Conjunction of: "Energy-based approach for estimating Coulomb and viscous friction" and "Adaptive pulse amplitude pulse width control of systems subject to Coulomb and viscous friction"

Report of internship:

Conjunction of: "Energy-based approach for estimating Coulomb and viscous friction" and "Adaptive pulse amplitude pulse width control of systems subject to Coulomb and viscous friction".

By: J.J.M.v.d.Wijdeven

DCT 2002-29

05-07-2002

Supervised by:
Dr. Tarunraj Singh
Dept. of Mech. & Aero. Eng.
University of Buffalo
Buffalo, NY 14260
http://code.eng.buffalo.edu/tdf/

Acknowledgments

My sincere gratitude goes out to professor Dr. T. Singh, for the support and advice he gave me during my project and for giving me the opportunity to come to University of Buffalo. His inspiration has been vital for the outcome of the project.

I would also like to thank the people in the lab: Dirk Tenne, Ulrich Staehlin, Jae-Jun Kim and Nidal Al-Masoud, for making me feel at home at UB. With their help, I succeeded to overcome the many challenges I encountered. Thanks to their kindness, I enjoyed every day of my stay at UB.

Finally, my gratitude goes out to my parents and Suzanne Marcelis for their unconditional support.

Preface

As the title implies, this report consists out of two topics which are integrated into *one* report. The topics are discussed in different articles, resulting in two seemingly incoherent stories.

However, the two subjects can be combined into one study. This will be briefly discussed at the end of this report in *Concluding Remarks*.

Energy-based approach for estimating Coulomb and Viscous Friction

Jeroen J.M. van de Wijdeven
Department of Mechanical Engineering
Technische Universiteit Eindhoven
Eindhoven, The Netherlands

Contents

1 Introduction **2**

2 System Modeling **4**

3 Conservation of Energy **5**

4 The Adaptation Algorithm **7**

5 Noise Interference **9**

6 Numerical Simulations **15**

6.1 Numerical Simulations With a Feedforward System 15

6.2 Numerical Simulations With a Feedback System 20

7 Conclusions and Recommendations **24**

Chapter 1

Introduction

Every mechanical system is subjected to some form of friction. Many processes depend on friction in order to function properly, for instance tires and brakes. However, in control systems, friction may result in undesirable behavior like limit cycles and steady state errors. To accurately control a system, it is necessary to have good knowledge about friction so that it can be compensated for.

A classic static friction model describes the effects of Coulomb friction and viscous damping. More advanced static models include stiction and the Stribeck effect as well. Stiction describes the break-away force or torque that is required to impel a system from zero velocity. The Stribeck effect represents the decrease of the friction force at increasing, yet low velocities. For accurate modeling of the friction dynamics, static models are not sufficient and dynamic models have to be included in the model.

Dahl [1] proposed a first-order dynamic friction model describing the spring-like behavior of the displacement of a system during stiction (presliding). Canudas de Wit *et al.* [2] extended the Dahl model in order to include arbitrary steady-state friction characteristics, such as the Stribeck effect. Dupont *et al.* [3] introduced a dynamic friction model based on an elasto-plastic model to describe presliding and sliding displacement of a system. In contrast to the model proposed by Canudas de Wit *et al.* [2], the elasto-plastic friction model by Dupont *et al.* [3] does not present non-existing drift during stiction.

The effectiveness of a friction model is dependent on the knowledge about the parameters of that model. Several papers present adaptation algorithms in order to estimate the parameters in different friction models. Markho [4] derived a closed form solution for a free vibrating mass-spring system under influence of Coulomb friction and viscous damping. For systems with relatively low damping, experimental data is used to obtain the damping parameter of a equivalent system without Coulomb friction. Friedland and Park [5] proposed a non-linear reduced-order observer for estimation of the Coulomb friction parameter of a system. The constraint that the velocity must be bounded away from zero, is relaxed by Yazdizadeh and Khorasani [6], who presented a Lyapunov scheme for finding the non-linear estimator structure of the observer. Liao and Chien [7] extended Yazdizadeh and Khorasani [6] by deriving an exponentially stable friction compensator together with a linear feedback controller.

Next to the algorithms discussed above, a Kalman filter [8] is a commonly used filter, which uses a recursive algorithm for estimating linear parameters. If the parameters in the model appear non-linear, parameter estimates can be obtained by linearizing the model and using an extended Kalman filter.

The algorithms presented in Yazdizadeh and Khorasani [6] and Liao and Chien [7] perform very well for systems with Coulomb friction. Their approaches turned out to be unsuccessful for estimating friction parameters in systems with Coulomb friction and viscous damping. The extended Kalman filter [8] has the drawback that it is based on a linearized model, which may lead to uncertainties in the estimated parameters.

The goal of this report is to find an algorithm that is capable of estimating the parameter of Coulomb friction as well as that of viscous damping.

The report is organized as follows: In Chapter 2, a model of a flexible arm under the influence of friction is derived. The idea behind conservation of energy is explained in Chapter 3. This is followed by the description of the adaptation algorithm in Chapter 4. In Chapter 5, the effect of noise on the adaptation algorithm is studied, followed by Chapter 6 which presents results from numerical simulations. This report ends with conclusions and recommendations in Chapter 7.

Chapter 2

System Modeling

The model used in this report, represents a flexible arm system under the influence of friction e.g a reading/writing arm in a hard disk drive. The driving force is applied to the first mass, which is experiencing friction. The second mass is connected to the first mass by a flexible arm, which can be modelled by a single spring and no damping. Figure 2.1 presents the model.

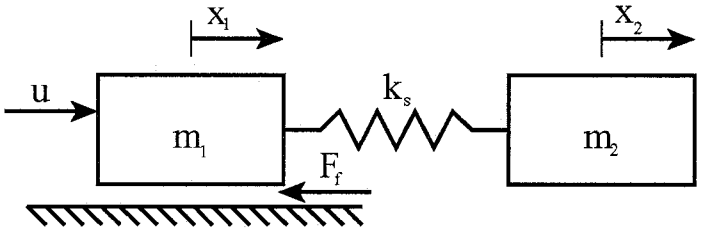

Figure 2.1: Flexible arm model under influence of friction

In Figure 2.1 x_1 and x_2 represent the displacement of the first and second mass respectively, u is the input signal, F_f stands for the friction force and k_s, m_1 and m_2 are the spring constant and the masses.

The friction force operating on the system has been chosen to be a combination of Coulomb friction with coefficient f_c and viscous damping with coefficient c. The equation for the friction is given by:

$$F_f = f_c \; sign(\dot{x}_1) + c\dot{x}_1. \tag{2.1}$$

Using Eq.(2.1), the equations of motion for the two masses are:

$$m_1\ddot{x}_1 + k_s(x_1 - x_2) = u - f_c \; sign(\dot{x}_1) - c\dot{x}_1 \tag{2.2}$$

$$m_2\ddot{x}_2 + k_s(x_2 - x_1) = 0. \tag{2.3}$$

Chapter 3

Conservation of Energy

The concept behind estimating the coefficients in the friction model is based on conservation of energy. The fact that energy is never lost, only converted, is used to isolate the dissipative energy caused by friction from the input energy, the kinetic energy and potential energy. Conservation of energy results in:

$$E_{in}(t) = E_{pot}(t) + E_{kin}(t) + E_{dis}(t). \tag{3.1}$$

The input energy is defined as the integral of the input signal u times the velocity $\dot{x}$ over time and is given by:

$$E_{in}(t) = \int_0^t u\dot{x}_1 dt. \tag{3.2}$$

In order to calculate the potential and kinetic energy, knowledge about the values for the masses and the stiffness is necessary. Since these values are relatively easy to derive from the transfer functions of the system, it is assumed that they are known. The potential and kinetic energy equations are:

$$\begin{aligned} E_{pot}(t) &= \int_0^t (K\underline{x}(t))^T \dot{\underline{x}}(t)dt = 0.5\underline{x}^T(t)K\underline{x}(t) \\ &= 0.5k_s\left(x_1^2(t) + x_2^2(t)\right) - k_s x_1(t)x_2(t) \qquad (3.3) \\ E_{kin}(t) &= \int_0^t (M\ddot{\underline{x}}(t))^T \dot{\underline{x}}(t)dt = 0.5\dot{\underline{x}}^T(t)M\dot{\underline{x}}(t) \\ &= 0.5m_1\dot{x}_1^2(t) + 0.5m_2\dot{x}_2^2(t). \qquad (3.4) \end{aligned}$$

Using Eq.(3.2), Eq.(3.3) and Eq.(3.4), the measured dissipation energy can be obtained. Hereby it is assumed that the positions as well as the velocities of both masses can be measured.

On the other hand, a theoretical expression for the dissipation energy can be derived by integrating the product of friction and the velocity of the first mass over time, which is:

$$E_{dis}(t) = \int_0^t [f_c \ sign(\dot{x}_1) + c\dot{x}_1]\dot{x}_1 dt. \tag{3.5}$$

For proper use in a discrete time environment, the expressions for the different energy terms are converted into discrete time notation. This is done by approximating the integrals by summations using the backward Euler method. The sampling time is represented by T.

$$\begin{aligned}
E_{in}(k) &= T\sum_{i=0}^{k} u(i)\dot{x}_1(i) && (3.6)\\
&= E_{in}(k-1) + Tu(k)\dot{x}_1(k) \\
E_{pot}(k) &= 0.5k_s\left(x_1^2(k) + x_2^2(k)\right) - k_s x_1(k)x_2(k) && (3.7)\\
E_{kin}(k) &= 0.5m_1\dot{x}_1^2(k) + 0.5m_2\dot{x}_2^2(k) && (3.8)\\
E_{dis}(k) &= T\sum_{i=0}^{k} \left[f_c \ sign(\dot{x}_1(i)) + c\dot{x}_1(i)\right]\dot{x}_1(i) && (3.9)\\
&= E_{dis}(k-1) + T\left[f_c|\dot{x}_1(k)| + c\dot{x}_1^2(k)\right].
\end{aligned}$$

Chapter 4

The Adaptation Algorithm

In order to compensate for friction, it is necessary to have good knowledge about the coefficients in the friction model. One way to obtain that knowledge is to extract the necessary information from experiments.

It is however also possible to include an estimator into the system and derive the coefficients on-line. Slotine and Li [9] present several continuous time algorithms for on-line parameter estimation. Haykin [10] extensively discusses the different parameter estimators, based on discrete time implementation.

The gradient estimator is based on updating the estimates of the parameters, using knowledge about the gradient of the squared error signal, with respect to the parameters.

The idea behind least squares estimators is to minimize a defined cost function, from which estimates for the parameters are obtained. In order to estimate time-varying parameters, the least squares estimator can be expanded with a learning factor. In discrete time implementation, the former is referred to as Recursive Least-Squares estimators, or RLS estimators, with exponential weighting.

A constraint on the estimators is that the system should be persistent excited, in order to guarantee exponential convergence of the estimated parameters. Nonetheless, it has been decided to use a RLS estimator with an option to implement the learning factor, for estimating the friction parameters.

The essence of RLS is to extract parameter information from measurements and compare them with a theoretically derived estimation of the parameters. The general model in the linear parametrization form presentation, is given by:

$$y(k) = W(k)\underline{a}. \tag{4.1}$$

In Eq.(4.1) $y(k)$ contains the measurement information or output, $\underline{a}$ contains the parameters and $W(k)$ contains the relations between the parameters and the measurement information.

In case of unknown parameters, Eq.(4.1) has to be altered to:

$$\hat{y}(k) = W(k)\underline{\hat{a}}(k). \tag{4.2}$$

In Eq.(4.2), $\underline{\hat{a}}$ represents the estimate for $\underline{a}$. Note that the parameters in $\underline{\hat{a}}$ are now time-dependent, since the parameters change in time. The difference between $\hat{y}$ and y on time kT is described as the prediction-error $e_y(k)$.

For the estimation of the friction parameters, the real output $y(k)$ is chosen to be the difference between the input energy and the potential and kinetic energy, derived from Eq.(3.1). The estimated output $\hat{y}(k)$ is obtained from Eq.(4.2), the vector $\underline{\hat{a}} = \begin{bmatrix} \hat{f}_c & \hat{c} \end{bmatrix}^T$ and W, which is given by:

$$W(k) = \begin{bmatrix} T\sum_{i=0}^{k} |\dot{x}_1(i)| & T\sum_{i=0}^{k} \dot{x}_1^2(i) \end{bmatrix}. \tag{4.3}$$

The expressions for $y(k)$ and $\hat{y}(k)$ are:

$$\begin{aligned} y(k) &= T\sum_{i=0}^{k} u(i)\dot{x}_1(i) - 0.5k_s\left(x_1^2(k) + x_2^2(k)\right) \\ &\quad + k_s x_1(k)x_2(k) - 0.5m_1\dot{x}_1^2(k) - 0.5m_2\dot{x}_2^2(k) \qquad (4.4) \\ \hat{y}(k) &= \hat{f}_c(k-1)T\sum_{i=0}^{k} |\dot{x}_1(i)| + \hat{c}(k-1)T\sum_{i=0}^{k} \dot{x}_1^2(i). \qquad (4.5) \end{aligned}$$

Haykin [10] presents the discrete time RLS algorithm, which is used for the friction estimation:

$$\begin{aligned} \pi(k) &= P(k-1)W(k) && (4.6) \\ P(k) &= \lambda^{-1}P(k-1) - \lambda^{-1}K(k)W^H(k)P(k-1) && (4.7) \\ K(k) &= \frac{\pi(k)}{\lambda + W^H(k)\pi(k)} && (4.8) \\ e_y(k) &= y(k) - \hat{a}^H(k-1)W(k) && (4.9) \\ \hat{a}(k) &= \hat{a}(k-1) + K(k)e_y^H(k). && (4.10) \end{aligned}$$

The superscript H stands for the Hermitian transpose. In this case, all values are real and thus H can be replaced by T. In Haykin [10] K is described as the time-varying gain vector and P as the inverse correlation matrix. The λ in the algorithm represents the forgetting factor. Generally the forgetting factor is selected to be smaller than and close to 1 and is set to 1 if all data is to be equally weighted.

For the initiation of the adaptation algorithm, initial conditions for $P(0)$, $\hat{f}_c(0)$ and $\hat{c}(0)$ are required. $\hat{f}_c(0)$ and $\hat{c}(0)$ should be chosen as accurately as possible and the inverse covariance matrix $P(0)$ is set to $P(0) = \delta^{-1}I$. δ should be large if the sensors are noisy and can be small otherwise.

Chapter 5

Noise Interference

The adaptation algorithm, presented in Chapter 4 uses measurements of position and velocity in order to obtain estimates for the coefficients of the friction model. Measurements can contain a stochastic noise signal, due to noise. In this chapter, the influence of noise on the adaptation algorithm is studied. It is assumed that the noise present in the system is white, has zero mean and that there is no correlation between different noise signals.

To make an analysis of the influence of noise possible, the measurement x_m is divided into the true deterministic value x_t together with stochastic white noise. Basic properties for x_m and $\dot{x}_m$ are given by:

$$x_m(k) = x_t(k) + w(k) \tag{5.1}$$
$$\dot{x}_m(k) = \dot{x}_t(k) + v(k) \tag{5.2}$$

$$E\{v\} = E\{w\} = 0 \quad E\{vw\} = 0 \tag{5.3}$$
$$E\{v^2\} = \sigma_v^2 \quad E\{w^2\} = \sigma_w^2. \tag{5.4}$$

σ represents the standard deviation of a noise signal.

Eq.(5.1) and Eq.(5.2) are inserted into Eq.(3.6), Eq.(3.7), Eq.(3.8), Eq.(3.9) and Eq.(4.3), after which the expected values of the different terms are taken in order to obtain an expression of the impact of noise on the energies through time.

The first energy term that is studied is the input energy. Using Eq.(5.2), the expected value of the input energy is given by:

$$
\begin{aligned}
E\{E_{in}(k,\dot{x}_{1,m})\} &= E\left\{T\sum_{i=0}^{k}u(i)\left(\dot{x}_{1,t}(i)+v_1(i)\right)\right\} \\
&= T\sum_{i=0}^{k}u(i)\dot{x}_{1,t}(i)+T\sum_{i=0}^{k}E\left\{u(i)v_1(i)\right\} \\
&= T\sum_{i=0}^{k}u(i)\dot{x}_{1,t}(i) \\
&= E_{in}(k,\dot{x}_{1,t}). \qquad (5.5)
\end{aligned}
$$

From Eq.(5.5) it is concluded that noise in the output measurements has no effect on the input energy.

Next, the influence of noise on the kinetic energy is derived in:

$$
\begin{aligned}
E\{E_{kin}(k,\dot{x}_{1,m},\dot{x}_{2,m})\} &= E\left\{0.5m_1(\dot{x}_{1,t}(k)+v_1(k))^2+0.5m_2(\dot{x}_{2,t}(k)+v_2(k))^2\right\} \\
&= 0.5m_1\dot{x}_{1,t}^2(k)+0.5m_2\dot{x}_{2,t}^2(k) \\
&\quad +0.5m_1E\left\{v_1^2(k)\right\}+0.5m_2E\left\{v_2^2(k)\right\} \\
&= E_{kin}(k,\dot{x}_{1,t},\dot{x}_{2,t})+0.5m_1\sigma_{v_1}^2+0.5m_2\sigma_{v_2}^2. \qquad (5.6)
\end{aligned}
$$

Eq.(5.6) illustrates that noise in the system results in an extra bias in the kinetic energy.

The impact of noise on the potential energy term can be derived in the same way as the kinetic energy. The result is presented by:

$$
\begin{aligned}
E\{E_{pot}(k,x_{1,m}),x_{2,m}\} &= 0.5k_sx_{1,t}^2(k)+0.5k_sx_{2,t}^2(k)-k_sx_{1,t}x_{2,t} \\
&\quad +0.5k_s\sigma_{w_1}^2+0.5k_2\sigma_{w_2}^2 \\
&= E_{pot}(k,x_{1,t},x_{2,t})+0.5k_s\left(\sigma_{w_1}^2+\sigma_{w_2}^2\right). \qquad (5.7)
\end{aligned}
$$

The result from Eq.(5.7) is similar to that of Eq.(5.6).

From Eq.(5.5), Eq.(5.6) and Eq.(5.7), it can be concluded that noise results in a bias in the measured dissipation energy.

Noise also influences W. The Coulomb and viscous terms in W will be analyzed separately. The first part, related to the Coulomb friction, exists of a summation of the absolute velocities of the first mass over time. In order to find the influence of noise on the Coulomb term, the expected value of an absolute function will be derived first, after which the expected value of the Coulomb term of W can easily be found.

To start the derivation, the following definitions are presented:

$$
\begin{aligned}
X: &= \dot{x}_{1,m} = \dot{x}_{1,t} + v_1 && (5.8)\\
\overline{X} &= E\{X\} = \dot{x}_{1,t} && (5.9)\\
E\{(X-\overline{X})^2\} &= \sigma_{v_1}^2 && (5.10)\\
Y: &= |X| && (5.11)\\
y &= g(\dot{x}_{1,m}) && (5.12)\\
g_1^{-1}(y) &= -y \quad \text{for} \quad \dot{x}_{1,m} < 0 && (5.13)\\
g_2^{-1}(y) &= y \quad \text{for} \quad \dot{x}_{1,m} \geq 0. && (5.14)
\end{aligned}
$$

More detailed definitions can be found in Soong [11].

For a random function Y, the probability distribution function (PDF) is given by:

$$
f_Y(y) = \sum_{j=1}^{r} f_X[g_j^{-1}(y)] \left| \frac{dg_j^{-1}(y)}{dy} \right|. \tag{5.15}
$$

f_X is the PDF of X and r stands for the number of different solutions for $g^{-1}(y)$, in this case $r = 2$. It is assumed that $f_X(x)$ has a Gaussian distribution.

Using Eq.(5.15), the PDF of Y is presented by:

$$
f_Y(y) = f_X[\dot{x}_{1,m} = -y] + f_X[\dot{x}_{1,m} = y] \quad with \quad y \geq 0. \tag{5.16}
$$

Note that the boundaries for y lay between 0 to ∞, since the probability that an absolute function results in a negative output is equal to zero.

Figure 5.1 presents the PDF for Gaussian distributed absolute functions with means 0 and 2 and a σ of 1. The figure shows that the PDF of $y = |\dot{x}|$ function is zero for negative y. The PDF of $y = |\dot{x}|$ is obtained from the PDF of $y = \dot{x}$ by folding the negative part of the PDF of $y = \dot{x}$ around the y-axis and adding it to the positive PDF values.

With the PDF from Eq.(5.16), the estimated value for an absolute function can be derived:

$$
\begin{aligned}
E\{Y\} &= \int_0^\infty y f_Y(y) dy && (5.17)\\
&= \int_0^\infty y \left[\frac{1}{\sigma_{v_1}\sqrt{2\pi}} \exp\left(\frac{-(-y-\dot{x}_{1,t})^2}{2\sigma_{v_1}^2} \right) \right.\\
&\quad \left. + \frac{1}{\sigma_{v_1}\sqrt{2\pi}} \exp\left(\frac{-(y-\dot{x}_{1,t})^2}{2\sigma_{v_1}^2} \right) \right] dy. && (5.18)
\end{aligned}
$$

In order to solve Eq.(5.18), two dummy variables z_1 and z_2 are introduced:

$$
z_1 = \frac{-y - \dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}} \quad z_2 = \frac{y - \dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}}. \tag{5.19}
$$

The expected value value for Y becomes:

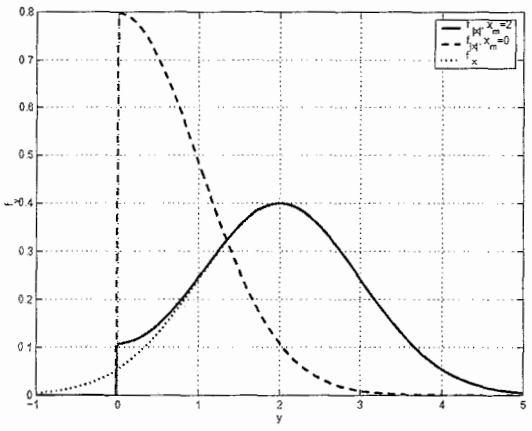

Figure 5.1: PDF of $y = |\dot{x}|$ compared with $y = \dot{x}$

$$
\begin{aligned}
E\{Y\} &= \int_{-\infty}^{\frac{-\dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}}} \frac{-1}{\sqrt{\pi}}(\sqrt{2}\sigma_{v_1} z_1 + \dot{x}_{1,t}) \exp\left(-z_1^2\right) dz_1 \\
&\quad + \int_{\frac{-\dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}}}^{\infty} \frac{1}{\sqrt{\pi}}(\sqrt{2}\sigma_{v_1} z_2 + \dot{x}_{1,t}) \exp\left(-z_2^2\right) dz_2 \\
&= \left[\frac{-\sigma_{v_1}}{\sqrt{2\pi}} \exp\left(-z_1^2\right)\right]_{-\infty}^{\frac{-\dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}}} + \left[\frac{-\sigma_{v_1}}{\sqrt{2\pi}} \exp\left(-z_2^2\right)\right]_{\frac{-\dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}}}^{\infty} \\
&\quad + 0.5\dot{x}_{1,t}\left[\mathrm{erf}(z_1)\right]_{-\infty}^{\frac{-\dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}}} + 0.5\dot{x}_{1,t}\left[\mathrm{erf}(z_2)\right]_{\frac{-\dot{x}_{1,t}}{\sqrt{2}\sigma_{v_1}}}^{\infty} \\
&= \sqrt{\frac{2}{\pi}}\sigma_{v_1} \exp\left(-\frac{\dot{x}_{1,t}^2}{2\sigma^2}\right) + \dot{x}_{1,t}\mathrm{erf}\left(\frac{\dot{x}_{1,t}}{\sqrt{2}\sigma}\right). \qquad (5.20)
\end{aligned}
$$

Figure 5.2 presents the expected values for $|\dot{x}_{1,t}|$ and $|\dot{x}_{1,m}|$.

Eq.(5.20) presents the expected value of an absolute function of a noisy signal. The result however is expressed in terms of the true velocity $\dot{x}_{1,t}$, while only $\dot{x}_{1,m}$ is known.

In order to solve this problem, $\dot{x}_{1,t}$ in Eq.(5.20) is substituted by $\dot{x}_{1,m} - v_1$. The resulting equation is very complex and taking the expected value from Eq.(5.20) will be an arduous job. To avoid having to find the expected value for this complex equation, a 6^{th} order polynomial approximation for Eq.(5.20) is derived. This polynomial function approximates Eq.(5.20) between $\pm 3v_1$. Outside these boundaries, it's assumed that the expected value for $|\dot{x}_{1,m}|$ is equal to $|\dot{x}_{1,t}|$. The approximation for Eq.(5.20) is given by:

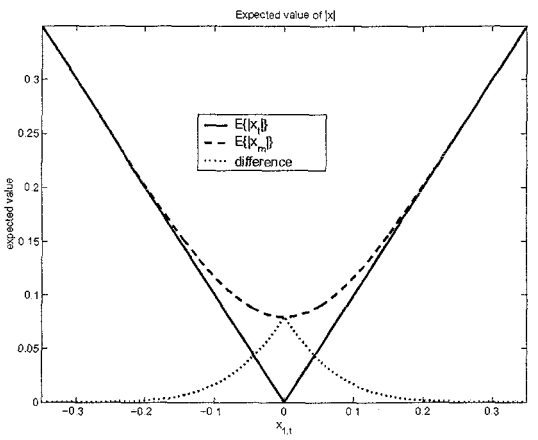

Figure 5.2: expected values for $|\dot{x}_{1,t}|$ and $|\dot{x}_{1,m}|$

$$E\{E\{Y\}\} = A_1 \left(\dot{x}_{1,m} - v_1\right)^6 + A_2 \left(\dot{x}_{1,m} - v_1\right)^4 + A_3 \left(\dot{x}_{1,m} - v_1\right)^2 + A_4 \qquad \text{for } |\dot{x}_{1,m}| \leq 3v_1 \tag{5.21}$$

$$E\{E\{Y\}\} = |\dot{x}_{1,t}| \qquad \text{for } |\dot{x}_{1,m}| > 3v_1. \tag{5.22}$$

The odd terms are zero, since the function is symmetric about the y-axis. With

$$E\{v^4\} = 3\sigma_{v_1}^4 \quad E\{v^6\} = 15\sigma_{v_1}^6 \,, \tag{5.23}$$

the expected value for Eq.(5.21) is given by:

$$\begin{aligned} E\{E\{Y\}\} \;=\; & A_1\dot{x}_{1,m}^6 + 15A_1\dot{x}_{1,m}^4\sigma_{v_1}^2 + 45A_1\dot{x}_{1,m}^2\sigma_{v_1}^4 + 15A_1\sigma_{v_1}^6 \\ & + A_2\dot{x}_{1,m}^4 + 6A_2\dot{x}_{1,m}^2\sigma_{v_1}^2 + 3A_2\sigma_{v_1}^4 + A_3\dot{x}_{1,m}^2 + A_3\sigma_{v_1}^2 + A_4. \end{aligned} \tag{5.24}$$

In order to find the bias induced by the noise, Eq.(5.24) is subtracted by $|\dot{x}_{1,t}|$. In Figure 5.3 the expected value for $E\{E\{|\dot{x}_{1,m}|\}\}$ is compared with $E\{\dot{x}_{1,m}\}$ and the final bias is presented.

The resulting bias is approximated by a 4^{th} order polynomial function, expressed in $|\dot{x}_{1,m}|$, since the original expression is too complex for implementation.

To finally find the influence of noise on the Coulomb term of W, the 4^{th} order polynomial function is integrated over time, using a backward Euler approach, represented by $W_c(\dot{x}_{1,m}, T)$. The final result shows a time dependent as well as a velocity dependent bias for the Coulomb term in W due to noise.

The second term of W is related to the viscous damping. The expected value is presented by:

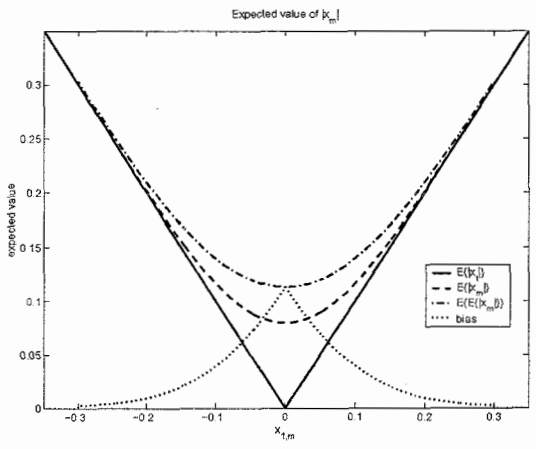

Figure 5.3: expected values for $|\dot{x}_{1,m}|$ and $E\{|\dot{x}_{1,m}|\}$

$$E\left\{T\sum_{i=0}^{k}\dot{x}_{1,m}^2\right\} = T\sum_{i=0}^{k}\dot{x}_{1,t}^2 + \sigma_{v_1}^2 kT. \tag{5.25}$$

Eq.(5.25) shows that the influence of noise results in a time dependent bias in the viscous term of W.

The total energy dissipation equation, including the extra terms introduced by noise are given by:

$$\begin{aligned} E_{measured} &= E_{calculated} \\ E_{measured} &= E_{in}(k,\dot{x}_{1,t}) + E_{kin}(k,\dot{x}_{1,t}\dot{x}_{2,t}) + E_{pot}(k,x_{1,t},x_{2,t}) \\ &\quad +0.5m_1\sigma_{v_1}^2 + 0.5m_2\sigma_{v_2}^2 + 0.5k_s\left(\sigma_{w_1}^2 + \sigma_{w_2}^2\right) \qquad (5.26) \\ E_{calculated} &= W_t(k)\hat{a} + \left[W_c(\dot{x}_{1,m},T) \quad \sigma_{v_1}^2 kT\right]\hat{a}. \qquad (5.27) \end{aligned}$$

Note that Eq.(5.26) and Eq.(5.27) present the *expected* values for the the different energy terms. Compensating noise will be based on these expected values.

Chapter 6

Numerical Simulations

The adaptation algorithm that has been derived in the former chapters, has been simulated in Simulink. The flexible arm, described in Chapter 2 is designed as a continuous time model, representing a real system, the adaptation algorithm is modelled in discrete time. The two segments are connected by Zero Order Hold blocks, simulating the A/D conversion.

The simulations can be divided into two sections: adaptation of the friction parameters in a feedforward system, and adaptation of the parameters in a feedback system. These are discussed in Section 6.1 and Section 6.2 respectively. The values of the parameters used in the simulations, are given in Table 6.1.

Table 6.1: Parameter Values

$m_1 = m_2$	$1\ kg$
k_s	$1\ N/m$
f_c	$0.5\ N$
c	$0.3\ kg/s$
T	$0.01\ s$
$\sigma_{w_1} = \sigma_{w_2}$	$0.1\ m$
$\sigma_{v_1} = \sigma_{v_2}$	$0.1\ m/s$
λ	1

6.1 Numerical Simulations With a Feedforward System

In the feedforward simulations, the plant model is subjected to a bang-bang input. The first mass comes to rest after approximately 25 s, while the second mass will keep oscillating. To clearly see the influence of noise, which is most dominant when velocities are small, the simulation is stopped after 60 s.

To start with, the adaptation algorithm and the plant model are joined together and the adaptation algorithm estimates the friction parameters without interference of noise.

To see what effect the noise has on the values of the estimates, noise is injected into the system. The first plots of Figures 6.1 and 6.2 present the evolution of the estimates for f_c and c for the case where no noise is present and the case where noise is present.

Note that the estimates in the first plots of Figures 6.1 and 6.2 do not converge to the true values. The biases are the result of using ZOH and the discrete time integration of the energy terms, since simulations using continuous time integration and without ZOH *do* converge to the true values.

Looking at the estimates for f_c and c, it can be concluded that noise has a negative influence on the value of the estimates. It has to be noted that the final estimates for f_c and c vary with time on which the simulation is ended, due to the influence of the noise.

To compensate for the influence of noise on the system, the different biases, derived in Chapter 5, are subtracted from $E_{measured}$ and $E_{calculated}$. The second plots of Figures 6.1 and 6.2 present the estimates for the Coulomb and viscous parameters, when the noise is compensated for. The dotted line is the result of using estimates for the σ's equal to 50% of the true values.

With precise estimates for the standard deviations, the estimates for the friction parameters are very close to the true values. However, when the σ's are off from their true values, the estimates for f_c and c are less accurate.

To compensate the effects of deviations between estimate and true σ's, an extra bias and a time-dependent function are introduced into the adaptation algorithm. The extra bias B must compensate for the error in the bias in $E_{measured}$, the time-dependent function $f(kT) \cdot S$ must compensate for the time-dependent error in $E_{calculated}$. In the function, S (as B) is a constant, to be estimated and $f(kT)$ is chosen to be equal to $W_c(\dot{x}_{1,m}, T)$, since $\sigma_{v_1}^2 kT$ is an order σ_{v_1} smaller than $W_c(\dot{x}_{1,m}, T)$ and therefore neglected. Vector $\hat{a}$ and row W become:

$$\hat{a} = \left[\hat{f}_c \quad \hat{c} \quad \hat{B} \quad \hat{S}\right]^T. \tag{6.1}$$

$$W(k) = \left[T\sum_{i=0}^{k}|x_1(i)|, \quad T\sum_{i=0}^{k}\dot{x}^2(k), \quad 1, \quad W_c(\dot{x}_{1,m}, T)\right]. \tag{6.2}$$

Note that errors in the biases are actually non-linear: the biases are multiplied by the friction parameters and thus errors in the biases are a function of the friction parameters too. From simulations, it can be seen that f_c and c converge rapidly to a value close to the true values, after which the parameters converge only very slowly. When the friction parameters do not alter significantly anymore, they are assumed to be constant, which results in a linear estimator for B and S. Although above arguments approve the use of Eq.(6.1) and Eq.(6.2), there will always be some interaction between the estimates for f_c and c and the estimates for B and S.

The third plots of Figures 6.1 and 6.2 present the resulting estimates for the friction parameters using B and S for the case that the σ's are exactly known and for the case that the estimates for the $\sigma's$ is 0.5 times their true values. It is seen that errors in the σ's do not influence the estimates for f_c and c, since B and S compensate for the fluctuations.

The influence of the different kind of compensations on the adaptation algorithm input $e_y(k)$ is illustrated by Figures 6.3, 6.4 and 6.5.

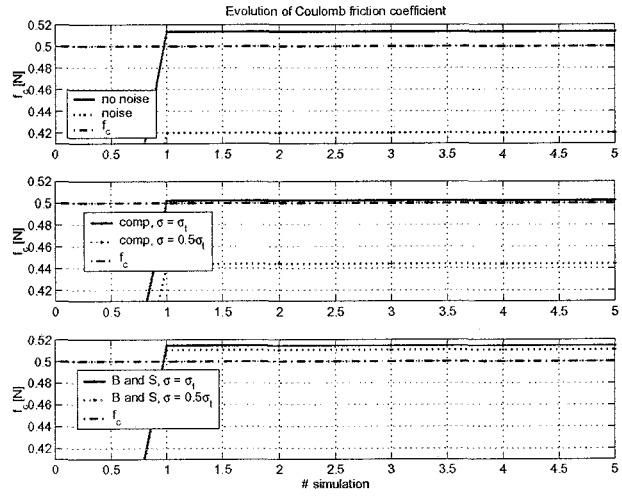

Figure 6.1: Feedforward system: Evolution of the estimate for f_c. 1^{st} plot: no noise present in the system; Noise present in the system. 2^{nd} plot: Full noise compensation; noise compensation with wrong estimate for σ's. 3^{rd} plot: extra compensation by using B and S; extra compensation by using B and S, with wrong estimate for σ's

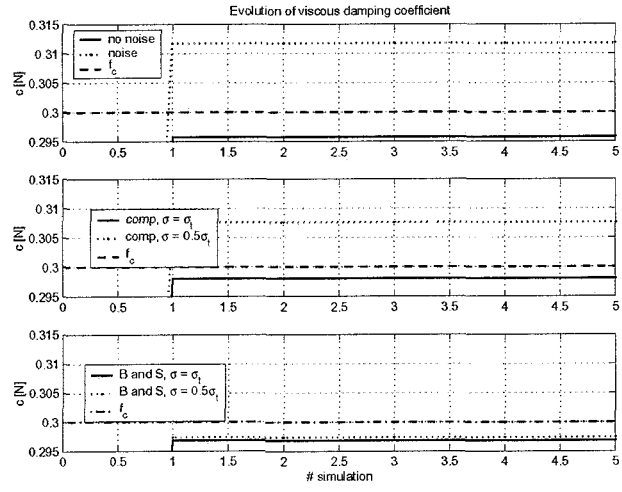

Figure 6.2: Feedforward system: Evolution of the estimate for c. 1^{st} plot: no noise present in the system; Noise present in the system. 2^{nd} plot: Full noise compensation; noise compensation with wrong estimate for σ's. 3^{rd} plot: extra compensation by using B and S; extra compensation by using B and S, with wrong estimate for σ's

It is seen that noise results in a bias and a slope in e_y. When B and S are used, the bias as well as the slope are almost fully compensated for after five simulations.

The Integral of Time-multiplied Squared Error (ITSE) in the different plots represents a quantitative measure of the performance of the different simulations. It is a way to quantify a measurement so that it can be compared with other measurements. ITSE is defined as:

$$ITSE = \int_0^t te^2(t)dt. \tag{6.3}$$

In Eq.(6.3), the square causes a higher penalty for larger errors, while multiplication with time results in larger penalties for errors occurring later on in the simulation. The different ITSE's are normalized so that the first simulation without any noise has a ITSE of 1. From the results it can be seen that B and S stabilize the adaptation input e_y, even in case of uncertainty in noise variance.

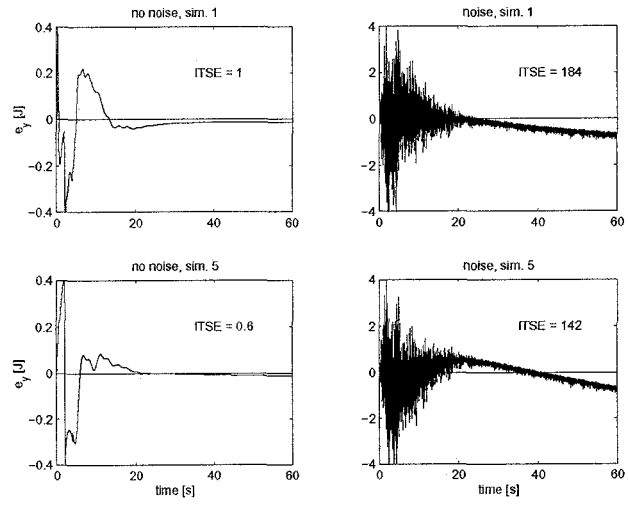

Figure 6.3: Feedforward system: left column: e_y in simulations without noise; right column: e_y in simulations with noise

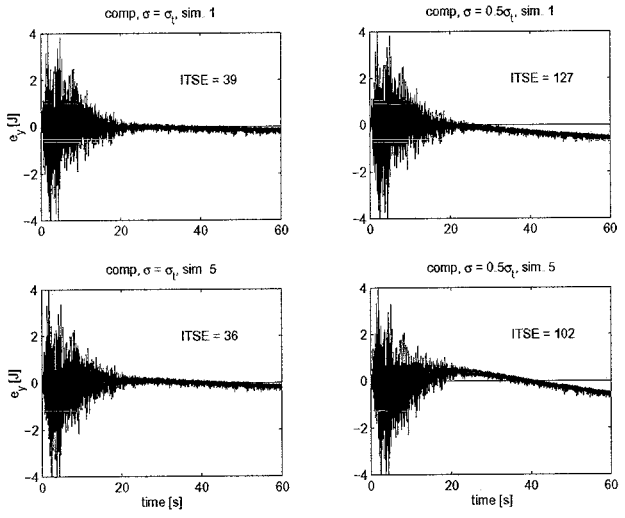

Figure 6.4: Feedforward system: left column: e_y in simulations with full noise compensation; right column: e_y in simulations with noise compensation with wrong estimate for σ's

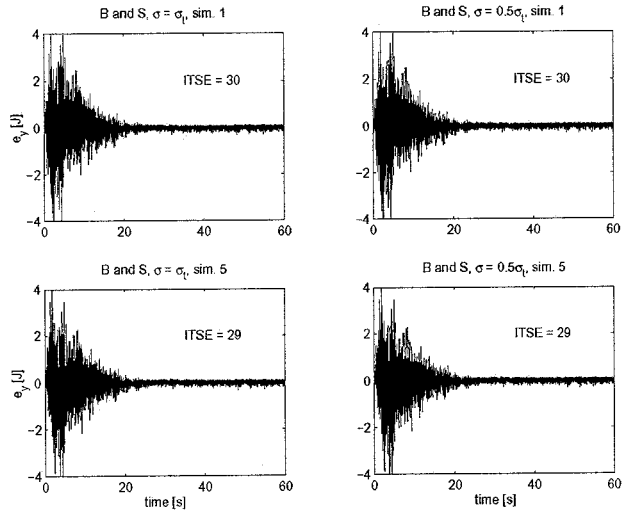

Figure 6.5: Feedforward system: left column: e_y in simulations with extra compensation by using B and S; right column: e_y in simulations with extra compensation by using B and S, with wrong estimate for σ's

6.2 Numerical Simulations With a Feedback System

In order to implement the adaptation algorithm in a feedback system, a feedback controller has to be designed. The controller used in the feedback system, consists of a PD controller and a notch, together with friction compensation for the Coulomb friction as well as the viscous damping. The friction is compensated, using estimates of the friction parameters.

Simulations with the feedback system are similar to those of the feedforward system, so they are not illustrated again. The resulting estimates for f_c and c are presented in Figure 6.6 and 6.7. Figures 6.8, 6.9 and 6.10 present e_y for the different simulations.

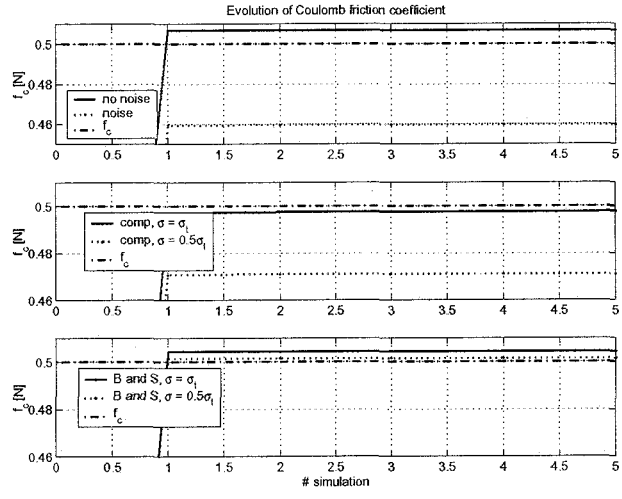

Figure 6.6: Feedback system: Evolution of the estimate for f_c. 1^{st} plot: no noise present in the system; Noise present in the system. 2^{nd} plot: Full noise compensation; noise compensation with wrong estimate for σ's. 3^{rd} plot: extra compensation by using B and S; extra compensation by using B and S, with wrong estimate for σ's

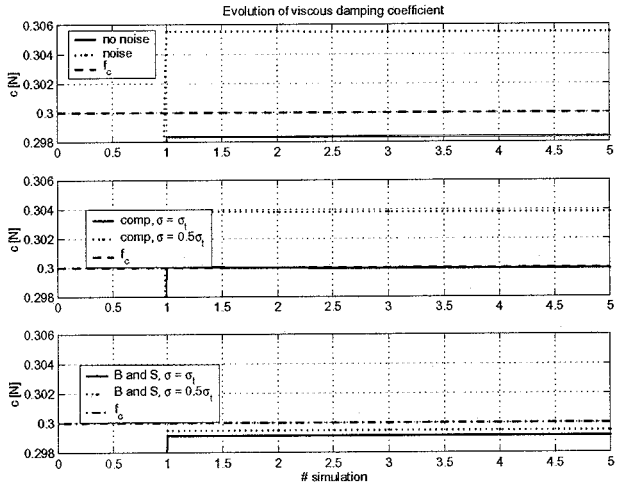

Figure 6.7: Feedback system: Evolution of the estimate for c. 1^{st} plot: no noise present in the system; Noise present in the system. 2^{nd} plot: Full noise compensation; noise compensation with wrong estimate for σ's. 3^{rd} plot: extra compensation by using B and S; extra compensation by using B and S, with wrong estimate for σ's

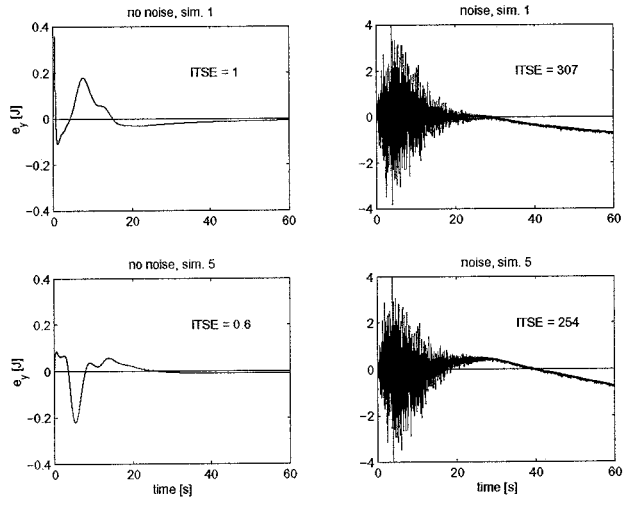

Figure 6.8: Feedback system: left column: e_y in simulations without noise; right column: e_y in simulations with noise

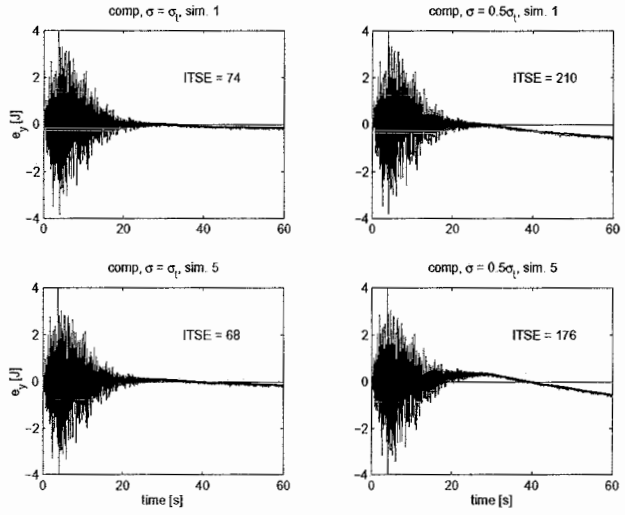

Figure 6.9: Feedback system: left column: e_y in simulations with full noise compensation; right column: e_y in simulations with noise compensation with wrong estimate for σ's

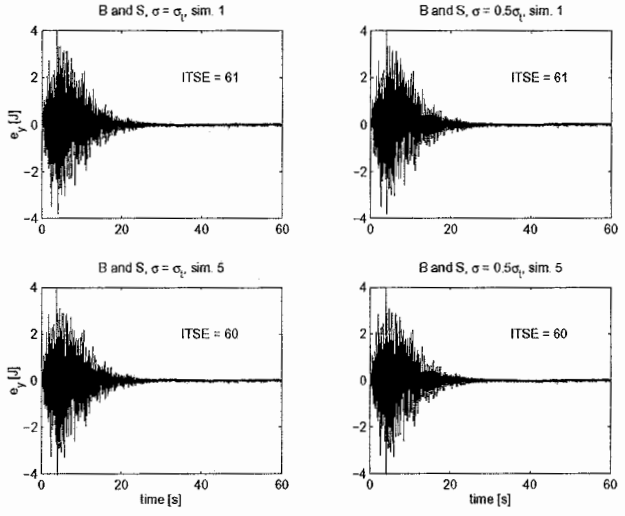

Figure 6.10: Feedback system: left column: e_y in simulations with extra compensation by using B and S; right column: e_y in simulations with extra compensation by using B and S, with wrong estimate for σ's

When Figures 6.1 and 6.2 are compared with Figures 6.6 and 6.7 respectively, one can see that the feedback system gives more accurate estimates for the friction parameters than the feedforward system, even if there is no noise present in the system. The errors in the estimates of the friction parameters in the noise free system are caused by the use of ZOH and discrete time representation of the adaptation algorithm. Apparently the feedback system has partly compensated for these errors, which results in better estimates for the Coulomb and viscous friction coefficients.

However, the conclusion drawn from the different simulations is the same: The adaptation algorithm including B and S gives the most stable results, even under uncertainties in the standard deviations of the noise signals.

Chapter 7

Conclusions and Recommendations

This report proposed an Recursive Least-Square algorithm, capable of estimating the parameters of Coulomb and viscous friction using conservation of energy. The influence of noise on the adaptation algorithm has been studied and compensated for. In order to compensate for uncertainties in the variance of noise signals, an extra bias and a time-dependent function have been introduced.

Numerical simulations on a feedforward system as well as a feedback system present promising results in estimating the friction parameters. The extra bias and the time-dependent function absorb fluctuations in the energy signals due to noise, resulting in accurate estimates for the friction parameters.

Future studies can expand the friction model to include the Stribeck effect. Therefore, the Stribeck effect should be linearized with respect to its parameters in order to estimate the parameters involved with the effect.

In order to complete the study, the friction parameter estimation algorithm has to be implemented on a experimental testbed.

Bibliography

[1] P. Dahl. *A Solid Friction Model.* Tech. Report TOR-0158(3107-18), Aerospace Corp., El Segundo, CA, 1968.

[2] C. Canudas de Wit, H. Olsson, K.J. Åström, and P. Lischinsky. A new model for control of systems with friction. In *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pages 419–425, March 1995.

[3] P. Dupont, V. Hayward, B. Armstrong, and F. Altpeter. Single state elasto-plastic models. In *IEEE Transactions on Automatic Control*, vol.47, pages 787–792, 2002.

[4] P.H. Markho. On free vibrations with combined viscous and coulomb damping. In *ASME Journal of dynamic Systems, Measurements and Control*, 102, pages 283–286, December 1980.

[5] B. Friedland and Y.-J. Park. On adaptive compensation. In *IEEE Transactions on Automatic Control*, vol. 37, no. 10, pages 1609–1612, October 1992.

[6] A. Yazdizadeh and K. Khorasani. Adaptive friction compensation based on the lyapunov scheme. In *Proceedings of the 1996 IEEE Int. Conference on Control Applications*, pages 1060–1065, Dearborn, MI, September 15-18 1996.

[7] Teh-Lu Liao and Tsun-I Chien. An exponentially stable adaptive friction compensator. In *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pages 977–980, May 2000.

[8] R.B. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, Inc., third edition, 1997.

[9] J.-J.E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, Inc., 1991.

[10] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, Inc., fourth edition, 2002.

[11] T.T. Soong. *Probabilistic Modeling and Analysis in Science and Engineering.* John Wiley & Sons, Inc., 1981.

Adaptive Pulse Amplitude Pulse Width Control of Systems subject to Coulomb and Viscous Friction

Jeroen J.M. van de Wijdeven
Dept. of Mechanical Engineering
Technische Universiteit Eindhoven
Eindhoven, 5600 MB,
The Netherlands

Tarunraj Singh
Dept. of Mech. & Aero. Eng.,
University of Buffalo,
Buffalo, NY 14260
http://code.eng.buffalo.edu/tdf/

Abstract

The focus of this paper is on adaptive control of maneuvering rigid bodies in the presence of friction. The paper describes a simple technique which include Pulse Amplitude and Pulse Width modulation to progressively move the system to the desired final position. To account for uncertainty in estimated friction coefficients and approximated system model, an adaptation algorithm is necessary to accurately track the desired position. The proposed technique is suited for discrete time implementation and is illustrated on a rest-to-rest maneuver. The proposed technique is shown to considerably reduce the steady state error which exists in previously proposed Pulse Width controllers.

1 Introduction

Friction is a phenomenon which is ubiquitous. It is desirable in applications such as tires, clutches, brakes etc. and is a challenging problem when precise position control is desired. In control systems, the presence of friction can result in undesirable behavior such as limit cycling and steady state errors. It is therefore necessary that the phenomenon of friction has to be well understood, and compensated for. There exist numerous models for friction spanning the range from the simple Coulomb model [1] to the comprehensive Lund-Grenoble model [2] which accounts for effects such as stiction, Stribeck effect and hysteresis. The model which is selected for the design of controllers is application dependent.

Yoshida and Tanaka [3], proposed applying a dither signal to linearize the nonlinear friction, which in conjunction with a feedback controller is used to position a flexible arm system. Friedland and Park [4] proposed to adaptively estimate the coefficient of Coulomb friction and used the estimate in a feedforward control to cancel the effects of friction. Recently, Liao and Chien [5] modified the adaptive estimation algorithm for tracking control systems to ensure that the tracking error and the parameter errors converge to zero exponentially, in the presence of persistent excitation.

One of the more novel approaches for precise positioning using an adaptive Pulse Width Control (PWC) has been proposed by Yang and Tomizuka [6]. They consider a laboratory

positioning table and study the effect of applying a pulse input on the displacement of the system subject to friction. They arrive at a closed form solution for the total displacement in the presence of Coulomb friction and show it to be a quadratic function of the pulse width. Next, they derive a closed form expression for the displacement as a function of the pulse width, including viscous friction and approximate it to make it compatible with their adaptation algorithm. When their algorithm is implemented in a discrete time environment, the discrepancy between the required pulse width and the pulse width permitted by sampled data system results in limit cycling of the closed loop response.

In this paper the technique proposed by Yang and Tomizuka [6] is modified to require the pulse width to be coincident with an integer multiple of the sampling interval. Simultaneously, the pulse amplitude is calculated to achieve the desired final displacement. An adaptation algorithm is an integral part of the proposed work. Unlike the approach proposed by Yang and Tomizuka where one parameter, which is a function of the friction coefficient, system mass and permitted pulse amplitude is estimated, the new algorithm also estimates the coefficient of Coulomb friction.

Section 2 reviews the adaptive Pulse Width Control technique proposed by Yang and Tomizuka [6]. Section 3.1 describes the variation of the parameter estimated in the work by Yang and Tomizuka as a function of varying displacement. This is the motivation for the use of an adaptation algorithm for precisely positioning the system at the desired value. This is followed by the description of the adaptive Pulse Amplitude Pulse Width Control (PAPWC) in Section 3.2. Section 4 illustrates the proposed technique on a simple rest-to-rest maneuver and compares its performance with adaptive PWC. The paper concludes with remarks and conclusions in Section 5.

2 Adaptive Pulse Width Control

2.1 System Model

The basic idea and motivation for the use of pulse width control (PWC) was introduced by Yang and Tomizuka [6]. With the assumption that the first resonance peak is sufficiently high with respect to the bandwidth and sampling frequency, their model of a X-Y table can be represented by a single mass subject to friction. The friction that is acting on the mass m is assumed to consist of Coulomb friction f_c, stiction f_s and viscous damping c. The equations of motion for this model are:

$$\ddot{x} = \begin{cases} \frac{1}{m}(u - f_c - c\dot{x}) & \text{if } \dot{x} \neq 0 \\ 0 & \text{if } \dot{x} = 0 \text{ and } |u| \leq f_s \\ \frac{1}{m}(u - sgn(u)f_s) & \text{if } \dot{x} = 0 \text{ and } |u| > f_s. \end{cases} \tag{1}$$

If the system described in Eq.(1), is driven by a single pulse with pulse height f_p and pulse width t_p, a closed form expression for the displacement can be derived. In [6] this is done for the model with and without viscous damping. The results are given by:

$$d_{no\ viscous\ damping} = \frac{f_p(f_p - f_c)}{2mf_c} t_p^2 \quad \text{for } f_p > 0 \tag{2}$$

$$d_{with\ viscous\ damping} = \frac{f_p t_p}{c} - \frac{mf_c}{c^2} ln \left[\frac{f_p}{f_c} \left(e^{ct_p/m} - 1 \right) + 1 \right]. \tag{3}$$

Assuming that the displacement is small, it was shown [6] that viscous damping can be neglected compared to the Coulomb friction. Thus, Eq.(2) is a reasonable approximation for small displacements. The displacement now is linearly proportional to the square of the pulse width. The coefficient of the term t_p^2 is represented by one parameter b, with $b \geq 0$. The final expression for the displacement is given as:

$$d(t_p) = bt_p^2\ sgn(f_p), \quad \text{with } b = \frac{f_p(f_p - f_c)}{2mf_c}. \tag{4}$$

Since the direction of the displacement must be equal to the sign of f_p, Eq.(4) includes $sign(f_p)$.

2.2 Adaptation Algorithm

The total control scheme that is presented for the system has two components. The first one is a simple feedback controller used in conjunction with a feedforward controller to compensate for Coulomb friction force. An estimate for the Coulomb friction parameter is obtained through experiments. This controller is used to move the system from its initial position to the vicinity of the desired position. Once the system sticks within an error tolerance area around the reference position, the controller switches from feedback control to the second component, the PWC. The feedback controller must be designed in such a way that the maximum steady state error is smaller than a predefined error tolerance.

The input of the PWC is the error e between the desired position and the current position $x_{ref} - x$ and is used to calculate the pulse width. In [6] two equations for u_p are derived to accomplish the above. The first equation is given by:

$$x(k+1) = x(k) + d(k+1) \tag{5}$$

$$d(k+1) = bu_p(k) \tag{6}$$

$$\text{with } u_p(k) = t_p^2(k)\ sgn\left(f_p(k)\right). \tag{7}$$

The second expression for u_p is found by defining the feedback control law to be:

$$e(k) = x_{ref} - x(k) \tag{8}$$

$$u_p(k) = \frac{K_c}{b} e(k). \tag{9}$$

Using Eq.(7) and Eq.(9), the expression for $t_p(k)$ is given by:

$$t_p(k) = \sqrt{\frac{K_c}{b\ sgn\left(f_p(k)\right)}\ e(k)}. \tag{10}$$

In Eq.(9), K_c is a control parameter with $0 < K_c < 2$ for stability reasons. k stands for the k^{th} pulse and should not be mistaken with the sampling time.

Since b is not known exactly, the pulse width cannot be derived from Eq.(9). Instead t_p is calculated with an estimate $\hat{b}$ of b. With the use of an adaptation algorithm $\hat{b}$ is updated after each pulse. Although [6] presents multiple ways to estimate $\hat{b}$ and $1/\hat{b}$, this paper only presents the self tuning regulator approach for estimating $\hat{b}$. The adaptation algorithm is given by the equations:

$$\begin{aligned}
\epsilon_0^0(k) &= d(k) - \hat{b}(k-1)u_p(k-1) && (11)\\
F^{-1}(k) &= \lambda_1 F^{-1}(k-1) + \lambda_2 u_p^2(k-1) && (12)\\
\hat{b}(k) &= \hat{b}(k-1) + F(k)u_p(k-1)\epsilon_0^0(k) && (13)\\
& F(0) > 0,\ 0 < \lambda_1 \leq 1,\ 0 \leq \lambda_2 < 2.
\end{aligned}$$

ϵ_0^0 represents the error between the real displacement d and the estimated displacement $\hat{b}u_p$. F is referred to as the time-varying gain matrix and λ_1 and λ_2 are parameters related to forgetting previous data. For λ_1 and λ_2 equal to one, all data is equally weighted.

The adaptation algorithm results in pulse widths that can take any positive value. However, in a discrete time system, the pulse width will automatically be rounded to the smallest integer number of the sampling time larger than the calculated pulse width. This is caused by the D/A converter that is assumed to be Zero Order Hold (ZOH). In this way, the calculated pulse is not the real input pulse on the system.

In [6] it is assumed that the pulse widths are relatively small, so that the pulse width calculated using Eq.(4) results in the desired displacement. However, when the required displacement increases, the approximation for the displacement described in Eq.(4) is not accurate anymore. The impact of the value of b on the final displacement of the PWC, when the required motion is large, needs to be studied.

In section 3 a technique to address the issue of the finite sampling time on the control performance is proposed.

3 Adaptive Pulse Amplitude Pulse Width Control

In this section, the influence of relatively large displacements on the estimation of $\hat{b}$ is studied. Next, the original PWC algorithm is modified, such that only an integer number of the sampling time is used to describe the pulse width. The consequences of this on the algorithm will be analyzed.

3.1 Motivation for Adaptation

Figure 1 illustrates the displacement as a function of the pulse width t_p. The solid line corresponds to the displacement, described by Eq.(3), the dotted line corresponds to the

Eq.(4) which is the approximate of Eq.(3) for small displacements. In this plot it is assumed that all the parameters are known. Their values are presented in Table 1.

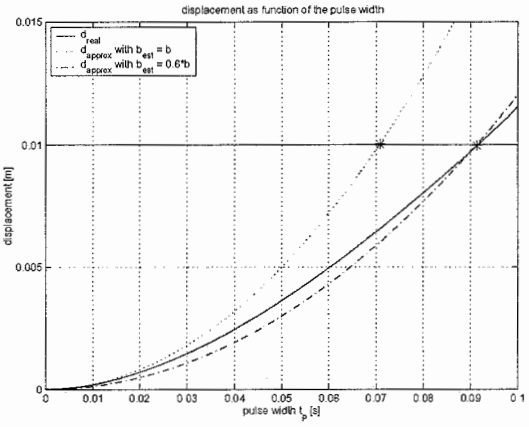

Figure 1: Displacement d as function of the pulse width t_p

Table 1: Parameter values

m	150 kg
f_c	100 N
c	700 kg/s
f_s	130 N
f_p	300 N
T	0.01 s

For a displacement of 0.01 m, Eq.(3) results in a pulse width of 0.092 s, while the approximate Eq.(4), results in a t_p of 0.070 s. In order to find the appropriate t_p of 0.092 s, the adaptation algorithm modifies $\hat{b}$ such that the dash-dotted line in figure 1, which corresponds to Eq.(4), coincides with the exact curve (solid line), for the specific desired displacement. For example, for a specific displacement of 0.01 m, the $\hat{b}$ which forces the dashed-dotted curve to coincide with Eq.(3) at 0.01 m has a value of $0.6b$.

To prevent the large fluctuations in the estimate of $\hat{b}$, due to large difference in displacements, t_p is constrained to a maximum. If a maximum for t_p=$t_{p,max}$ of 0.03 s is chosen, the estimate $\hat{b}$ is between 81% and 100% of that of the theoretical b. The lower percentage is a function of the parameters of the system.

A point to be noted is that, if the feedback controller results in the same error, every time this controller is used, there's no need for a $t_{p,max}$. The estimated $\hat{b}$ will not converge to the theoretical b, rather, the adaptation algorithm will estimate the $\hat{b}$ in such a way, that only one pulse is needed to move the system from its initial position to the desired position.

If for some reason the system isn't at the reference position after this first pulse, more

pulses have to be used. The pulse widths of these subsequent pulses result in the overshoot of the system response. If for example, the residual error is 2.0 mm, the dash-dotted line will correspond to a pulse width of 41 ms, while the real pulse width should be 36 ms. With the t_p calculated from the dash-dotted line, the true displacement is 2.5 mm instead of the desired 2.0 mm and thus overshoot occurs.

As the residual error decreases with subsequent pulses, the difference between the estimated and real displacement (e_0^0) is small resulting in minimal adaptation of $\hat{b}$.

For a finite $t_{p,max}$, the difference between $\hat{b}$ and b is smaller than without limits on the pulse width, resulting in a better estimation of the displacement. Thus, the adaptation algorithm estimates $\hat{b}$ more precisely, and overshoot is less likely to occur. A drawback of introducing a maximum pulse width is that the system takes longer to reach the final position.

3.2 Adaptive PAPWC

The idea behind adaptive PAPWC is that the pulse width resulting from Eq.(4) should be transformed to be equal to an integer time the sampling time (nT), so that it can be implemented in a discrete time system. Since the desired displacement remains the same with PAPWC as with PWC, altering the pulse width results in an adjustment of the pulse amplitude as well; pulse amplitude and pulse width are the only free design parameters for the pulse shape. This is the starting point for the modification, proposed in this section.

The pulse width needs to be rounded to the higher integer multiple of T, i.e. $T_x = nT \geq t_p$, so that the corresponding pulse amplitude is smaller than or equal to the maximum permitted pulse amplitude f_p.

The displacement in Eq.(4) is dependent on the sign of f_p. Note that $sign(f_p)$ can be replaced by $sign(e)$ without changing the outcome of Eq.(4). The desired displacement must be the same for both t_p and T_x, which results in:

$$\begin{aligned} d(t_p) &= d(T_x) \\ b\, t_p^2 sgn(e) &= b^*\, T_x^2 sgn(e) \\ \frac{f_p(f_p - f_c)}{2mf_c} t_p^2 sgn(e) &= \frac{f_p^*(f_p^* - f_c)}{2mf_c} T_x^2 sgn(e). \end{aligned} \tag{14}$$

The constant b must change to b^*, in order to satisfy Eq.(14). This can only be done by varying the pulse amplitude f_p^*, since this is the only free parameter in b^*. Solving Eq.(14) for f_p^* results in:

$$f_p^* = 0.5 f_c \pm 0.5 \sqrt{f_c^2 + 4 f_p (f_p - f_c) \frac{t_p^2}{T_x^2}}. \tag{15}$$

Instead of using t_p and f_p, the control pulse is specified by T_x and f_p^*, where $\pm$ should be replaced by $+$ for a positive pulse height. The above algorithm works, if the Coulomb friction coefficient f_c is known. But unfortunately that isn't the case.

In order to solve this problem, b is divided into two parameters:

$$
\begin{aligned}
b &= A_1 f_p^2 - A_2 f_p \qquad (16)\\
\text{with } A_1 &= \frac{1}{2mf_c} \text{ and } A_2 = \frac{1}{2m}\\
a &= \begin{bmatrix} A_1 & A_2 \end{bmatrix}^T
\end{aligned}
$$

From the estimates $\hat{A}_1$ and $\hat{A}_2$, an estimate of f_c can be derived by dividing $\hat{A}_2$ by $\hat{A}_1$. Using $\hat{f}_c$ in Eq.(15), the pulse amplitude is calculated. The next pulse width is calculated from Eq.(4), using the $\hat{b}$ obtained from $\hat{A}_1$ and $\hat{A}_2$.

In [7] an adaptation algorithm is proposed, that does not need to invert F. Now that there are two parameters to be estimated, this algorithm can save computation time. The algorithm is given by the equations:

$$
\begin{aligned}
\pi(k) &= P(k-1)u(k) \qquad (17)\\
P(k) &= \lambda^{-1}P(k-1) - \lambda^{-1}K(k)u^H(k)P(k-1) \qquad (18)\\
K(k) &= \frac{\pi(k)}{\lambda + u^H(k)\pi(k)} \qquad (19)\\
\xi &= d(k) - \hat{a}^H(k-1)u(k) \qquad (20)\\
\hat{a}(k) &= \hat{a}(k-1) + K(k)\xi^H(k) \qquad (21)\\
\\
u(k) &= \begin{bmatrix} f_p^2 t_p^2 & f_p t_p^2 \end{bmatrix}^T \qquad (22)
\end{aligned}
$$

The superscript H stands for the Hermitian transpose. In this case, all values are real and thus H can be replaced by T. u, d and ξ in Eq.(17)-(21) have the same interpretation as u_p, d and e_0^0 respectively, in Eq.(11)-(13). In [7] K is described as the time-varying gain vector and P as the inverse correlation matrix. The λ in the algorithm represents the forgetting factor. Generally the forgetting factor is selected to be smaller than and close to 1 if old data is to be 'forgotten' and is set to 1 if all data is to be equally weighted.

For the initiation of the adaptation algorithm, initial conditions for $P(0)$, $\hat{A}_1(0)$ and $\hat{A}_2(0)$ are required. $\hat{A}_1(0)$ and $\hat{A}_2(0)$ should be chosen as accurately as possible and the inverse covariance matrix $P(0)$ is set to $P(0) = \delta^{-1}I$. δ should be large if the sensors are noisy and can be small otherwise.

4 Numerical Simulations

For the system shown in Figure 2, with parameters listed in Table 1, the proposed algorithm is simulated and compared to the adaptive PWC. In Figure 2 the Coulomb friction and stiction are represented by f.

The forgetting factor is set to 1, so there is no forgetting of data. The initial value for $F^{-1}(0)$ and δ^{-1} are both $1e^{-5}$, since there's no noise present in the system. In order to

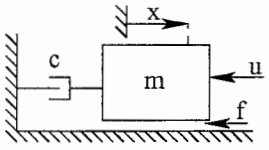

Figure 2: Model of the system

compare the two algorithms, the initial estimations for $\hat{b}(0)$ and $\hat{b}(0) = f\left(\hat{A}_1(0), \hat{A}_2(0)\right)$ are the same. This is done by calculating both $\hat{b}(0)$'s assuming that m is exactly known and $f_{c,\ est}$ is some percentage of the the real f_c. This provides a fair means of comparing $\hat{b}(0)$, $\hat{A}_1(0)$ and $\hat{A}_2(0)$. The initial value for $f_{c,\ est}$ is set to $0.8f_c$, so that $\hat{b}(0)$ is 1.375 b.

Figure 3 and 4 illustrate the evolution of the position errors for the algorithm proposed by [6] and from the adaptive PAPWC algorithm respectively. Both the figures present the error profiles for the first iteration and the 10th iteration respectively. Hereby, the value of $\hat{b}$ of the last simulation is set to be the initial estimate in the subsequent simulation.

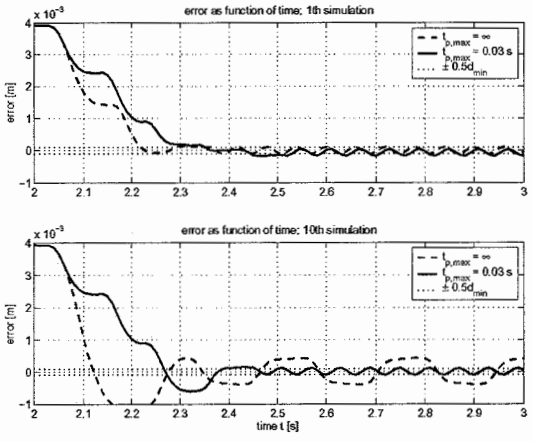

Figure 3: Error as a function of time using the original PWC approach

From Figure 3 which corresponds to adaptive PWC, it can be seen that for the case when $t_{p,max}$ is not bounded, overshoot occurs, as predicted. However, for the case when $t_{p,max}$ is bounded, overshoot continues to exist. This is caused by a minimum possible displacement due to a minimum pulse width of one sample time. The theoretical minimum displacement can be found using

$$d_{min}(T) = \frac{f_p T}{c} - \frac{m f_c}{c^2} ln \left[\frac{f_p}{f_c}\left(e^{cT/m} - 1\right) + 1\right]. \tag{23}$$

d_{min} is equal to the maximum final error. If the PWC isn't stopped after the error becomes smaller than d_{min}, a limit cycle around the reference position is the result.

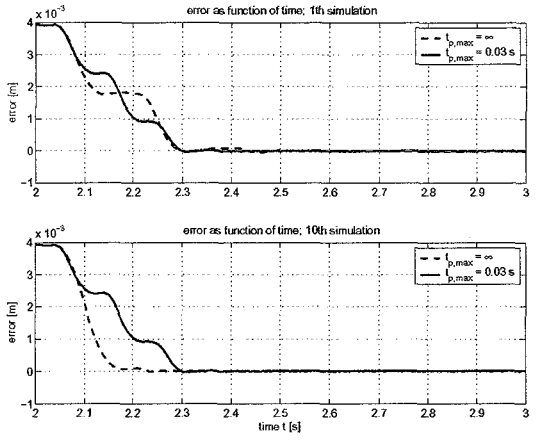

Figure 4: Error as a function of time using adaptive PAPWC

In the second plot of Figure 3, the overshoot for $t_{p,max} = \infty$ is larger than the displacement given by Eq.(23). This is caused by the fact that the pulse width now is $2T$, which is a result of the poor estimate for $\hat{b}$.

In Figure 4, one can see that the the final error is around zero. Due to Coulomb friction, fluctuations with small velocities around zero result in numerical problems in Simulink. To avoid this situation an error tolerance of $2e^{-5}m$ is introduced. If the error is smaller than this value, the control input is set to zero and the system will stay at rest. The final results thus have a maximum allowed value of $2e^{-5}m$.

The minimum displacement with adaptive PAPWC is given by:

$$d_{min}(T) = \frac{f_s T}{c} - \frac{m f_c}{c^2} ln \left[\frac{f_s}{f_c} \left(e^{cT/m} - 1 \right) + 1 \right], \tag{24}$$

and is caused by stiction. When the calculated pulse amplitude becomes smaller than the stiction, the mass simply doesn't move.

To compare the two presented minimum displacements, their values have been calculated using Table 1. PWC results in a d_{min} of $1.9e^{-4}m$, while PAPWC has a d_{min} of $1.3e^{-5}m$.

In the second plot of Figure 4 for $t_{p,max} = \infty$, one can note that theoretically only one pulse is needed to get the system very close to zero error. The final error from the first pulse will practically never be zero, since a second pulse which needs to move the system by a small amount will marginally modify $\hat{b}$ such that in the next iteration, the first pulse cannot satisfy the desired motion.

The pulse sequences for the adaptive PWC and PAPWC are presented in Figures 5 and 6 respectively. Figure 5 illustrates that the input f_p into the system contains only maximum positive and negative pulses. The switching between the positive and negative pulses is caused by the limit cycle.

Figure 6 reveals that the pulses die out after a short period. If a $t_{p,max}$ is used, only positive pulses drive the system. Without a bound on $t_{p,max}$, overshoot occurs, which

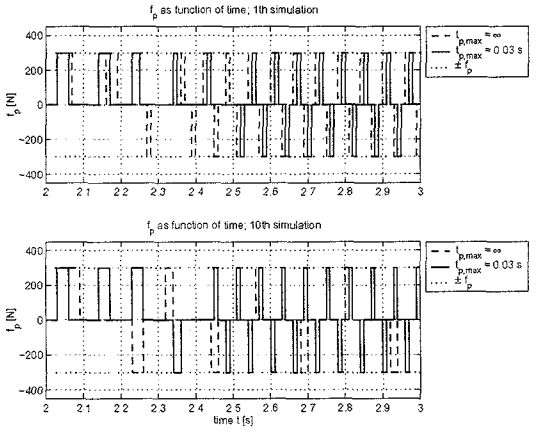

Figure 5: f_p As a function of time using the original PWC approach

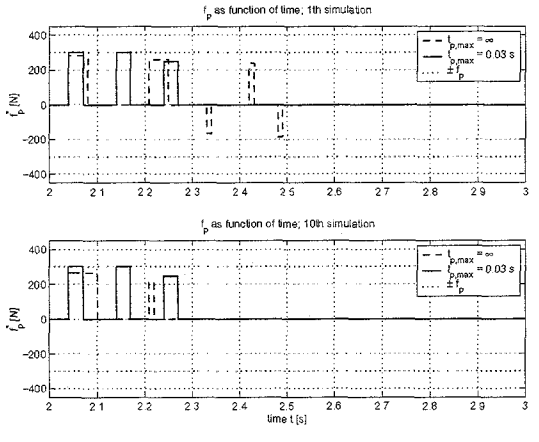

Figure 6: f_p As a function of time using adaptive PAPWC

results in subsequent negative pulses. After ten iterations, the unbounded pulse width control sequence, needs two pulses to reach the reference position, while the input with the bounded pulse width hasn't changed much.

The estimates for $\hat{b}$ are plotted in Figures 7 and 8.

Although both algorithms have used the same initial value for the inverse covariance matrix, it can be seen that the adaptation speed of the adaptive PAPWC (Figure 8) is much faster than that of the adaptive PWC (Figure 7). In the second plot of Figure 7, it can be seen that the estimate for $\hat{b}$ with bounded $t_{p,max}$ is closer to the theoretical b than for the unbounded $t_{p,max}$. This confirms earlier findings.

The estimate for $\hat{b}$ for the case that $t_{p,max} = \infty$ in the second plot of Figure 8 shows some unexpected results. The estimated value is larger than the theoretical one. Looking

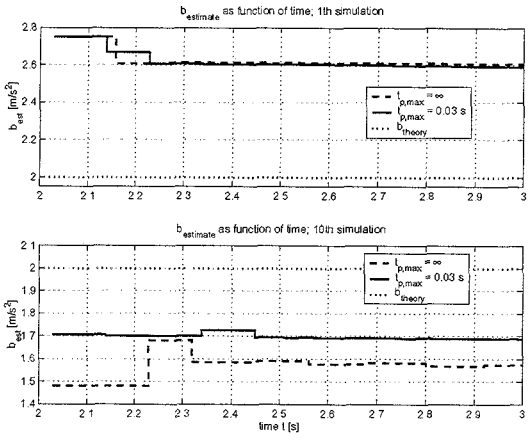

Figure 7: $\hat{b}$ As a function of time using the original PWC approach

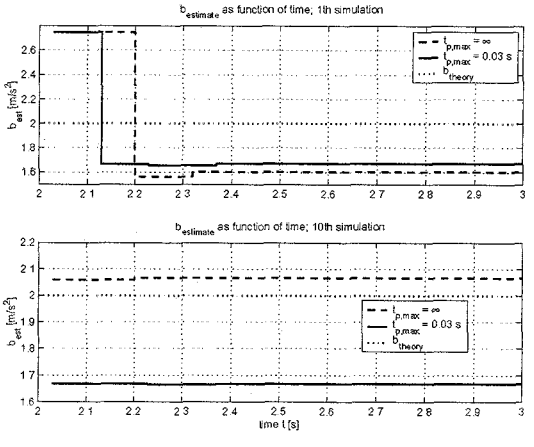

Figure 8: $\hat{b}$ As a function of time using PAPWC

at the simulation results, only one large pulse is driving the system. After this first pulse, only a small error in position is left. Small position errors result in a relatively small ξ. Looking at Eq.(21), a small ξ results in almost no update of the parameters.

Since there are two parameters to be estimated from one equation, only the ratio between the two parameters can be derived. Apparently the adaptation algorithm found values for $\hat{A}_1$ and $\hat{A}_2$ that result in an unexpected value for $\hat{b}$ that still result in the desired displacement. If the sampling time is decreased, the system returns to the situation where $\hat{b}$ is smaller than the theoretical b.

The final results are presented in Figure 9, which presents the estimated Coulomb friction.

From Eq.(16), one can conclude that $\hat{b}$ increases when $\hat{f}_c$ decreases and vice versa. The

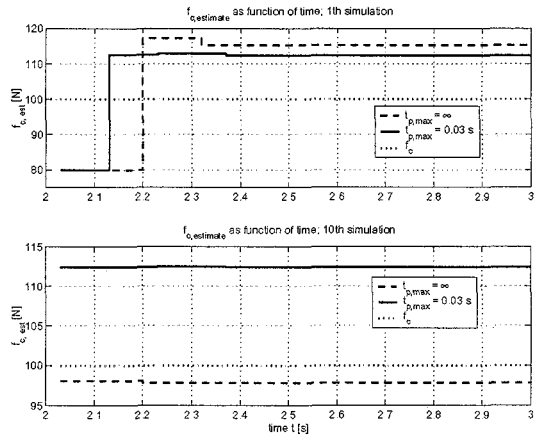

Figure 9: $\hat{f}_c$ As a function of time using adaptive PAPWC

results that are shown are thus to be expected, after seeing Figure 7 and 8.

When simulations are done *without* updating the initial value of the inverse covariance matrix after each simulation, the values for $\hat{b}$ and $\hat{f}_c$ at the tenth simulation are fluctuating more. This results in more overshoot in the error signal and more sign switching of the input pulses. This however does not influence the final error significantly.

5 Conclusions

This paper proposed a modification to the Pulse Width Control technique by forcing the input pulse width to be coincident with an integral multiple of the sampling period. This, in conjunction with adaptation of the pulse amplitude results in the Pulse Amplitude Pulse Width Controller. To account for uncertainties in the estimated system parameters, an adaptation algorithm is used to estimate the coefficient of friction which is subsequently used by the PAPWC.

Numerical simulations illustrate the significant reduction in the final position error. The adaptive PAPWC controller will be implemented on an experimental testbed in the future and extended to systems with low frequency resonance.

References

[1] C. A Coulomb. Theorie des machines implies, en ayant egard au frottement de leurs parties, et a la roideur dews cordages. *Mem. Math Phy.*, pages 161–342, 1785.

[2] C. Canudas de Wit, H. Olsson, K. J. Åström, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995.

[3] Y. Yoshida and M Tanaka. Position control of a flexible arm using a dither signal. *JSME International Journal*, Series C, 36(1):93–99, 1993.

[4] B. Friedland and Y-J. Park. On adaptive friction compensation. *IEEE Transactions on Automatic Control*, 37(10):1609–1612, October 1992.

[5] T-L. Liao and T-I. Chien. An exponentially stable adaptive friction compensator. *IEEE, Transactions on Automatic Control*, 45(5):977–980, May 2000.

[6] S. Yang and M. Tomizuka. Adaptive pulse width control for precise positioning under the influence of stiction and coulomb friction. *Journal of Dynamic Systems, Measurements, and Control*, Vol. 110:221–227, September 1988.

[7] S. Haykin. *Adaptive Filter Theory.* Prentice Hall, Inc., fourth edition, 2002.

Concluding Remarks

Let's begin with the concept behind adaptive Pulse Width Control. Adaptive PWC uses small pulses for precise positioning of a mass after a basic feedback controller has moved the system to the vicinity of the desired position. Using an estimated coefficient, a pulse is calculated which moves the system to the desired position. When the final error between the actual position and the desired position is too large, a new pulse is calculated and put on the system.

Adaptive PWC can be implemented on systems that can be represented as a single mass under influence of Coulomb and viscous friction. For more complex systems, PWC isn't useful anymore. However, the idea of using pulse control for precise positioning is!

Current study has found an algorithm for rest-to-rest positioning of a flexible arm system under influence of Coulomb and viscous friction, using multiple pulses. To calculate the pulse series, knowledge about the friction parameters is needed. Use of the adaptation algorithm suggested by adaptive PWC will result in an algorithm that is too complex for implementation.

This is where the energy-based algorithm can be useful. This algorithm estimates the friction parameters while a simple feedback controller moves the system from its initial position to the vicinity of the desired position. Using the estimates for the friction parameters, the multiple-pulse algorithm calculates the pulse series which moves the system to the desired position. If the error between actual position and the desired position is too large, a new series of pulses can be calculated which will drive the system closer to the desired position.

When a new desired position is introduced, a feedback controller takes over again and estimation of the friction parameters will continue.

The above shows the synergy between pulse control and energy-based friction estimation.