A Fast Adaptive Controller for Motion Control

Karl Johan Åström[†] and **Jagannathan Kanniah**[‡]

[†] *Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.*
[‡] *Department of Electronics, Singapore Polytechnic, Singapore 0513*

1. Introduction

Limitations in computing speed have for a long time precluded the use of adaptive regulators for controlling fast processes of the type often found in motion control. The advent of digital signal processors (DSP), which admits fast computations at reasonable costs, have removed one obstacle for use of advanced algorithms for motion control. This paper describes implementation of a fast adaptive controller using a DSP. The algorithm used is based on robust pole-placement control and a square-root implementation of a recursive estimator. Special attention is given to pre- and post-filtering, which are also handled adaptively using dual rate sampling. The system is implemented using a Macintosh computer with DSP, AD- and DA-boards from National Instruments. The DSP board uses the Texas Instruments TMS320C30, which has floating point hardware. The code is written in C and LabView is used as the man-machine interface. The adaptive controller can be run at sampling rates of several kHz.

2. Adaptive Prefiltering

High-quality measurements are essential for good control. Because of the aliasing that occurs in sampling it is important to use prefiltering to reduce signal components with frequencies above the Nyquist frequency. Proper design of a prefilter is essential in all digital systems (Åström and Wittenmark, 1990). In adaptive systems there is an additional difficulty because the aliased signals may give rise to estimation errors, see (Åström and Wittenmark, 1989). For high-performance systems with fixed sampling rates it is common practise to design analog prefilters, since the sampling frequency is known. An analog filter with variable bandwidth is complicated and costly. For a general purpose system, where the sampling rates may vary significantly, a fixed prefilter matched with the fastest sampling rate is typically used. Such a system may have poor performance at low sampling rates. In a high-performance system it is therefore attractive to use a prefilter that is always matched to the sampling rate of the controller. A schematic diagram of such a prefilter is shown in Figure 1. The filter has a fixed analog prefilter that is matched to the fast sampling rate. The sampled signal is then prefiltered filtered digitally to give sampled signals y_k at the slower sampling rate used by the controller. There are many approaches to design a prefilter. In our particular application it is possible to state the filtering problem in a reasonable way that admits a simple analytic solution. The goal is to reduce signal components with frequencies higher than

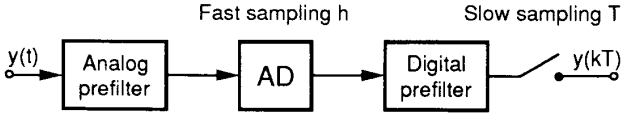

Figure 1. Schematic diagram of a system with adaptive prefiltering.

0-7803-1223-6/93/$3.00 © 1993 IEEE

the Nyquist frequency to negligible levels and to make the signal transmission up to the Nyquist frequency as close to unity as possible. Consider the case of a continuous-time prefilter. The properties expressed above can be captured by the following optimization problem. Find a filter transfer function G that minimizes the loss function

$$J(G) = \int_0^{\omega_N} |G(iw) - 1|^2 dw + \int_{\omega_N}^{\infty} |G(iw)|^2 dw \tag{1}$$

with the constraint $G(0) = 1$. An analogous loss function is obtained in the discrete time case.

FIR filters are attractive to use in signal-processor applications, because they can be implemented very effectively, see (Ahmed, 1991). Consider the case of a digital FIR filter with the transfer function

$$H(z) = b_o + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n} \tag{2}$$

Straightforward calculations show that the loss function (1) is minimized for

$$b_k = b'_k + \alpha \tag{3}$$

where

$$\begin{aligned} b'_o &= \frac{\omega_N h}{\pi} \\ b'_k &= \frac{1}{\pi k} \sin(\omega_N k h) \qquad k = 1, 2, \ldots, n \end{aligned} \tag{4}$$

and

$$\alpha = \frac{1}{n+1} \sum_{k=0}^{n} b'_k \tag{5}$$

3. Adaptive Postfiltering

In the traditional implementation of a digital controller the result of the digital control calculations is sent to a DA-converter. This is typically designed so that its output is kept constant over the sampling interval. The action of the DA-converter can thus be well approximated as a first-order hold circuit. The control signal sent to the process will then be piecewise constant. This implies that there will be drastic changes of the control signal at the sampling instants. Such jumps will excite high-frequency modes of the process. This is highly undesirable in any control system. For adaptive systems it may also give a serious deterioration of the parameter estimation. A simple analog post filter with response time of about one fourth of the control interval may somewhat alleviate the problem, but will not solve it. A better solution is to replace the zero-order hold by another hold circuit.

The predictive first order hold circuit (Bernhardsson, 1990), which makes the control signal linear between the sampling instants, is an attractive solution. Such a hold circuit will work as follows: Let u be the control signal and let T be the sampling period used by the controller. At the sampling instant kT the hold circuit should then deliver the signal

$$u(t) = u(kT) + \frac{t - kT}{T} (u(kT + T) - u(kT))$$

At time kT the control law should thus deliver a prediction of the control signal at time $kT + T$. This explains the name *predictive first-order hold.* If the control law is described as

$$R(q)u(q) = T(q)u_c(q) - S(q)y(q) \tag{6}$$

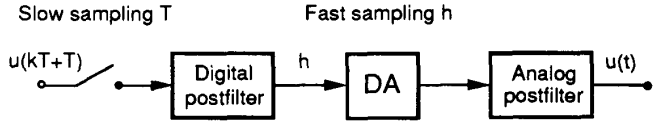

Figure 2. Schematic diagram of a system with adaptive postfiltering.

the polynomials should thus have the property

$$\begin{aligned} \deg R &\geq \deg S + 1 \\ \deg R &\geq \deg T + 1 \end{aligned} \tag{7}$$

Notice that the predictive first-order hold is different from the first-order hold, which gives a discontinuous output. It is possible to implement the predictive first-order hold with analog equipment, but this requires special equipment. A very good approximation is, however, easily implemented using dual rate sampling as is shown in Figure 2. At the sampling instant $t = kT$ the signal $u(kT + T)$ is sent to the analog postfilter. This filter has stored the previous value $u(kT)$ and it generates the output

$$u(l) = \frac{h}{T}\left(u(k+1) - u(k)\right) l \tag{8}$$

where h is the fast sampling rate. This output is then sent to the DA converter. This implementation means that the linear signal from the predictive first-order hold is approximated with a staircase function having many steps. If the fast sampling rate is fixed, an analog postfilter can also be added as indicated in the figure.

4. Adaptive Control

To design an adaptive controller we need a model, an estimation scheme, and a method for control design. Use of the adaptive postfilter only requires marginal modification of the standard methods in (Åström and Wittenmark, 1989). The pulse transfer function obtained by zero-order hold sampling is

$$H_{ZOH}(z) = (1 - z^{-1})\mathcal{Z}\mathcal{L}^{-1}\left(\frac{G(s)}{s}\right) \tag{9}$$

This formula is obtained from the observation that the input signal is piecewise constant. To obtain a similar formula for predictive first-order hold sampling we simply have to observe that the input is piecewise linear. Hence

$$H_{POF}(z) = \frac{(z-1)}{T}\left((1 - z^{-1})\mathcal{Z}\mathcal{L}^{-1}\left(\frac{1}{s}\cdot\frac{G(s)}{s}\right)\right) = \frac{(z-1)^2}{Tz}\mathcal{Z}\mathcal{L}^{-1}\left(\frac{G(s)}{s^2}\right) \tag{10}$$

The input-output relation can be written as

$$A(q)y(k) = B(q)u(k) \tag{11}$$

Notice that this is different from the model obtained with zero-order hold sampling, because polynomials A and B have the same degree in Equation (11). Because of the predictive first-order hold there is also an extra delay in the controller as discussed in Section 3.

Parameter estimation is done in the usual way with datafilters and a square-root algorithm as described in (Åström and Wittenmark, 1989). The control design is based on pole placement. Parameters of the controller (6) are thus obtained by finding polynomials R and S that satisfy the Diophantine equation

$$AR + BS = A_o A_m \tag{12}$$

and the degree constraints (7). Integral action is obtained by requiring that polynomial R has a factor $z-1$. Similarly robustness can be improved by requiring that $z+1$ is a factor of polynomial S. This implies that the controller has zero gain at the Nyquist frequency.

The following numbers gives an indication of the computing times for the different tasks. The time for an AD conversion is 40 μs. The prefilter requires 30 μs for a filter with 100 taps. Computing the control signal from Equation (6) with anti-windup protection requires 20 μs for a second-order controller. One iteration of the parameter estimator requires 235 μs for a model with five parameters. Of this time 5 μs is spent on updating the regression vector. Solving the diophantine equation requires 40 μs for the same model. In addition there is a total overhead of 15 μs. The numbers do not scale simply with the complexity of the model and the controller. Because of the architecture of the DSP and the coding used, the penalty for increasing model complexity is not so severe.

5. Experiments

The controller was implemented using a Macintosh II computer, which served as a host. The computer was provided with a National Instrument Board NB-DSP2300, which is based on the TMS320c30 digital signal processor (National Instruments, 1989). The AD and DA conversion was done using the The National Instruments Board NB-M10-16, which have 16 channels of analog input with 12 bit resolution and 2 channels of analog output (National Instruments, 1990). The code was written in C and compiled using the TI compiler for the DSP, (Texas Instruments, 1986; Texas Instruments, 1990), which runs on the Macintosh II under the Macintosh Programmers Workshop (MPW).

The adaptive controller was implemented in a well structured form so that it could be run in many different ways, with different sampling rates in the controller and the prefilter. It was also possible to run the estimator at a slower rate than the control algorithm. To avoid loading the DSP with tasks that were not necessary for the control, analog signals were measured using a storage oscilloscope.

Experiments were performed on several motion control systems. Typical servo tasks were performed by introducing commands and investigating tracking performance and adaptation rates. The convergence rates were similar to those obtained in simulations, e.g. like Figure 5.4 in (Åström and Wittenmark, 1989). Examples of sample experiments are given below.

EXAMPLE 1—A servo controller
This is an experiment with one of the standard servos in the Control Laboratory, (Åström and Lundh, 1992). Extra measurement noise was introduced in the system to emphasize the usefulness of the adaptive prefilter. The sampling rate for the controller was 10 Hz, the adaptive prefilter was run at 1 kHz and the postfilter at 7.23 kHz. The analog output and the analog control signal were measured. The command signal was a square wave. Five model parameters were estimated. The estimates converged after a few steps of the excitation signal. Typical responses obtained are shown in Figure 3. The piecewise linear character of the control signal due to the predictive first-order hold is clearly visible in the figure. Because of the large ratio of $T/h = 723$ the steps in the control signal are not visible. Notice that the measured signal is quite noisy. The prefilter reduces this noise significantly as is seen in Figure 3. The high frequency gain of the controller is 27 and the control signal would saturate without the prefilter. □

The slow sampling rate used in Example 1 does not tax the computational speed of the algorithm. Much faster sampling is used in the next example.

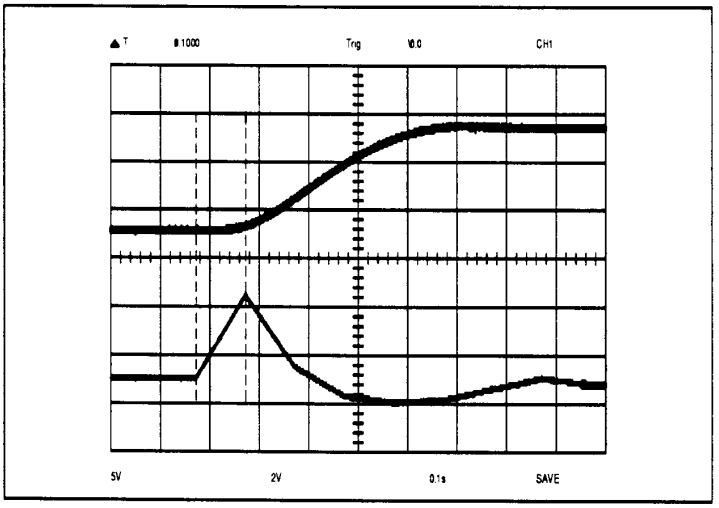

Figure 3. Servo position (upper curve) and drive voltage (lower curve) for adaptive control of a servo.

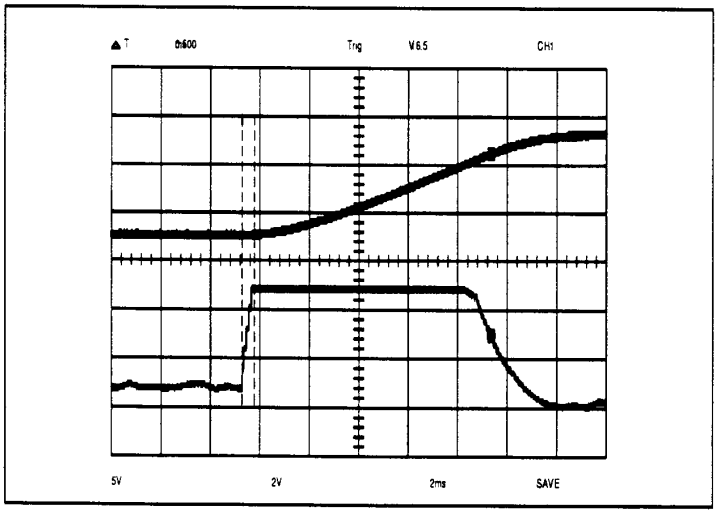

Figure 4. Effect of PFOH-5 steps between control changes.

EXAMPLE 2—A fast controller

In this case we are controlling a system whose dynamics has an integrator and a time constant of 270 ms. It was desired to have a closed loop system with a bandwidth of 600 rad/s. In this case the adaptive controller was run at 2 kHz and the pre- and postfilters were run at 10 kHz. Figure 4 shows a typical step response obtained when the parameters had converged. The steps in the control signal are now visible, since $T = 5$h. □

6. Conclusions

Adaptive control and automatic tuning is attractive for many motion control problems, because it is possible to have one controller, which can easily be used in a wide variety of applications. Fast sampling is, however, required in many applications. In this paper we have summarized experiences of implementing fast adaptive controller for motion control. Special attention has also been given to pre- and post-filtering. It has been found that the control problem can be solved conveniently using robust pole-placement. The code for the adaptive controller has been implemented in a signal processor with floating-point hardware. The software has been structured so that it is easy to use different sampling rates for different parts of the control algorithm. This multi-rate feature is essential to achieve the required performance.

The examples given above indicate the performance that can be obtained. The sampling rate can be increased even further if parameter estimation is not performed every time the controller is sampled. The code is implemented in such a way that it is easy to update the estimates irregularly. This only requires that the regression vector is computed at each sampling instant. The overhead for doing this is small. For typical motion control problems we have found it to be sufficient to update the estimates every fifth sampling of the controller. The sampling rate for the controller in Example 2 can then be increased to 5 kHz. A drawback of implementing the system on the DSP is that there is no real-time kernel available for the DSP. Such a kernel has been developed in the thesis (Carlsson, 1993), and will be used in future projects.

Acknowledgements
This project has been supported by the Swedish National Board for Industrial and Technical Development (NUTEK) under contract 91-01709P. This support is gratefully acknowledged.

7. References

AHMED, I., Ed. (1991): *Digital Control Applications with the TMS 320 Family.* Selected Application Notes. Texas Instruments.

ÅSTRÖM, K. J. and M. LUNDH (1992): "Lund control program combines theory with hands-on experience." *IEEE Control Systems Magazine*, **12:3**, pp. 22–30.

ÅSTRÖM, K. J. and B. WITTENMARK (1989): *Adaptive Control.* Addison-Wesley, Reading, Massachusetts.

ÅSTRÖM, K. J. and B. WITTENMARK (1990): *Computer Controlled Systems—Theory and Design.* Prentice-Hall, Englewood Cliffs, New Jersey, second edition.

BERNHARDSSON, B. (1990): "The predictive first order hold circuit." In *Proceedings of the 29th IEEE Conference on Decision and Control*, pp. 1890–1891, Honolulu, Hawaii.

CARLSSON, A. (1993): "Liten, snabb realtidskärna i C – implementation för DSP med exempel," (A small, fast real-time kernel in C – Implementation for DSP with example). Master thesis ISRN LUTFD2/TFRT--5469--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

NATIONAL INSTRUMENTS (1989): *NB-MIO-16 User Manual.* National Instruments Corporation, Austin, Texas.

NATIONAL INSTRUMENTS (1990): *NB-DSP2300 User Manual.* National Instruments Corporation, Austin, Texas.

TEXAS INSTRUMENTS (1986): *Digital Signal Processing with TMS320 Family.* Texas Instruments Inc, Texas.

TEXAS INSTRUMENTS (1990): *TMS320c30 Reference Guide.* Texas Instruments Inc, Texas.