CODEN: LUTFD2/(TFRT-7467)/1–15/(1990)

DSP Implementation of a Disk Drive Controller

Hermann Steingrímsson
Karl Johan Åström

Department of Automatic Control
Lund Institute of Technology
October 1990

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

Document name
INTERNAL REPORT

Date of issue
October 1990

Document Number
CODEN: LUTFD2/(TFRT-7467)/1-15/(1990)

Author(s)
Hermann Steingrímsson
Karl Johan Åström

Supervisor

Sponsoring organisation

Title and subtitle

DSP Implementation of a Disk Drive Controller

Abstract

This report describes design and implementation of a controller for a simple disk drive using DSP with fix point calculations.

Key words
State feedback; Observers; DSP fix point calculations; Scaling; Testing

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
15

Security classification

Recipient's notes

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

DSP Implementation of a Disk Drive Controller[†]

Hermann Steingrímsson
Graduate School of Business
University of Wisconsin
Madison, Wisconsin, USA

Karl Johan Åström
Department of Automatic Control
Lund Institute of Technology
Lund, Sweden

1. Introduction

The purpose of this paper is to study implementation of a controller based on state estimation and feedback from estimated states on a digital signal processor. Design of a control system for a disk drive is chosen as an example. The controller is implemented on a DSP that does not have floating point hardware. The control problem is described in Section 2, which also describes mathematical models of different complexity. Design of a controller is discussed in Section 3. This section contains a derivation of a continuous time controller and a discrete time controller. The continuous controller is used to choose design parameters and to estimate orders of magnitude. The discrete time controller is the algorithm implemented on the DSP. The section on control design also contains a discussion of design trade-offs. Implementation of the controller on a DSP is discussed in Section 4. Scaling of parameters and states is a major issue. An outline of the code is given. The complete code is listed in the appendix. Testing of the code is described in Section 5 and the paper ends with conclusions and references.

2. Disk Drive Control

Modern disk drive use fast voice coil actuators to position the magnetic heads on a track and to keep them on track under closed loop control. The task of the position control system is twofold: to position the heads over a desired track and to keep it there. The first task is a servo problem whereas the second task is a regulation problem. This paper treats the regulation problem.

Two methods are currently used for feedback measurements. In a *dedicated servo* an entire surface is used for position information, that could have been used for data. In an *embedded servo* the position information are embedded into the data track at the beginning of each sector, instead of using a separate surface. It is also possible to have dual layers so that the servo information is on a layer below the data layer.

The advantage of the dedicated servo is that position information available continuously. With a dedicated servo it is therefore possible to use a controller with a high bandwidth. In an embedded servo, position information is only obtained at a sector boundary. This limits the track following bandwidth and results in longer seek times, and more sluggish track following. A dedicated servo uses an extra surface for the position information. Thermal differences between the position surface and the data surfaces also give rise to errors.

Linear or rotary actuators with a permanent magnet and a voice coil are used to move the head across the tracks. The arm is ideally a rigid body which can be modeled as a double integrator. The large accelerations will, however, excite resonant modes. This makes it difficult to achieve a high bandwidth for positioning and track following.

Analog controllers have been used for servos. They contain amplifiers, compensation networks, notch filters, switches and passive components. The parameters of the analog components change with temperature and component aging can result in deteriorated performance of the servo.

There are several advantages in using a digital servo. Components having drift and aging are avoided, the number of components can be reduced and servo performance can be increased. a digital servo will, however, require high sampling rates. This makes a microcontroller less suitable. The inexpensive DSP's offer computational power an or-

[†] Part of this work was done when the second author was visiting professor and the first author a graduate student at the University of Texas at Austin.

der of magnitude greater than the microcontrollers and some, like the TMS320C14, do also have the hardware for input-output similar to a micro controller. Such components are ideally suited for implementation of fast servos of the type used in disk drives.

Position Detector

The head/track misalignment is the only information available to the controller. Control thus has to be based on error feedback. The position detector generates a voltage which is proportional to the misalignment of the head and track. The operating range is 23μm and the output voltage is in the range 0-5 V. After A/D-conversion one unit in the processor corresponds to a track/head misalignment of 11.5μm . The useful track width is approximately 4.3μm.

Control Signal

The D/A-converter generates a voltage in the range ±5 V. This voltage is amplified by an amplifier which generates a current. The current passes through the voice coil and generates a torque to move the arm.

Physical Constants of the Drive

The drive system has the following parameters:

Pivot to head radius

R:	0.08 m

Power amplifier gain

K_{pa}:	0.5 A/V

Torque constant of the actuator

K_t:	0.09 Nm/A

Total moment of inertia

J:	$50 \cdot 10^{-6}$ Kgm^2

Mathematical Model

A mathematical model describing the position of the arm as a function of the current trough the coil is a double integrator

$$J\frac{d^2\varphi}{dt^2} = K_t I \tag{1}$$

where φ is the angle of the arm. The transfer function from voltage u to arm position y is

$$G_p(s) = \frac{Y(s)}{U(s)} = \frac{K_p}{s^2} \tag{2}$$

where

$$K_p = K_{pa}K_tR/J \approx 72\text{m/s}^2\text{V}$$

The model given by Equation (1) neglects the fact that the arm has compliance. If this is considered, the plant transfer function becomes $G_{p1} = G_pG_1$, where

$$G_1 = \frac{\omega_1^2}{s^2 + 2\zeta\omega_1 s + \omega_1^2} \tag{3}$$

or $G_{p2} = G_{p1}G_2 = G_pG_1G_2$, where

$$G_2(s) = \frac{s^2 + 2\zeta_3\omega_3 s + \omega_3^2}{s^2 + 2\zeta_1\omega_2 s + \omega_2^2} \tag{4}$$

Typical values of ω and ω_1 are 2 KHz and 3 KHz. The model given by Equation (1) is a good approximation of low frequencies. Because of the resonances this model does, however, not describe the system well at frequencies approaching one kHz. For those frequencies it is necessary to use models like (3) and (4) or even more complicated models.

Disturbances

The major disturbances acting on the system are low frequency load disturbance and a periodic tracking error. Load disturbances are due to the torque from the wires connected to the arm. This torque is almost constant at a given track, but it changes with the track. It may also change with temperature. The second disturbance is due to the eccentricity of the disk which translates into a periodical tracking error. Since the amplitude of this error is small, the disturbance can be approximated by a sinusoid with the rotational frequency of the disk. By introducing the state x_3, the load disturbance can be added to equation (1), giving

$$\begin{aligned}\frac{d^2\varphi}{dt^2} &= K_pu + x_3\\ \frac{dx_3}{dt} &= 0\end{aligned} \tag{5}$$

3. Controller Design

Control algorithms for the disk drive will be derived in this section. A continuous time controller for the simple rigid body model is first derived.

This derivation gives insight into the control problem and guide lines for choosing the design parameters. The controller is obtained using a straightforward pole-placement method. See [Åström and Wittenmark, 1990]. A discrete time algorithm is then derived. This algorithm is the basis for the DSP implementation.

A state-space model of (5) is

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{6}$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ K_p \\ 0 \end{pmatrix}, \quad y = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

and
x_1: position [m]
x_2: velocity [m/s]
x_3: torque [Nm]
K_p: gain [m/s^{2}V]
u: control signal [V]

Continuous-Time Controller

It is easily verified that the states x_1 and x_2 of the model (6) are controllable. The disturbance state x_3 is naturally not controllable. All the states of the system are observable. A controller based on a state-feedback and an observer can therefore be designed.

State Feedback. The controller will now be derived in the straightforward manner. See [Åström and Wittenmark, 1990]. It is first assumed that all states are measurable. The state feedback

$$u = -Lx = -l_1x_1 - l_2x_2 - l_3x_3 \tag{7}$$

gives the closed-loop system

$$\dot{x}(t) = (A - BL)x(t) \tag{8}$$

The gains l_1 and l_2 are selected such that the characteristic polynomial of the closed loop system becomes

$$s(s^2 + 2\zeta_p\omega_p s + \omega_p^2) \tag{9}$$

Notice that the zero at the origin is due to the uncontrollable disturbance mode. The characteristic polynomial of (8) is

$$s\big(s^2 + K_pl_2s + K_pl_1\big)$$

To obtain (9) the feedback gains l_1 and l_2 should thus be chosen as

$$\begin{aligned} l_1 &= \omega_p^2/K_p \\ l_2 &= 2\zeta_p\omega_p/K_p \end{aligned} \tag{10}$$

The gain l_3 is chosen to give perfect disturbance cancellation, i.e.

$$l_3 = 1/K_p \tag{11}$$

The control law (7) can be interpreted as a feedback from the process states x_1 and x_2 and a feedforward from the disturbance state x_3.

State Observer. A state observer is given by

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K\big(y(t) - C\hat{x}(t)\big) \tag{12}$$

where $\hat{x}$ is the estimate of the state vector x. The reconstruction error $\tilde{x} = x - \hat{x}$ is given by

$$\dot{\tilde{x}}(t) = (A - KC)\tilde{x}(t) \tag{13}$$

The characteristic polynomial of this system is

$$s^3 + k_1s^2 + k_2s + k_3$$

The observer gains k_1, k_2 and k_3 are chosen so that the observer has the characteristic polynomial

$$(s + a_o)(s^2 + 2\zeta_o\omega_o s + \omega_o^2) \tag{14}$$

The following observer gains are then obtained

$$\begin{aligned} k_1 &= 2\zeta_o\omega_o + a_o \\ k_2 &= \omega_o^2 + 2\zeta_o\omega_o a_o \\ k_3 &= \omega_o^2 a_o \end{aligned} \tag{15}$$

Discrete-Time Controller

To derive a discrete time controller the system (6) is sampled. This gives

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) \end{aligned} \tag{16}$$

where

$$\Phi = \begin{pmatrix} 1 & h & h^2/2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} K_ph^2/2 \\ K_ph \\ 0 \end{pmatrix} \tag{17}$$
$$C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

and h is the sampling interval. The states x_1 and x_2 of the discrete time system (17) are controllable but disturbance state x_3 is of course uncontrollable. All states are observable.

First consider the case when all states are measured. With state feedback the closed loop system has the characteristic polynomial

$$(z-1)\left(z^2+\left(K_p\frac{h^2}{2}l_1+K_phl_2-2\right)z + 1-K_phl_2+K_p\frac{h^2}{2}l_1\right) \tag{18}$$

Notice that the pole $z=1$ is due to the uncontrollable disturbance mode. The desired closed loop characteristic polynomial is obtained by sampling (9). This gives

$$(z-1)\left(z^2+a_{p1}z+a_{p2}\right)$$

where

$$\begin{aligned} a_{p1} &= -2e^{-\zeta_p\omega_p h}cos\left(\omega_p h\sqrt{1-\zeta_p^2}\right)\\ a_{p2} &= e^{-2\zeta_p\omega_p h}\end{aligned} \tag{19}$$

Choosing the feedback gains l_1 and l_2 so that (19) and (18) are the same gives

$$\begin{aligned} l_1 &= \frac{a_{p1}+a_{p2}+1}{K_ph^2}\\ l_2 &= \frac{a_{p1}-a_{p2}+3}{2K_ph}\end{aligned} \tag{20}$$

State Observer. A state-observer of the form

$$\begin{aligned} \hat{x}(k|k) &= \hat{x}(k|k-1)+K\big(y(k)-\hat{y}(k|k-1)\big)\\ \hat{x}(k+1|k) &= \Phi\hat{x}(k|k)+\Gamma u(k)\\ \hat{y}(k+1|k) &= C\hat{x}(k+1|k)\end{aligned} \tag{21}$$

is chosen. The reconstruction error is then given by

$$\tilde{x}(k+1|k)=\Phi(I-KC)\tilde{x}(k|k-1) \tag{22}$$

This system has the characteristic polynomial

$$z^3+\left(k_1+hk_2+\frac{h^2}{2}k_3-3\right)z^2 + \left(3-2k_1-hk_2+\frac{h^2}{2}k_3\right)z+k_1-1$$

Requiring that this polynomial be equal to

$$(z-a_{o3})(z^2+a_{o1}z+a_{o2})$$

where a_{o1} and a_{o2} are given by equation (19) and

$$a_{o3}=e^{-a_o h} \tag{23}$$

gives

$$\begin{aligned} k_1 &= 1-a_{o2}a_{o3}\\ k_2 &= \frac{a_{o1}-a_{o2}-a_{o3}+a_{o1}a_{o3}+3a_{o2}a_{o3}+3}{2h}\\ k_3 &= \frac{a_{o1}+a_{o2}-a_{o3}-a_{o1}a_{o3}-a_{o2}a_{o3}+1}{h^2}\end{aligned} \tag{24}$$

The Control Algorithm

Reorganizing the calculations to minimize the delay between the A/D- and D/A-conversions gives the following algorithm.

ALGORITHM 1

```
1. Read       $y(k)$
2. Compute    $\hat{x}(k|k)=\hat{x}(k|k-1)+Ke(t)$
              $e(t)=y(k)-\hat{y}(k|k-1)$
              $v(k)=-L\hat{x}(k|k)$
              $u(k)=f(v(k))$
3. Output     $u(k)$
4. Update     $\hat{x}(k+1|k)=\Phi\hat{x}(k|k)+\Gamma u(k)$
              $\hat{y}(k+1|k)=C\hat{x}(k+1|k)$
5. Wait
```

where the function f is a model of the actuator nonlinearity. □

Notice that the algorithm has been organized so that the computational delay between the A/D and D/A converters are minimized. Notice also that Step 2 of this algorithm can be expressed as

$$\begin{aligned} \hat{x}(k+1|k) &= \Phi_n\hat{x}(k|k-1)+\Gamma_n y(k)\\ v(k) &= C_n\hat{x}(k|k-1)+D_n y(k)\end{aligned} \tag{25}$$

where

$$\begin{aligned} \Phi_n &= \Phi-\Phi KC-\Gamma L+\Gamma LKC\\ \Gamma_n &= \Phi K-\Gamma LK\\ C_n &= -L+LKC\\ D_n &= -LK\end{aligned}$$

Sampling Frequency and Anti-Aliasing Filter

The following rule of thumb for the selection of sampling frequency for a digital controller with a zero-order hold, is given by [Åström and Wittenmark, 1990].

$$0.2 \leq \omega_c h \leq 0.6 \tag{26}$$

where ω_c is the crossover frequency. With a sampling frequency of 20 KHz the crossover frequency can be at least 1 kHz. This was judged adequate for the application.

A prefilter in the form of a second order Bessel filter with the bandwidth 7500Hz was chosen to avoid aliasing.

Design Parameters

The controller has the design parameters: ω_p, ζ_p, ω_o, ζ_o, a_o and h that must be chosen. The choice of sampling interval has already been discussed. Parameters ζ_p and ζ_o, which represent relative damping, can easily be chosen. Then there remain three parameters ω_p, ω_o and a_o. Requirements on desired settling time and disturbance rejection have to be matched against constraints due to model uncertainty. Recall that the rigid body model used for the design was not valid for frequencies approaching 1 kHz. After some experimentation the following design parameters were chosen for the nominal case.

$$\omega_p = 1000\pi$$
$$\zeta_p = 0.80$$
$$\omega_o = 1500\pi$$
$$\zeta_o = 0.80$$
$$a_o = 200\pi$$

Figure 1 shows a simulation of the response of the system with the nominal design parameters. In the simulation a step command at 11.5 μm is first applied. After 3 ms a torque disturbance in the form of a step of 0.013 Nm is applied.

The simulation was performed assuming that the plant model is given by Equation (3), which has a resonance of 2 kHz. The settling time is about 1.5 ms and the resonant modes are not much excited by the command signal. With the rigid body process model given by Equation (2) the system has an amplitude margin of 3.2 and phase margin of 31°. The gain cross-over frequency is 460 Hz and the phase cross-over is 1036 Hz. This indicates that the design based on the rigid body model has acceptable margins.

The effects of the neglected dynamics on the margins can be estimated as follows. Assuming

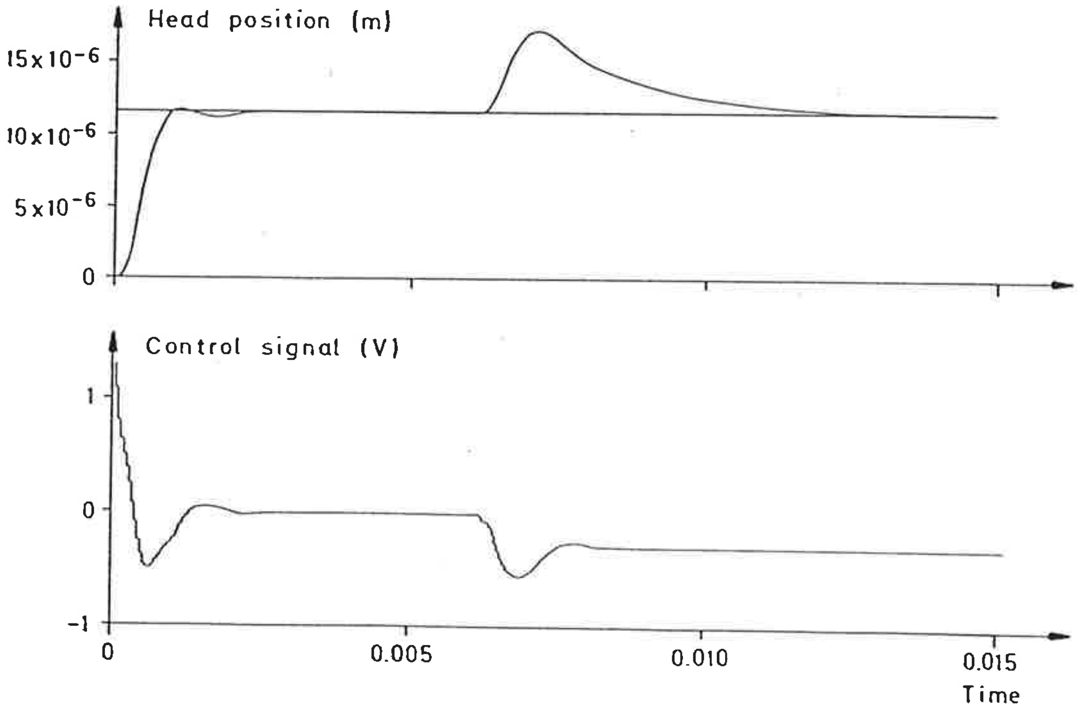

Figure 1. Step response of the closed loop system.

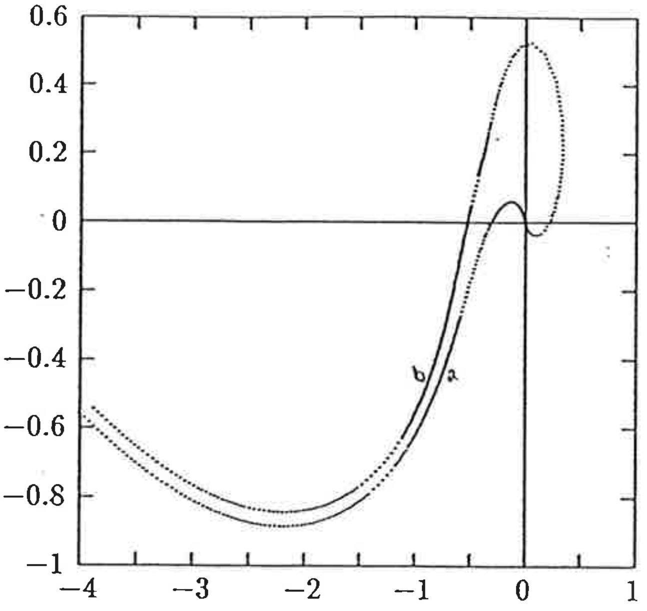

Figure 2. Nyquist curves for the loop transfer functions with the rigid body dynamics, Equation (2), and the dynamics with one resonant mode, Equations (2) and (3).

that the system dynamics is described by the model having one resonant mode, Equation (3). The additional dynamics is then given by

$$G_1(s) = \frac{\omega_1^2}{s^2 + 2\zeta\omega_1 s + \omega_1^2} \tag{27}$$

where ω_1 is the undamped natural frequency (2 KHz) and ζ is the relative damping (0.1). The magnitude M of the transfer function G_1 at ω is

$$M = \frac{1}{\sqrt{(1-\omega^2/\omega_1^2)^2 + (2\zeta\omega/\omega_1)^2}} \tag{28}$$

Introducing $\omega = \omega_{\phi c} = 1036$ Hz this equation gives $M = 1.36$. The gain margin is thus decreased to 1.77. The argument of the transfer function of ω is

$$\alpha = -\arctan\frac{2\zeta\omega/\omega_1}{1-\omega^2/\omega_1^2} \tag{29}$$

with $\omega_{gc} = 460$ Hz, which gives 2.8°. Figure 2 shows the Nyquist curves with the nominal process transfer function (2) and the transfer function with one resonant mode (3). These curves show that the essential effects of the resonant mode is to decrease the amplitude margin.

An additional illustration of the sensitivity to gain variations is illustrated in the simulation in Figure 3, which shows the time response of the closed loop systems, where the loop gain changes with ±20%. Compare with the nominal case in Figure 1.

Tracking Error

Misalignment errors is a common source of tracking errors. Such disturbances can be approximated by a sinusoidal. The sensitivity of the closed loop system to such errors can be modeled by the pulse transfer function.

$$H_{track}(z) = \frac{1}{1 - H(z)G(z)} \tag{30}$$

With the chosen controller we find that disturbances of 60 Hz are attenuated by a factor of 32. This agrees well with the simulation results that showed a reduction from 5 μm to 0.2 μm.

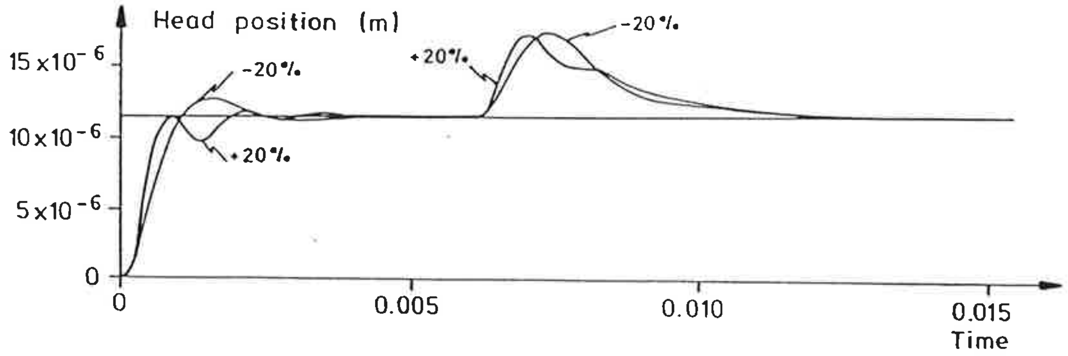

Figure 3. Responses of the closed loop system to a step command and a step change in the torque when the process gain changes by ±20%.

4. DSP Implementation

Implementation of the controller using a DSP with fixed point calculations will now be discussed. The key issues are scaling of coefficients and states. See [Roberts and Mullins, 1987], [Hanselmann, 1987], [Texas Instruments, 1986], [Texas Instruments, 1988a], [Texas Instruments, 1988b], [Texas Instruments, 1989a], [Texas Instruments, 1989b], [Texas Instruments, 1990a] and [Texas Instruments, 1990b].

The controller derived can be described by the matrices:

$$\begin{aligned}
\Phi &= \begin{pmatrix} 1 & 5\cdot 10^{-5} & 1.25\cdot 10^{-9} \\ 0 & 1 & 5\cdot 10^{-5} \\ 0 & 0 & 1 \end{pmatrix} \\
\Gamma &= \begin{pmatrix} 9.25\cdot 10^{-8} & 3.7\cdot 10^{-3} & 0 \end{pmatrix}^T \\
C &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\
K &= \begin{pmatrix} 3.352917424019266\cdot 10^{-1} \\ 1.100808656418762\cdot 10^{3} \\ 5.695461161564441\cdot 10^{5} \end{pmatrix} \\
L &= \begin{pmatrix} 1.176909751519137\cdot 10^{5} \\ 6.300506379182784\cdot 10^{1} \\ 1.351351351351351\cdot 10^{-2} \end{pmatrix}^T
\end{aligned} \tag{31}$$

The elements of these matrices have numbers that are widely spread. To accommodate this on a DSP with fix point arithmetic it is necessary to scale the numbers appropriately.

I/O-Scaling

The range of the output signal in tracking mode corresponds to ±11.5μm. The scaling will be chosen so that this corresponds to ±1 units in the DSP. The input scaling factor s_y is therefore

$$s_y = 11.5\ \mu\text{m}$$

Since the dimensions of l_1, l_2 and l_3 are [v/m], [sv/m] and [s^{2}v/m] respectively, it is advantageous to multiply L with s_y rather than dividing C with s_y. The output must also be scaled since the D/A-converter converts ±1 into ±5 V. The matrix L is thus multiplied by the output scaling factor

$$s_u = \frac{2}{10\ \text{V}} = 0.2\ \text{V}^{-1}$$

The vector Γ is in the same way as L. Hence

$$\begin{pmatrix} L & \Gamma \end{pmatrix} \Leftrightarrow \begin{pmatrix} s_y s_u L & \dfrac{1}{s_y s_u}\Gamma \end{pmatrix}$$

and the scaled vectors Γ and L become

$$\Gamma = \begin{pmatrix} 4.021739130434783\cdot 10^{-2} \\ 1.608695652173913\cdot 10^{3} \\ 0 \end{pmatrix} \tag{32}$$

$$L = \begin{pmatrix} 2.706892428494014\cdot 10^{-1} \\ 1.449116467212040\cdot 10^{-4} \\ 3.108108108108108\cdot 10^{-8} \end{pmatrix}^T \tag{33}$$

Coefficient Scaling

The coefficients of system (31), and (32) and (33) can not be represented in the DSP. A similarity transform $\hat{x} \Leftrightarrow T_c\hat{x}$ is used to scale the coefficients. This gives

$$\begin{aligned} &\begin{pmatrix} \Phi & \Gamma & C & K & L \end{pmatrix} \Leftrightarrow \\ &\begin{pmatrix} T_c\Phi T_c^{-1} & T_c\Gamma & CT_c^{-1} & T_cK & LT_c^{-1} \end{pmatrix} \end{aligned} \tag{34}$$

The elements of the matrices Φ, K, Γ and L are proportional to powers of h. It is therefore natural to use a scaling matrix of the form

$$T_c = \operatorname{diag}\begin{pmatrix} cs_1 & cs_2 & cs_3 \end{pmatrix}$$

The following scaling matrix was obtained after some trial and error

$$T_c = \operatorname{diag}\begin{pmatrix} \frac{1}{1.05k_1} & \frac{1}{5k_2} & \frac{1}{40k_3} \end{pmatrix} \tag{35}$$

State-vector scaling

With the chosen scaling of all controller coefficients have magnitudes less than one. It now remains to scale the state-vector. Simulations showed that overflow could occur when the head is positioned at the edge of the track and the disk controller is switched to track-following. The scale factors s_1, s_2 and s_3 were chosen from a simulation of this case. It was found that x_1 had to be scaled down and that x_2 could be scaled up. Scaling of x_3 depends on the maximum possible load disturbance. For a load disturbance of 0.3 Nm it was not necessary to scale

x_3. The following transformation was therefore chosen to scale the state vector

$$T_{sc} = \mathrm{diag}\Big(1/2.8 \quad 1/0.3 \quad 1\Big) \qquad (36)$$

The following controller matrices were obtained after scaling:

$$\Phi = \begin{pmatrix} 1 & \phi_{12} & \phi_{13} \\ 0 & 1 & \phi_{23} \\ 0 & 0 & 1 \end{pmatrix}$$
$$\Gamma = \begin{pmatrix} 4.079845420409798 \cdot 10^{-2} \\ 9.742508490121855 \cdot 10^{-1} \\ 0 \end{pmatrix}$$
$$C = \begin{pmatrix} 9.857577226616642 \cdot 10^{-1} & 0 & 0 \end{pmatrix} \qquad (37)$$
$$K = \begin{pmatrix} 3.401360544217687 \cdot 10^{-1} \\ 6.666666666666667 \cdot 10^{-1} \\ 2.5 \cdot 10^{-2} \end{pmatrix}$$
$$L = \begin{pmatrix} 2.668340115802361 \cdot 10^{-1} \\ 2.392799926898983 \cdot 10^{-1} \\ 7.080843606269305 \cdot 10^{-1} \end{pmatrix}^T$$

where

$$\phi_{12} = 8.375348965918672 \cdot 10^{-2}$$
$$\phi_{13} = 2.888874735967583 \cdot 10^{-2}$$
$$\phi_{23} = 6.898517895130376 \cdot 10^{-1}$$

The system matrices are finally transformed to integers to fit the 16 bit fractional format of the DSP. The transformation is done by multiplying the coefficients with 2^{15} and rounding each coefficient to the nearest integer. The matrices then become

$$\Gamma = \begin{pmatrix} 1337 \\ 31924 \\ 0 \end{pmatrix}^T$$
$$\Phi = \begin{pmatrix} 1 & 2744 & 947 \\ 0 & 1 & 22605 \\ 0 & 0 & 1 \end{pmatrix}$$
$$C = \begin{pmatrix} 32301 & 0 & 0 \end{pmatrix} \qquad (38)$$
$$K = \begin{pmatrix} 11146 \\ 21845 \\ 819 \end{pmatrix}$$
$$L = \begin{pmatrix} 8744 & 7841 & 23203 \end{pmatrix}^T$$

The largest roundoff error 0.04% occurs in ϕ_{13}. To find how the poles of the controller are affected by the coefficient rounding, the characteristic equation of the controller was calculated. The largest pole deviation is 0.0013% from the design value.

5. The DSP Code

The control algorithm was implemented on the TMS320C25 by using the Texas Instruments Software Development System. The complete code is listed in Appendix A. The organization of the code is straightforward. It is composed of the following steps:

1. Perform A/D conversion.
2. Compute the state estimate.
3. Compute the new control signal.
4. Saturate control signal.
5. Perform D/A conversion.
6. Update equations for state estimate.

Compare with Algorithm 1. Approximately 32% of the computational power of the TMS320C25 used when the controller was running.

It can be estimated how processor loading increases with the order of the controller. Neglecting saturation arithmetic and anti-windup calculations, the number of multiply/accumulate instructions are proportional to $n^2 + 5n$ where n is the order of the controller. A 6th order controller would therefore exhaust the computational power of the C25. The saturation arithmetic routine must be

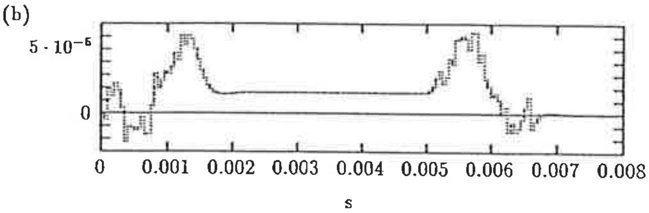

Figure 4. Impulse response of the controller (a) and the error compared to an ideal implementation (b).

called approximately 2n times. In our case, the saturation arithmetic consumes almost 50% of the total execution time. Therefore, if saturation arithmetic can be avoided by using more careful scaling, one can estimate that a 10th order controller can be implemented on the C25.

6. Testing

The open loop behavior of the controller was tested using the development system. The impulse response of the controller was generated and compared to the ideal impulse response. Figure 4(a) shows the responses of the controller to two impulses of magnitude 0.9 and −0.9. Figure 4(b) shows the error between the ideal and actual impulse response of the controller. The small error small is due to the roundoff in the controller. Notice that the quantization step is approximately $3 \cdot 10^{-5}$.

The observer was tested separately. A control signal was generated and the corresponding ideal response of the arm was calculated. The input signal was piecewise constant with jumps at $t = 0$, 0.001, 0.0018, and 0.0021. A load disturbance that was unknown to the observer was added at time $t = 0.0025$. All signals were scaled appropriately and fed to the observer whose response was recorded. Figure 5 shows the velocity estimate and its error. Figure 6 shows the position estimate and its error. The error is very small before time $t = 0.0025$, where the load disturbance was introduced. The load disturbance does, however, introduce significant errors both in velocity and position estimates. This is natural, because the observer does not have information about this load disturbance. The error will, however, decrease when the observer improves its estimate of the disturbance as is indicated in Figure 6.

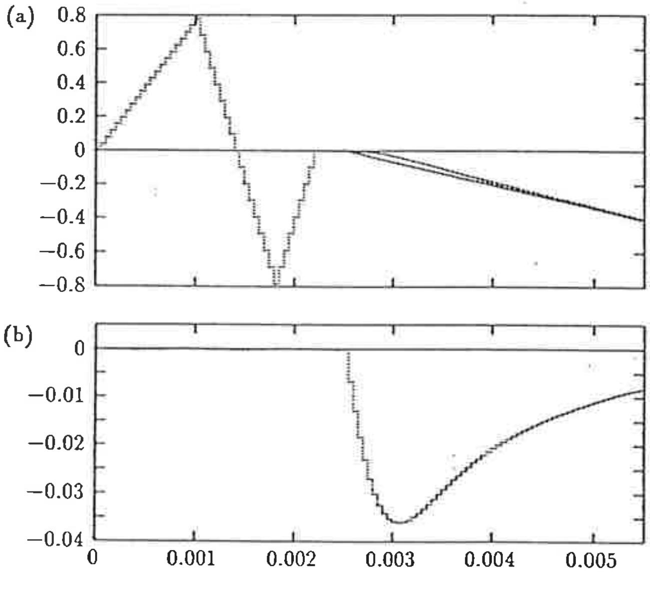

Figure 5. Actual and estimated velocity (a) and estimation error (b).

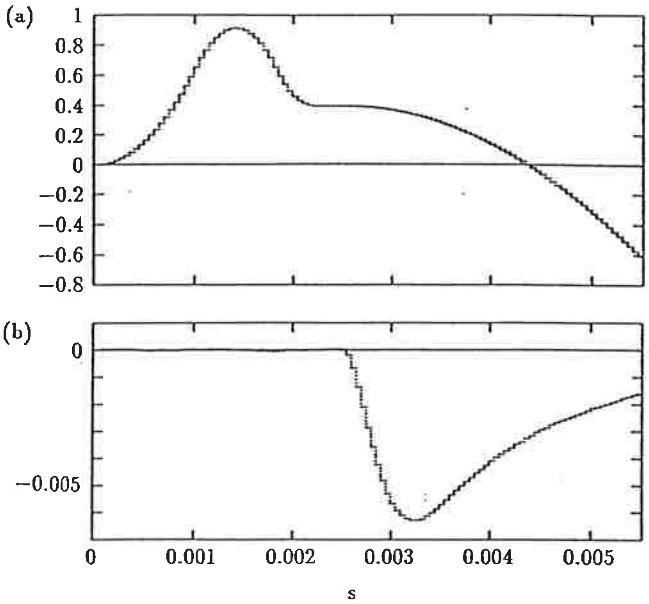

Figure 6. Actual and estimated position (a) and estimation error (b).

Although open loop testing can never replace actual closed loop testing of the whole system, these results indicate that the controller works properly.

Remarks on a Roundoff Algorithm

The first tests of the algorithm used a roundoff scheme found in a programming example in [Texas Instruments, 1986]. This resulted in a large estimation error, see Figure 7. The problem was investigated, since the error was larger than estimates based on analysis of roundoff errors. The reason for this is an error in the roundoff algorithm. To reduce quantization errors the numbers are rounded off, rather than truncated, before they are stored as 16-bit numbers. This roundoff is done in software. To roundoff a positive number, a bit is added to the MSB of the lower half of the 32-bit number before it is stored away. At first sight it appears natural to subtract the bit from the number to roundoff a negative number. This was done in the coding example [Texas Instruments, 1986]. This is not correct

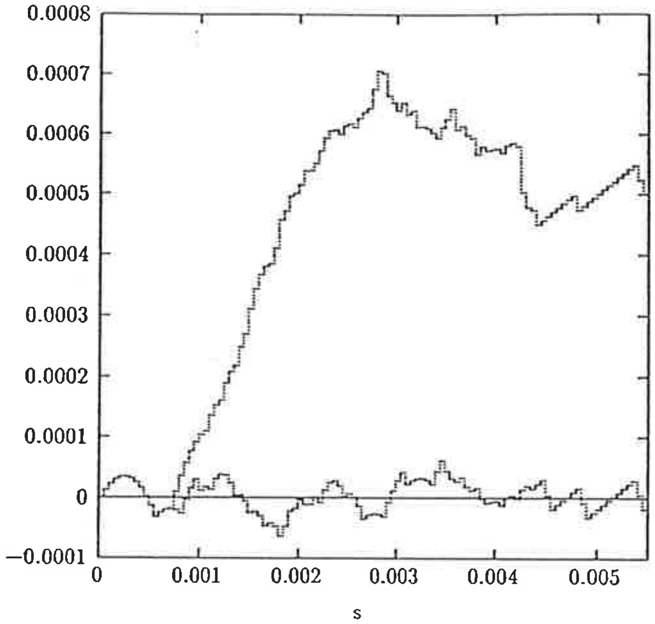

Figure 7. Position error with an incorrect roundoff algorithm.

with the chosen number representation. The roundoff algorithm gives –2 when applied to the number –1 because of the computational scheme used in the DSP. If the upper half of the number is complemented without considering the lower half, the result will not be the same as if the whole number is complemented. The correct code for the roundoff is given in Appendix A.

7. Conclusions

This paper shows that it is straightforward to implement a controller based on an observer and feedback from the observed states using a DSP with fix point calculations. Some effort is required to obtain proper scaling. The coefficient scaling is quite straightforward and can be automated. Scaling of the states is more difficult. It requires that the ranges of the states are known. This can be determined from simulation. Great care has to be exercised to find the worst cases. The code for the disk controller is much simpler than the code for the PID-controller discussed in [Åström and Steingrímsson, 1991]. The reason is that the disk controller is designed for a specific process while the PID-controller is designed as a general purpose controller. The coefficient ranges for the PID-controller are therefore much wider. This requires more complex scaling and saturation arithmetic, which is a large part of the code [Åström and Steingrímsson, 1990].

8. References

Åström, K. J. and H. Steingrímsson (1990): "Implementation of a PID controller on a DSP," Texas Instruments.

Åström, K. J., and B. Wittenmark (1990): *Computer Controlled Systems – Theory and Design*, Second edition, Prentice-Hall, Englewood Cliffs, NJ.

Dote, Y. (1990): *Servo Motor Control Using Digital Signal Processor*, Prentice Hall, Texas Instruments.

Hanselmann, H. (1987): "Implementation of digital controllers – A survey," *Automatica*, **23**.

Roberts, R. A., and C. T. Mullins (1987): *Digital Signal Processing*, Addison-Wesley Publ Co.

Texas Instruments (1986): *Digital Signal Processing Applications with the TMS320 Family – Theory, Algorithms, and Implementations*, Digital Signal Processing, Semiconductor Group.

Texas Instruments (1988a): *First-Generation TMS320 – User's Guide*, TI Digital Signal Processing, Prentice Hall.

Texas Instruments (1988b): *Second-Generation TMS320 – User's Guide*, TI Digital Signal Processing, Prentice Hall.

Texas Instruments (1989a): *TMS320C1x / TMS320C2x – User's Guide*, Digital Signal Processor Products.

Texas Instruments (1989b): *TMS320 Family Development Support – Reference Guide*, Digital Signal Processor Products.

Texas Instruments (1990a): *Digital Signal Processing – Applications with the TMS320 Family*, Application book volume 3, Digital Signal Processor Products.

Texas Instruments (1990b): *TMS320C3x – User's Guide*, Digital Signal Processor Products.

Appendix A: Disk Controller for TMS320C25

```
; Disk Controller for the TMS320C25
; Based on a Rigid Body Model of the Arm
; Version 1.0
; Author: Hermann Steingrimsson
; Date: 3-31-1990
;
;
; RESERVE SPACE IN DATA MEMORY FOR CONSTANTS AND VARIABLES
DTbeg .bss A12,1 ;The matrix A (or Phi)
.bss A13,1
.bss A23,1
.bss B1,1 ;The vector B (or Gamma)
.bss B2,1
.bss C1,1 ;C1
.bss K1,1 ;The vector K (in this case CK/2)
.bss  K2,1
.bss K3,1
.bss L1,1 ;The vector L
.bss L2,1
.bss L3,1
.bss    MAXNUM,1    ;Maximum number
.bss MINNUM,1    ;Minimum number
.bss UMAX,1 ;Saturation limits
.bss UMIN,1
.bss ONE,1 ;ONE=1
.bss MODE,1
DTend .bss CLOCK,1 ;End of parameters in data memory
.bss XE1,1 ;State vector x(k+1|k)
.bss XE2,1
.bss XK1,1     ;Vector x(k|k)
.bss    XK2,1
        .bss XK3,1
.bss YE,1 ;Estimate of ye
.bss Y,1 ;Input
.bss ERR,1
.bss V,1 ;Control signal before saturation
.bss U,1 ;Control signal after saturation U=SAT(V)

;Begin program memory

.sect "IRUPTS"
B START ;Branch to start of program

;Store parameters in program memory

.data
Ptable .set $
.word 2744,947,22608,1337,31924,32301,11146,21845,819,8744,7841,23203
```

```
.word   32767,-32768,32766,-32766,1,1,1
.word -1
Ptend .set $-1
SCALE .set 15

;Initialize

.text
START DINT ;Disable interupts
NOP
SOVM ;Set overflow mode
SSXM ;Set sign-extension mode
SPM   0 ;No shifting from P register

;Load coeff from prog. mem to data mem. use BLKP

LRLK  AR0,DTbeg  ;AR0 points to begining of data block
LARP  AR0
RPTK  Ptend-Ptable ;Set up counter
BLKP  Ptable,*+ ;Move data
;=> Coeff loaded into data memory

;Initialize variables

LDPK  A12 ;Point to correct data page
ZAC ;Clear variables
SACL  XE1
SACL  XE2
SACL  XK3
SACL  YE
SACL  U

OUT  MODE,PA4 ;Init analog board
OUT  CLOCK,PA5
LARP 0 ;Point to AR0

;Begin loop

;WAIT BIOZ  GET ;Wait for input
; B     WAIT
WAIT IN     Y,PA0 ;Change WAIT to GET when ; are removed


ZALH  Y ;Form ERR = y(k) - ye(k|k-1)
SUBH  YE
SACH  ERR

;Compute x(k|k) = x(k|k-1) + K*err
```

```
LAC    XE1,SCALE ;Calculate x1(k|k)
LT     ERR
MPY    K1
APAC
LRLK   AR0,XK1
CALL   ROUOF

LAC    XE2,SCALE ;Calculate x2(k|k)
LT     ERR
MPY    K2
APAC
LRLK   AR0,XK2
CALL   ROUOF

LAC    XK3,SCALE ;Calculate x3(k|k) (Estimate xe3 not needed)
LT     ERR
MPY    K3
         APAC
LRLK   AR0,XK3
CALL   ROUOF

;Calculate control signal u(k) = -Lx(k|k)

ZAC
LT     XK1
MPY    L1

LTS    XK2
MPY    L2

LTS    XK3
MPY    L3
SPAC

LRLK   AR0,V
CALL   ROUOF

;Saturation function (12 instr. cycles)

ZALH   V
SUBH   UMIN
BLZ    LOWER1 ;Branch if v < umin
ZALH   V
SUBH   UMAX
BLZ    SAME ;Branch if v < umax
ZALH   UMAX ;v >= umax
SACH   U ;u = umax
B      FIN ;Begining of loop
```

```
LOWER1 ZALH  UMIN
SACH  U ;u = umin
NOP ;Always same time
NOP
NOP
NOP
B     FIN

SAME ZALH  V
SACH  U ;u = v
NOP
NOP

FIN OUT   U,PA2 ;Output control signal

;Update the estimate xe(k+1|k) = Ax(k|k) + Bu(k)
;                    ye(k+1|k) = Cxe(k+1|k)

LAC   XK1,SCALE ;Calculate xe1

LT    XK2
MPY   A12

LTA   XK3
MPY   A13

LTA   U
MPY   B1
APAC

LRLK  AR0,XE1
CALL  ROUOF

LAC   XK2,SCALE ;Calculate xe2

LT    XK3
MPY   A23

LTA   U
MPY   B2
APAC

LRLK  AR0,XE2
CALL  ROUOF

;No need to update xe3 (xe3 = xk3)

LT    XE1 ;Calculate ye
MPY   C1
```

```
PAC

LRLK  ARO,YE
CALL  ROUOF

B     WAIT ;Loop

; Rounding, overflow and shifting function (11 cycles)

ROUOF BLZ   NEG ;Check if number negative
ADD   ONE,SCALE-1 ;Round
SACH  *,16-SCALE ;Store value
SUB   MAXNUM,SCALE ;Subtract scaled max pos number
BLEZ  NOOV ;If acc <= 0 then no overflow
ZALS  MAXNUM ;else store max num
SACL  *
RET

NEG ADD   ONE,SCALE-1 ;Round
SACH  *,16-SCALE ;Store value
SUB   MINNUM,SCALE ;Subtract scaled min neg number
BGEZ  NOOV ;If acc >= 0 then no overflow
ZALS  MINNUM ;else store min neg number
SACL  *
RET

NOOV NOP
NOP
RET

.endæ
```