'
MARK a 16.5 6.7
MARK "t [sec]
AXES h 0 10 v -1 1
SHOW u /stepr
TEXT ' u [V]'
MARK a 16.5 0
MARK "t [sec]

END
```

Appendix C

MACSYMA is a general symbolic manipulation program from Symbolics Inc., Cambridge, Mass., USA. It is a valuable tool for performing otherwise tedious algebraic calculations. For further details, see [13]. This appendix contains the MACSYMA programs that were used to calculate the transfer functions in Chapter 3 and the coefficients of the characteristic equation for the regulator in Chapter 6.

Contents:

```
/*  SERVOSS.BAT

    Flexible servo state space model

*/
a: matrix([-(dd1+d)/j1,   d/j1,      k/j1],
          [     d/j2,  -(dd2+d)/j2, -k/j2],
          [       -1,        1,         0  ])$

b: matrix([km*ki/j1], [0], [0])$

c: [kw1, 0, 0]$



/* TRFUNC.DEF

   Calculates the transfer function of a (SISO) system given on
   state space form:

        .
        x = Ax + Bu
        y = Cx + Du

   Author: Anders Wallenborg
*/
trfunc(a,b,c,d) := c.((diagmatrix(length(a),s)-a)^^-1).b + d$



/* SERVOTRF.BAT

   Transfer functions for flexible servo
   Author: Anders Wallenborg

*/
batch("trfunc.def");
batch("servoss.bat");
writefile("servotrf.log")$
a;
b;
c;
g1: ratsimp(trfunc(a,b,c,0));
c: [0, kw2, 0];
g2: ratsimp(trfunc(a,b,c,0));

/* simplified transfer functions with d=0 */

ratsubst(0,d,g1);
ratsubst(0,d,g2);

closefile();
```

```
/*-----------------------------------------------------------------
ACKERMANN.DEF

  Function for single input state feedback gain calculation
  with Ackermanns formula.
  Ref. Kailath: "Linear Systems", p. 201
  It works in the same way as PLACE in CTRLC.

  Inputs:  A   matrix from system desription
           B   matrix from system desription
           P   vector with desired closed loop poles

  Output:  Vector with feedback gains

  Author: Michael Lundh
  Revised: Anders Wallenborg /861110
  ----------------------------------------------------------------*/
ackermann(a,b,p):=
  Block([n,cc,r,tmp,i,q,pa,l],
  n:length(b),
  r:ematrix(1,n,1,1,n),
  /* create controllability matrix */
  cc:b,
  tmp:b,
  for i:1 thru n-1 do (tmp:a.tmp, cc:addcol(cc,tmp) ),
  q:r.invert(cc),
  /* Evaluate A in new characteristic polynomial */
  pa:ident(n),
  for i:1 thru n do pa:pa.(a-determinant(p[i])*ident(n)),
  l:q.pa,
  for i:1 thru n do l[1,i]:ratsimp(l[1,i]),
  return:l)$



/* CPD3.DEF

   CPD3(z,w,a) calculates a vector with eigenvalues corresponding
   to a 3rd order system with the characteristic equation

        (s+a)(s*s+2zws+w*w)=0

   Inputs:  a = real pole
            z = relative damping of complex poles
            w = natural frequency of complex poles
   Output:  vector with desired poles

   Author: Anders Wallenborg
*/
cpd3(z,w,a) := matrix([-a],
                      [-z*w + %i*w*sqrt(1-z*z)],
                      [-z*w - %i*w*sqrt(1-z*z)])$
```

```
/* REGCP1.BAT

   MACSYMA batch file for calculating the coefficients of
   the characteristic polynomial of a state feedback/observer
   for the flexible servo with COLOCATED speed sensor (drive
   motor speed measured). Regulator design based on the
   simplified servo model.
   Author: Anders Wallenborg                                    */

/* load external functions */

batch("ackermann.def")$   /* pole placement (Ackermann's formula) */
batch("cpd3.def")$        /* desired closed loop poles */

/* simplified servo model */

a:matrix([ 0, 0,  k/j1],
         [ 0, 0, -k/j2],
         [-1, 1,   0  ])$
b:matrix([km*ki/j1], [0], [0])$
c:[kw1, 0, 0]$

/* control design */

p:cpd3(z,w,w)$
l:ackermann(a,b,p);
p:cpd3(z,alfa*w,alfa*w)$
k:ackermann(transpose(a),transpose(c),p)$
k:transpose(k);

/* controller char. polynomial */

areg:a-b.l-k.c$
cp:charpoly(areg,s)$
cp:ratsimp(cp,s)$
kill(k)$

array(coef,3)$
for i:0 thru 3 do coef[i]:ratcoeff(cp,s,3-i)$
a0:coef[0]$
for i:0 thru 3 do
   ( coef[i]:ratsimp(coef[i]/a0,w), display(coef[i]) );

/* substitute dimensionless quantity x = j2*w**2/k  */

a2:ratsubst(x*k/j2,w**2,coef[2])$
a2:ratsimp(a2,x);

a3:ratsimp(coef[3]/w,w)$
a3:ratsubst(x*k/j2,w**2,a3)$
a3:ratsimp(a3*w,x);

e:ratsimp(coef[1]*coef[2]-coef[3],w)$
e:ratsimp(e/(w**3),w)$
e:ratsubst(x*k/j2,w**2,e)$
e:ratsimp(e*w**3,x)$
```

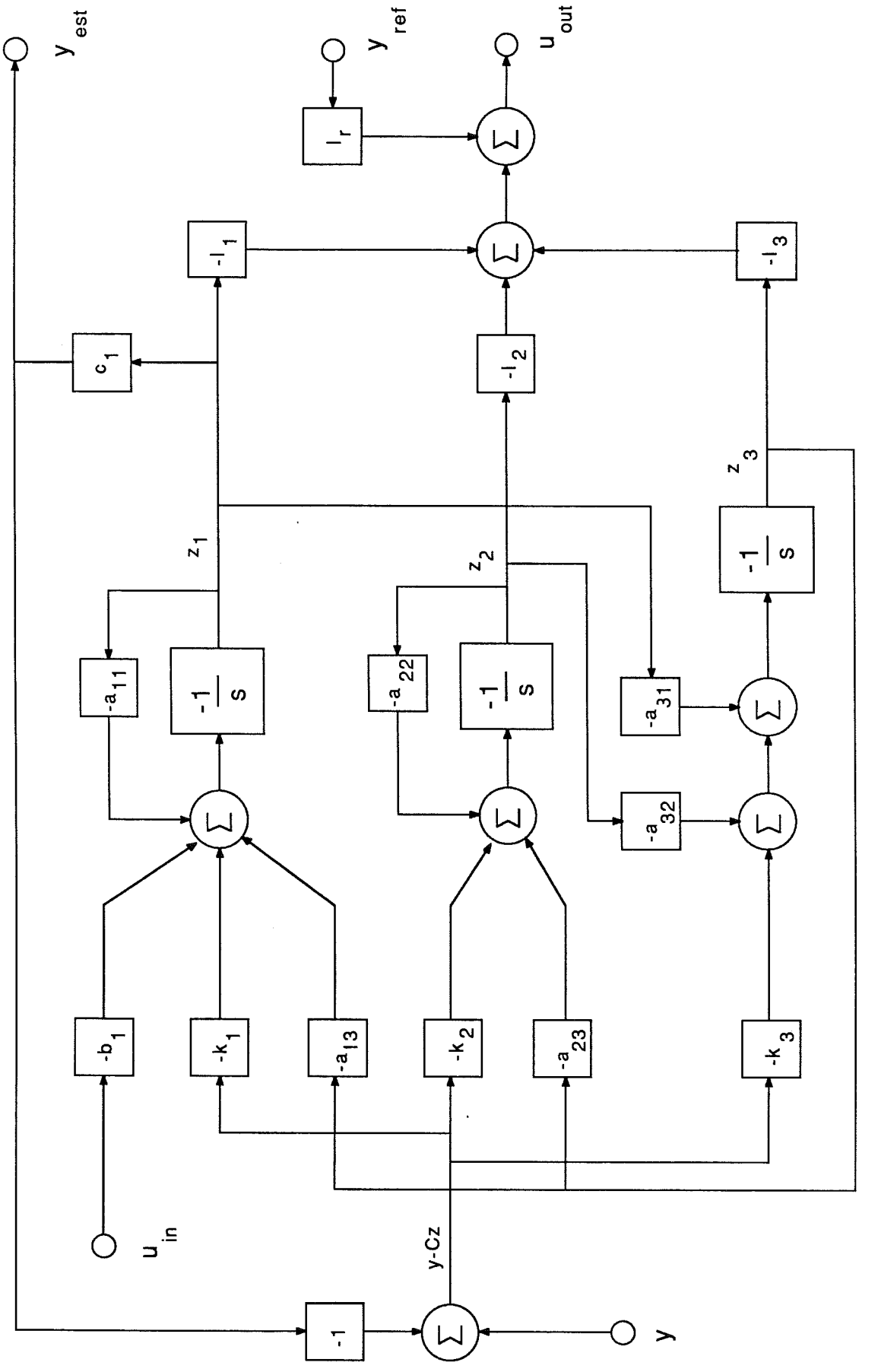
```
en:ratsimp(e*j2/(j1*w**3),x);
array(c,2)$
for i:0 thru 2 do
   ( c[i]:ratsimp(ratcoeff(en,x,2-i),z), display(c[i]) );

/* print results in log file */

writefile("regcp1.log");
for i:0 thru 3 do display(coef[i]);
a2;
a3;
for i:0 thru 2 do display(c[i]);
closefile();



/* REGCP2.BAT

   MACSYMA batch file for calculating the coefficients of
   the characteristic polynomial of a state feedback/observer
   for the flexible servo with NON-COLOCATED speed sensor.
   Regulator design based on the simplified servo model.
   Author: Anders Wallenborg                                    */

/* load external functions */

batch("ackermann.def")$   /* pole placement (Ackermann's formula) */
batch("cpd3.def")$        /* desired closed loop poles */

/* simplified servo model */

a:matrix([ 0, 0,  k/j1],
         [ 0, 0, -k/j2],
         [-1, 1,   0  ])$
b:matrix([km*ki/j1], [0], [0])$
c:[0, kw2, 0]$

/* control design */

p:cpd3(z,w,w)$
l:ackermann(a,b,p);
p:cpd3(z,alfa*w,alfa*w)$
k:ackermann(transpose(a),transpose(c),p)$
k:transpose(k);

/* controller char. polynomial */

areg:a-b.l-k.c$
cp:charpoly(areg,s)$
cp:ratsimp(cp,s)$
kill(k)$

array(coef,3)$
for i:0 thru 3 do coef[i]:ratcoeff(cp,s,3-i)$
a0:coef[0]$
for i:0 thru 3 do
```

```
   ( coef[i]:ratsimp(coef[i]/a0,w), display(coef[i]) );

/* substitute dimensionless quantity x = j2*w**2/k  */

a2:ratsubst(x*k/j2,w**2,coef[2])$
a2:ratsimp(a2,x);

a3:ratsimp(coef[3]/w,w)$
a3:ratsubst(x*k/j2,w**2,a3)$
a3:ratsimp(a3*w,x);

e:ratsimp(coef[1]*coef[2]-coef[3],w);

/* print results in log file */

writefile("regcp2.log");
for i:0 thru 3 do display(coef[i]);
a2;
a3;
e;
closefile();
```

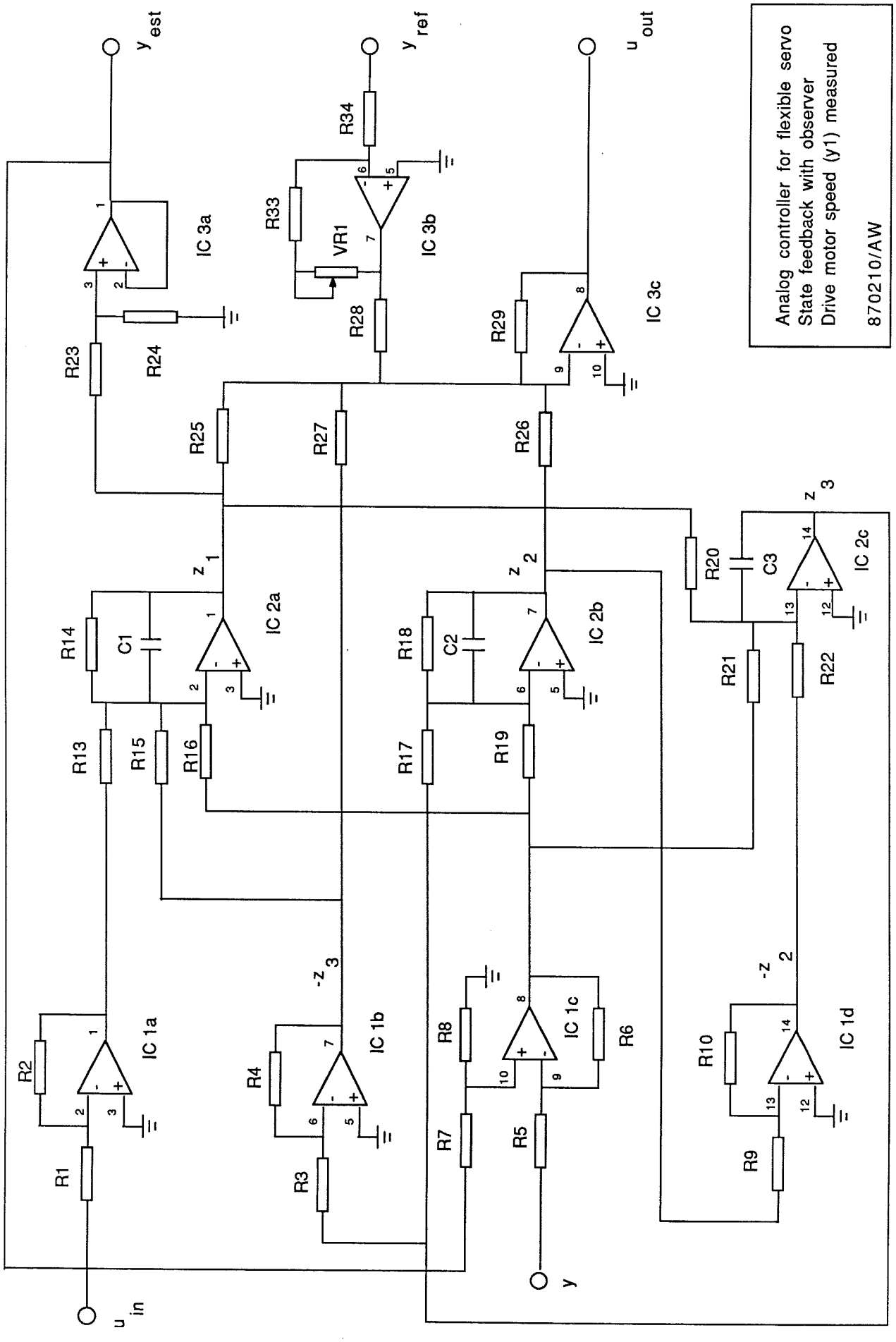
Appendix D

This appendix contains circuit diagrams and component lists for the analog controller and the friction comensation circuit.

Contents:



Analog linear controller for flexible servo - Block diagram


Analog controller for flexible servo
State feedback with observer
Drive motor speed (y1) measured
870210/AW
y est
y ref
u out
u in
y
IC 1a
IC 1b
IC 1c
IC 1d
IC 2a
IC 2b
IC 2c
IC 3a
IC 3b
IC 3c
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29
R33
R34
VR1
C1
C2
C3
z 1
z 2
z 3
-z 3
-z 2

Analog controller - Component list

Output: $y = 0.1 \cdot \omega_1$

Spring constant: $k = 2.4 \cdot 10^{-3}$

Nominal design: $\omega_{cl} = 12$, $\omega_{obs} = 18$

Resistors

R1	10k
R2	39k
R3	100k
R4	100k
R5	10k
R6	100k
R7	10k
R8	100k
R9	100k
R10	100k
R11	-
R12	-
R13	5k6 + 1k8
R14	470k
R15	39k
R16	47k
R17	120k
R18	1M
R19	22k
R20	47k
R21	82k
R22	47k
R23	s/c
R24	o/c
R25	39k
R26	15k
R27	100k
R28	10k
R29	10k
R30	-
R31	-
R32	-
R33	15k
R34	18k

Capacitors

C1	4.7 μF
C2	10 μF
C3	1.0 μF

Integrated Circuits

IC1	LM 324
IC2	LM 324
IC3	LM 324

Variable Resistors

VR1	10k

Analog controller - Component list

Output: $y = 0.1\omega_1$

Spring constant $k = 2.4 \cdot 10^{-3}$

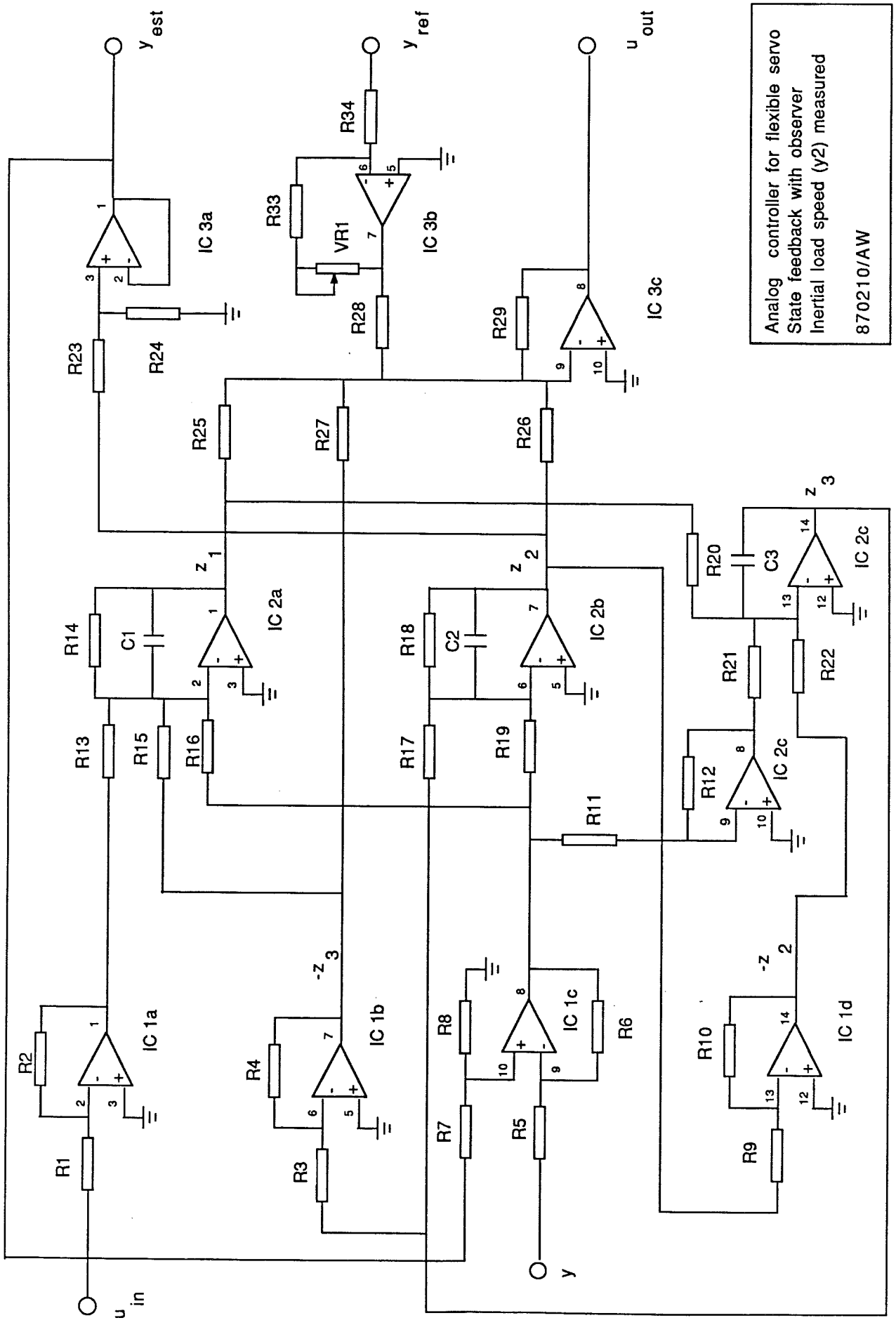
Decreased bandwidth design: $\omega_{cl} = 8$, $\omega_{obs} = 12$

Resistors	
R1	10k
R2	39k
R3	100k
R4	100k
R5	10k
R6	100k
R7	10k
R8	100k
R9	100k
R10	100k
R11	-
R12	-
R13	5k6 + 1k8
R14	470k
R15	39k
R16	82k
R17	120k
R18	1M
R19	82k
R20	47k
R21	270k
R22	47k
R23	s/c
R24	o/c
R25	56k
R26	100k
R27	820k
R28	33k
R29	10k
R30	-
R31	-
R32	-
R33	15k
R34	18k

Capacitors	
C1	4.7 μF
C2	10 μF
C3	1.0 μF

Integrated Circuits	
IC1	LM 324
IC2	LM 324
IC3	LM 324

Variable Resistors	
VR1	10k


Analog controller for flexible servo
State feedback with observer
Inertial load speed (y2) measured
870210/AW
y est
y ref
u out
u in
y
IC 1a
IC 1b
IC 1c
IC 1d
IC 2a
IC 2b
IC 2c
IC 2c
IC 3a
IC 3b
IC 3c
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29
R33
R34
VR1
C1
C2
C3
z 1
z 2
z 3
-z 3
-z 2

Analog controller - Component list

Output: $y = 0.1 \cdot \omega_2$ (non-colocated speed sensor)
Spring constant: $k = 2.4 \cdot 10^{-3}$
Nominal design: $\omega_{cl} = 12$, $\omega_{obs} = 18$

Resistors

R1	10k
R2	39k
R3	100k
R4	100k
R5	10k
R6	100k
R7	10k
R8	100k
R9	100k
R10	100k
R11	100K
R12	100K
R13	5k6 + 1k8
R14	470k
R15	39k
R16	39k
R17	120k
R18	1M
R19	22k
R20	47k
R21	12k
R22	47k
R23	s/c
R24	o/c
R25	39k
R26	15k
R27	100k
R28	10k
R29	10k
R30	-
R31	-
R32	-
R33	15k
R34	18k

Capacitors

C1	4.7 μF
C2	10 μF
C3	1.0 μF

Integrated Circuits

IC1	LM 324
IC2	LM 324
IC3	LM 324

Variable Resistors

VR1	10k

Department of Automatic Control Lund Institute of Technology	Flexible Servo Friction Compensation	870210 RB/AW fcomp.sch