A Self-Tuning Controller

ร.อ.ปเสฏฐา สารลักษณ์*

Abstract- This article describes the design steps for one type of the adaptive controller, called a *self-tuning controller*. The adaptive control algorithms used are the least-square direct adaptive pole assignments with *constant* and *variable* forgetting factors. The use of a variable forgetting factor can avoid one of the major difficulties associated with constant forgetting factor-namely, *blowing-up* of the covariance matrix of the estimates and subsequent unstable control. Furthermore, *anti-reset windup* technique is incorporated into the adaptive regulators in order to avoid "summing up" of the integrator if the control signal saturates when there is a control error. Performance comparisons in controlling angular positions of the *time-varying-inertia system* with the adaptive and the non-adaptive (Proportional-Integral-Derivative or PID) controllers are implemented in both simulations and real-time experimentations. The adaptive control schemes both with *constant* and *variable* forgetting factors show better performance in controlling position of the time-varying-inertia system than the non-adaptive counterpart does. This is due to the abilities in adapting the controllers' parameters of both adaptive control schemes to the change in dynamic of the controlled system.

1. Introduction

Controllers can be designed following any one of a number of procedures available [1]. Generally, most control systems design procedures involve the following steps:

1) Derive a nonlinear or linear mathematical model based on the physical properties of the process.
2) Identify the parameters of the model from off-line experiments on the process.
3) Design the controller and establish its parameters based on the process model identified in the above steps.
4) Implement the controller.
5) Tune the controller by repeating steps 3 and 4 until satisfactory performance is achieved.

It is usually desirable to repeat the above procedures if the system dynamics or disturbance characteristics change significantly.

The classical *Proportional-Integral-Derivative* (*PID*) controller is widely used in industry. It is often necessary to adjust the controller according to the changing process requirement. In practice, tuning and readjustment of this type of controller is highly subjective because it depends upon practical experience and familiarity with the process of the individual involved. Values of the parameters for a fixed parameter controller are normally chosen on the basis of the best compromise for one operating point.

For the system to operate optimally as the operating point changes, it is desirable that the controller parameters be tuned to the operating conditions. Availability of the modern

---

*อาจารย์ กองวิชาวิศวกรรมเครื่องกล ส่วนการศึกษา โรงเรียนนายร้อยพระจุลจอมเกล้า

digital devices with on-board on-line computational capabilities make this possible using the following procedures:

1) Model the complicated nonlinear process by a low-order model with time-varying parameters.
2) On-line identification of the parameters of such a model will track the operating conditions.
3) The control algorithm can use the estimated (identified) values of the model parameters to generate the required control on-line.

An *adaptive* controller designed according to the above procedures is called a *self-tuning regulator or controller*.

The original self-tuning regulator was designed to operate on a process with constant but unknown parameters [2]; the process parameters are estimated on-line by recursive least-square estimation (RLS) and these estimates are then used in designing a minimum variance controller. As the parameter converges, better control is achieved and it has been shown that the optimal control law will result even in the case of convergence to biased estimates of the process parameters. The above algorithm is called an *indirect adaptive algorithm* because the regulator parameters are not updated directly, but rather indirectly via the estimation of the process model. It is, however, sometimes possible to reparameterize the process so that the model can be expressed in terms of the regulator parameters. This gives a significant simplification of the algorithm, because the design calculations are eliminated and the regulator parameters are updated directly. This give a *direct adaptive algorithm*. Analysis of the asymptotic properties of a direct self-tuning regulator was also first given in 1973 by Åström and Wittenmark [2].

However, typical processes exhibit some degree of time varying and nonlinear dynamics; these may arise, for example, by nonlinearities in actuators and other gradual drifts of the system, all of which violate the assumption of linear, time invariant system. An example of a nonlinear system is a robot shown in Figure 1.

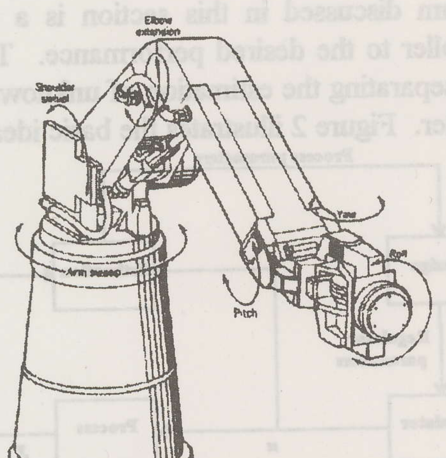

Figure 1. An industrial robot.

The motor turning at the elbow moves the wrist through the forearm, which has some flexibility. Variations in the moment of inertia are also common in current industrial robots. The moment of inertia usually depends on the geometry of the robot and on the load it handles. With variations in these two variables during the operation, the moment of inertia may change significantly from one position to another. This makes it necessary to incorporate some adaptive mechanism in the estimator to make it able to adjust the estimates to follow these changes. One way to solve this problem is to prevent the recursive estimator from converging by introducing a forgetting factor [3] and many applications show that this algorithm behaves well for wide range of plants. Some other methods are reviewed by Goodwin and Payne [4].

Nevertheless, a problem often referred to as an *estimator windup* can arise when the algorithm with *constant* forgetting factor is used in steady-state regulation of the plant. Old information is continually forgotten while there is very little new dynamic information coming from the plant. If the forgetting factor is not carefully chosen, this may lend to an exponential growth of the covariance matrix and a system which is extremely sensitive to disturbances and susceptible to numerical and computational difficulties. One way to solve this problem is to vary the forgetting factor at each step such that a *measure* for the information content in the estimator is kept constant [5]. With a reasonable choice of information measure, it was shown that the algorithm with *variable* forgetting factor could prevent the covariance matrix from blowing up while still retaining the adaptability of the algorithm.

This article describes the design steps for the *direct adaptive control algorithm*. To demonstrate the behavior of the adaptive control scheme, a series of simulation results are given. And to evaluate and exhibit the features of the simulations done, real-time experimentation results are also given in this article.

2. Adaptive control algorithms

The adaptive control algorithm discussed in this section is a *self-tuning* type since it automatically tunes the controller to the desired performance. The *Self-tuning regulator* (*STR*) is based on the idea of separating the estimation of unknown parameters of the plant from the design of the controller. Figure 2 illustrates the basic idea of the STR [7].

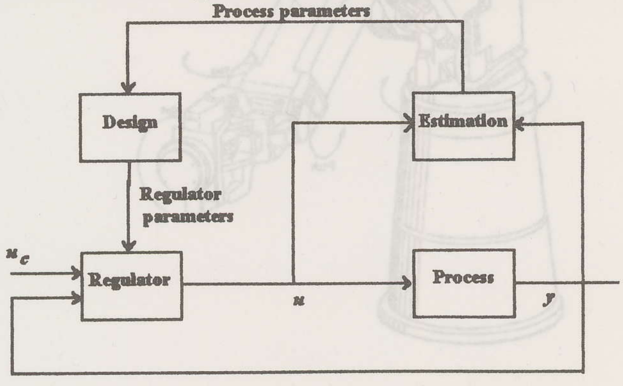

Figure 2. The self-tuning regulator (STR) block diagram.

The unknown parameters are estimated on-line, using recursive estimation schemes such as:

- Least square (LS)
- Extended and generalized least square
- Stochastic approximation
- Instrumental variable
- Maximum likelihood.

The *underlying design problem*, which is represented by the block *Design* in Figure 2, generates an on-line solution to the design problem for a system with known parameters. The design methods that can be used are:

- Pole placement
- Model following
- Minimum variance
- Linear quadratic.

The STR is very flexible with respect to choices of the design and estimation methods. Different combinations of the estimation and design methods lead to STR with different properties.

For the self-tuner shown in Figure 2, the regulator parameters are updated *indirectly* via design calculations. This gives an *indirect adaptive algorithm*. It is, however, often possible to reparameterize the plant model in the regulator parameters so that the regulator parameters are updated *directly*. This gives a *direct adaptive algorithm*. This direct approach results in a significant simplification of the algorithm because the design calculation is eliminated. The direct adaptive algorithm will be discussed in this section.

2.1 Process model

In the direct self-tuning regulator, the idea is to use the specifications, in terms of the desired locations of the poles and zeros, to reparameterize the process model so that the design step is trivial [17]. Consider a process described by the difference equation in the delay operator q^{-1}:

$$A\left(q^{-1}\right)y(t) = q^{-d}B\left(q^{-1}\right)u(t) \qquad (2.1)$$

where d = deg(A) - deg(B), is called the *pole excess* or the *prediction horizon*. It represents the number of time steps before the input u affects the output y. The specifications the process output is to mimic are usually given in the form of the desired closed-loop response:

$$A_m\left(q^{-1}\right)y(t) = q^{-d}t_0u_c(t) \qquad (2.2)$$

where A_m is the desired closed-loop characteristic polynomial. The degree of A_m is chosen such that:

$$\deg\left(A_m\right) \geq \deg(A) - \deg(B) \qquad (2.3)$$

A pole placement design, which gives the above desired response, is obtained by solving the equation:

$$A\left(q^{-1}\right)R\left(q^{-1}\right)+q^{-d}B\left(q^{-1}\right)S\left(q^{-1}\right)=A_o\left(q^{-1}\right)A_m\left(q^{-1}\right)B\left(q^{-1}\right) \qquad \textbf{(2.4)}$$

where A_o is the desired observer polynomial. The degree of A_o is chosen such that:

$$\deg\left(A_o\right)\geq 2\deg(A)-\deg\left(A_m\right)-\deg(B)-1 \qquad \textbf{(2.5)}$$

The observer polynomial should be stable and faster than the desired closed-loop response determined by A_m. In (2.4), $R(q^{-1})$ and $S(q^{-1})$ are the polynomials:

$$R\left(q^{-1}\right)=r_0+r_1q^{-1}+r_2q^{-2}+\ldots+r_kq^{-k}$$

$$S\left(q^{-1}\right)=s_0+s_1q^{-1}+s_2q^{-2}+\ldots+s_lq^{-l} \qquad \textbf{(2.6)}$$

of the control law:

$$R\left(q^{-1}\right)u(t)+S\left(q^{-1}\right)y(t)=t_0A_o\left(q^{-1}\right)u_c(t) \qquad \textbf{(2.7)}$$

The degrees of $R(q^{-1})$ and $S(q^{-1})$ are chosen such that:

$$\deg\left(R\left(q^{-1}\right)\right)=\deg\left(A_m\left(q^{-1}\right)\right)+\deg\left(A_o\left(q^{-1}\right)\right)-\deg\left(A\left(q^{-1}\right)\right)$$

$$\deg\left(S\left(q^{-1}\right)\right)=\deg\left(A\left(q^{-1}\right)\right)-1 \qquad \textbf{(2.8)}$$

To reparameterize the process model in term of the regulator parameters, multiply (2.4) by $y(t)$:

$$A\left(q^{-1}\right)R\left(q^{-1}\right)y(t)+q^{-d}B\left(q^{-1}\right)S\left(q^{-1}\right)y(t)=A_o\left(q^{-1}\right)A_m\left(q^{-1}\right)B\left(q^{-1}\right)y(t) \qquad \textbf{(2.9)}$$

Using (2.1) to eliminate $y(t)$ in the first term of (2.9):

$$q^{-d}B\left(q^{-1}\right)R\left(q^{-1}\right)u(t)+q^{-d}B\left(q^{-1}\right)S\left(q^{-1}\right)y(t)=A_o\left(q^{-1}\right)A_m\left(q^{-1}\right)B\left(q^{-1}\right)y(t) \qquad \textbf{(2.10)}$$

To simplify the design, it is assumed that the plant is *minimum phase* i.e. $B(q^{-1})$ is a stable polynomial (Note that this assumption is considered as one constraint in utilizing direct adaptive algorithm). Therefore, $B(q^{-1})$ can be canceled in (2.10), giving:

$$\begin{aligned}A_o\left(q^{-1}\right)A_m\left(q^{-1}\right)y(t)&=q^{-d}R\left(q^{-1}\right)u(t)+q^{-d}S\left(q^{-1}\right)y(t)\\&=R\left(q^{-1}\right)u(t-d)+S\left(q^{-1}\right)y(t-d)\end{aligned} \qquad \textbf{(2.11)}$$

or in the a *regression model* form:

$$y(t)=\varphi_f^T(t-d)\theta \qquad \textbf{(2.12)}$$

where

$$\varphi_f^T = \frac{1}{A_o\left(q^{-1}\right)A_m\left(q^{-1}\right)}\left[u(t-d)\ldots u(t-d-k)\ \ y(t-d)\ldots y(t-d-l)\right]$$

$$= \left[u_f(t-d)\ldots u_f(t-d-k)\ \ y_f(t-d)\ldots y_f(t-d-l)\right] \quad (2.13)$$

$$\theta^T = \left[r_0 r_1 r_2 \ldots r_k\ s_0 s_1 s_2 \ldots s_l\right]$$

φ_f is a filtered *regressor vector*, whose components are filtered by the polynomial $A_o(q^{-1})A_m(q^{-1})$. Note that (2.12) is linear in the parameters $r_o, \ldots, r_k, s_o, \ldots, s_l$.

2.2 Parameter estimation

The parameters in θ are estimated using the least square (LS) typed estimator. Since typical processes exhibits some degree of time-varying, and nonlinear dynamic behaviors, it is necessary to incorporate some adaptive mechanism in the estimator to make it able to adjust the estimates to follow these changes. A *forgetting factor* is introduced into the estimator to solve the problem. The forgetting factor prevents the recursive estimator from converging by exponentially weighting data coming into the estimator. The most recent data is given the highest weight and older data are weighted exponentially. Therefore, the method is called *exponential forgetting*.

The LS estimator with *constant* exponential weighting of past data is given by the following recursive relationships [7, 12]:

$$\varepsilon(t) = y(t) - \varphi_f^T(t-1)\hat{\theta}(t-1)$$

$$K(t) = \frac{P(t-1)\varphi_f(t-1)}{\left(\lambda \mathrm{I} + \varphi_f^T P(t-1)\varphi_f(t-1)\right)}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$$

$$P(t) = \frac{\left(\mathrm{I} - K(t)\varphi_f^T(t-1)\right)P(t-1)}{\lambda} \quad (2.14)$$

where

ε ≡ prediction error of $y(t)$

K ≡ *Kalman gain*

P ≡ estimated error *covariance matrix*

λ ≡ forgetting factor; $0 < \lambda \le 1$

$\hat{\theta}^T$ ≡ estimated parameter vector

$= \left[\hat{r}_0\, \hat{r}_1\, \hat{r}_2 \ldots \hat{r}_k \quad \hat{s}_0\, \hat{s}_1\, \hat{s}_2 \ldots \hat{s}_l\right]$

In (2.14), the prediction horizon d is assumed to be equal to 1. Note that the term *constant forgetting factor* comes from the fact that λ is a constant value. The forgetting factor λ enables the algorithm to adjust its parameters to a time-varying plant by preventing the covariance matrix $P(t)$ from being "too small". The speed of adaptation is determined by the *asymptotic memory length* [5, 18]:

$$N = \frac{1}{(1-\lambda)} \qquad (2.15)$$

which implies that the information coming into the estimator dies away with time constant N sample intervals.

When the algorithm is started up and the estimates are poor, the regulator will make large excursions until the estimates improve. This is observed as a quick decrease in $P(t)$; however, as control gets better $P(t)$ may become very large, especially during long periods of near-steady-state operation with little or no information about the system dynamics. This *blowing up* phenomena of $P(t)$ matrix can also be understood from the updating equation of $P(t)$ from (2.14):

$$P(t) = \frac{\left(I - K(t)\varphi_f^T(t-1)\right)P(t-1)}{\lambda} \qquad (2.16)$$

The negative term on the right-hand side represents the *reduction in uncertainty* of the estimates due to the last measurement. If no information is in the last measurement, i.e. steady-state condition, $P(t)\varphi_f(t)$ will not change direction and (2.16) reduces to:

$$P(t) = \frac{P(t-1)}{\lambda} \qquad (2.17)$$

As $P(t)$ is constantly divided by a forgetting factor λ, which is less than 1, the $P(t)$ matrix will "blow up" exponentially. This is called *estimator windup*. The estimator will forget the proper value of the estimates. As a result, the regulator becomes very sensitive to disturbances or numerical errors. And often a set point change or a random input will lead to a temporary unstable system or worse if the numerical errors or nonlinearities are very serious.

In order to avoid such difficulties, the LS estimator with *variable* forgetting factor is introduced [5, 18]. The idea is to define a *measure* for the information content in the estimator; the forgetting factor can then be varied at each step such that this measure of information is kept constant. The *weighted sum of the squares of the prediction error* Σ is defined as a measure of the information content of the estimator. Recursively, this can be expressed as:

$$\Sigma(t) = \lambda(t)\Sigma(t-1) + \left[1 - \varphi_f^T(t-1)K(t)\right]\varepsilon(t)^2 \qquad (2.18)$$

If $\Sigma(t)$ is kept constant:

$$\Sigma(t) = \Sigma(t-1) = \Sigma(t-2) = \ \ldots\ = \Sigma_0, \qquad (2.19)$$

i.e., the amount of forgetting at each step is kept equal to the amount of new information in the latest measurement (therefore, the estimation will be always based on the same amount of information), from (2.18) we get:

$$\lambda(t) = 1 - \frac{1}{N(t)} \tag{2.20}$$

where

$$N(t) = \frac{\Sigma_0}{\left[1 - \varphi_f^T(t-1)K(t)\right]\varepsilon(t)^2} \tag{2.21}$$

N(t) is the *equivalent asymptotic memory length.* Σ_o influences the velocity of adaptation of the controller in reaction to abrupt changes. An excessively high value of Σ_o leads to a longer adaptation time. Too small a value of Σ_o may lead to a blowing-up of the covariance matrix *P(t)* and corresponding unstable system.

The LS estimator with *variable* exponential weighting of past data is given by the recursive relationships [5, 18]:

$$\varepsilon(t) = y(t) - \varphi_f^T(t-1)\hat{\theta}(t-1)$$

$$K(t) = \frac{P(t-1)\varphi_f(t-1)}{\left(1 + \varphi_f^T P(t-1)\varphi_f(t-1)\right)}$$

$$\lambda(t) = \begin{cases} 1 - \dfrac{\left[1 - \varphi_f^T(t-1)K(t)\right]\varepsilon(t)^2}{\Sigma_0} & ;if \quad \lambda(t) > \lambda_{\min} \\ \lambda_{\min} \quad ;if \quad \lambda(t) \le \lambda_{\min} \end{cases}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$$

$$P(t) = \frac{\left(I - K(t)\varphi_f^T(t-1)\right)P(t-1)}{\lambda(t)} \tag{2.22}$$

In (2.22), the prediction horizon d is assumed to be 1. Note that a lower limit on the forgetting factor λ_{min} is introduced in order to prevent λ from becoming too small or even negative.

Another modification is introduced into both of the above adaptive algorithms in order to increase the robustness of the algorithms against *drift* in estimated parameters due to measurement noise in the absence of persistent excitation on the input signal. Reconsider the updating equation of the estimated parameters:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \varphi_f^T(t-1)\hat{\theta}(t-1)\right) \tag{2.23}$$

If the output of the plant *y(t)* is corrupted with measurement noise ξ for all t, i.e.:

$$y(t) = \varphi_f^T(t-1)\theta^* + \xi(t) \tag{2.24}$$

where θ^* represents the *true* parameter vector, therefore, $\hat{\theta}$ is not going to be able to converge to the true value θ^*. During this time, the absence of persistent excitation condition on the input signal can cause θ to drift away from an equilibrium. The gains will increase and the system will finally become unstable. The drifting phenomena can also be understood from (2.23). Using (2.24), (2.23) can be converted to:

$$\tilde{\theta}(t) = \left(I - K(t)\varphi_f^T(t-1)\right)\tilde{\theta}(t-1) - K(t)\varphi_f^T(t-1)\xi(t) \qquad (2.25)$$

where $\tilde{\theta} = \theta^* - \hat{\theta} \equiv$ parameter error. In the absence of persistent excitation on the input signal, the integral action can occur in (2.25). In other words, the parameter error growth causes instability in the control system.

An algorithm modification to (2.23) is to weaken this integration effect by converting it to a *leaky* integration. Therefore, this modification is called *leakage* [12]. (2.23) is modified to:

$$\hat{\theta}(t) = \left(1 - \lambda^*\right)\hat{\theta}(t-1) + K(t)\left(y(t) - \varphi_f^T(t-1)\hat{\theta}(t-1)\right) \qquad (2.26)$$

where $1 > \lambda^* > 0$. This converts (2.25) to:

$$\tilde{\theta}(t) = \left(\left(1 - \lambda^*\right)I - K(t)\varphi_f^T(t-1)\right)\tilde{\theta}(t-1) - K(t)\varphi_f^T(t-1)\xi(t) + \lambda^*\theta^* \qquad (2.27)$$

If φ_f is bounded, (2.27) implies that parameter errors (thus parameter estimates) remain bounded when $1 > \lambda^* > 0$.

2.3 Adaptive control law:

An adaptive control law can now be formulated as follows:

1) Update the parameter estimates by (2.14) or (2.22).
2) Determine a control law such that:

$$R\left(q^{-1}\right)u(t) = t_0 A_o\left(q^{-1}\right)u_c(t) - S\left(q^{-1}\right)y(t) \qquad (2.28)$$

3) Repeat the above steps at each sampling instant.

However, an additional modification to the regulator (2.28) is necessary because it may contain an unstable mode. An integral action embedded in the adaptive regulator is an unstable system; sometimes the integrator can assume very large value if the control signal u saturates while there is a control error. This is called *integral or reset windup*. To solve *integral windup* problem [7], (2.28) is rewritten by adding $A_{oc}(q^{-1})u(t)$ on both sides:

$$A_{oc}\left(q^{-1}\right)u(t) = t_0 A_o\left(q^{-1}\right)u_c(t) - S\left(q^{-1}\right)y(t) + \left(A_{oc}\left(q^{-1}\right) - R\left(q^{-1}\right)\right)u(t) \qquad (2.29)$$

An *anti-reset windup* regulator is then given by:

$$A_{oc}\left(q^{-1}\right)v(t)=t_0A_o\left(q^{-1}\right)u_c(t)-S\left(q^{-1}\right)y(t)+\left(A_{oc}\left(q^{-1}\right)-R\left(q^{-1}\right)\right)u(t)$$

$$u(t)=sat(v(t)) \tag{2.30}$$

where $A_{oc}(q^{-1})$ is the stable observer polynomial of the controller and $sat(.)$ is the saturation function of the control signal $u(t)$. Note that the regulator (2.30) is equivalent to (2.28) when the control signal does not saturate. Figure 3 shows block diagram describing the regulator (2.30) [7].

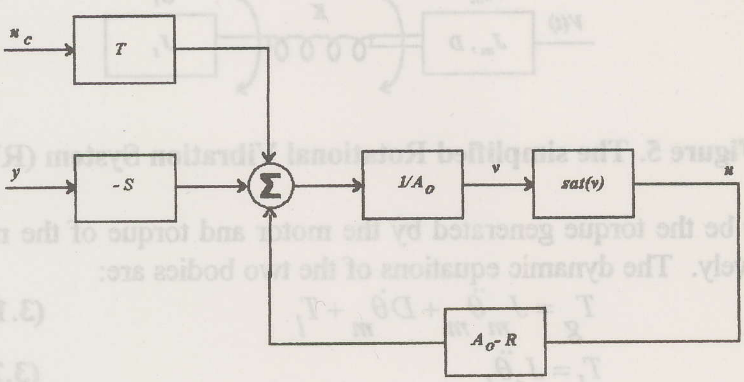

Figure 3. Block diagram of equation (2.30), which avoids integral windup.

3. Modeling

In this section, a model of the plant to be controlled is derived. Figure 4 gives a general view of the rotational vibration system (RVS) with time-varying inertia.

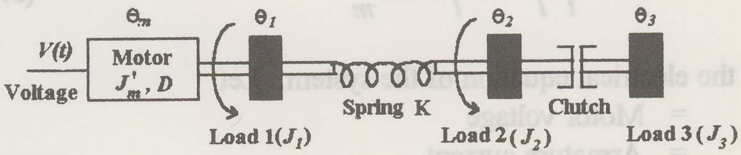

Figure 4. The Rotational Vibration System (RVS) with Time-Varying Inertia.

A DC motor is coupled to an inertial load (load 1) through a shaft. Load 1 is then coupled to another inertial load (load 2) through a spring. Finally, load 2 and the other inertial load (load 3) are coupled together through an electrical clutch. The electrical clutch engages or disengages load 3 to the rest of the system if activated or deactivated, respectively. As a result, the rotational inertia of the system when load 3 is engaged is different from that when load 3 is disengaged. Therefore, the dynamic of the controlled system is changed when load 3 is engaged. The system in Figure 4 can represent a robot shown in Figure 1.

3.1 Derivation of the mathematical model of the controlled system

In order to derive the dynamic equation for the controlled system, denote the moments of inertia and angular positions of the motor, load 1, load 2 , and load 3 by J'_m, J_1, J_2, J_3

and θ_m, θ_1, θ_2, θ_3, respectively. Also, denote the damping factors of the motor and the stiffness factor of the spring by D and K, respectively.

To simplify the derivation, assuming the inertia of the shafts connecting J'_m to J_1 and J_2 to J_3 are negligible and theirs stiffness factors are so high that $\theta_m = \theta_1$ and $\theta_2 = \theta_3$ (when J_3 is engaged). As a result of the above simplification, denote $\theta_l = \theta_2 = \theta_3$, which is the controlled variable. And denote $J_m = J'_m + J_1$. Also, denote $J_t = J_2 + J_3$; therefore J_t is equal to J_2 if J_3 is not engaged and is equal to $J_2 + J_3$ if J_3 is engaged. Figure 5 shows the resulting simplified controlled system.

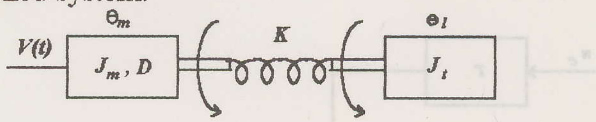

Figure 5. The simplified Rotational Vibration System (RVS).

Let T_g and T_l be the torque generated by the motor and torque of the motor shaft acting on J_t, respectively. The dynamic equations of the two bodies are:

$$T_g = J_m \ddot{\theta}_m + D\dot{\theta}_m + T_l \tag{3.1}$$

$$T_l = J_t \ddot{\theta}_l \tag{3.2}$$

The deflection of the spring is described by the equation:

$$T_l = K(\theta_m - \theta_l) \tag{3.3}$$

Substitute (3.2) for T_l in (3.1) and (3.3), the dynamic equations become:

$$T_g = J_m \ddot{\theta}_m + D\dot{\theta}_m + J_t \ddot{\theta}_l \tag{3.4}$$

$$J_t \ddot{\theta}_l + K\theta_l = K\theta_m \tag{3.5}$$

Now consider the electrical equation of the system. Let

V = Motor voltage
I = Armature current
L_a = Motor inductance
K_T = Motor torque constant
K_E = Back emf constant
R = Winding resistance

The motor equivalent circuit is approximated by a circuit shown in Figure 6. E_g is an internally generated voltage, which is proportional to the motor velocity.

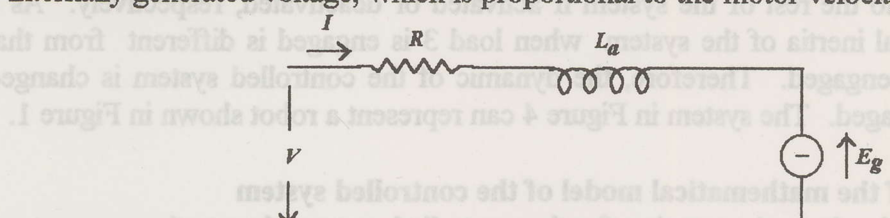

Figure 6. The motor equivalent circuit.

From the circuit above, it can be found that :

$$V = L_a \dot{I} + RI + E_g \qquad (3.6)$$

Since E_g is proportional to the motor velocity:

$$E_g = K_E \dot{\theta}_m \qquad (3.7)$$

Therefore, the electrical equation of the system becomes:

$$V = L_a \dot{I} + RI + K_E \dot{\theta}_m \qquad (3.8)$$

Assuming the magnetic field in the motor is constant, the current I produces a proportional torque T_g:

$$T_g = K_T I \qquad (3.9)$$

In order to simplify the analysis, substitute (3.9) in (3.4):

$$K_T I = J_m \ddot{\theta}_m + D\dot{\theta}_m + J_t \ddot{\theta}_l \qquad (3.10)$$

Now apply the Laplace transformation to equations (3.5), (3.8), and (3.10), which describe the behavior of the system:

$$s^2 J_t \theta_l(s) + K\theta_l(s) = K\theta_m(s) \qquad (3.11)$$

$$V(s) = sL_a I(s) + RI(s) + sK_E \theta_m(s) \qquad (3.12)$$

$$K_T I(s) = s^2 J_m \theta_m(s) + sD\theta_m(s) + s^2 J_t \theta_l(s) \qquad (3.13)$$

After rearranging the terms in equations (3.11),(3.12), and (3.13), this system of equations can be written in matrix form as below:

$$\begin{bmatrix} 0 & -K & \left(s^2 J_t + K\right) \\ \left(sL_a + R\right) & sK_E & 0 \\ -K_T & \left(s^2 J_m + sD\right) & s^2 J_t \end{bmatrix} \begin{bmatrix} I(s) \\ \theta_m(s) \\ \theta_l(s) \end{bmatrix} = \begin{bmatrix} 0 \\ V(s) \\ 0 \end{bmatrix} \qquad (3.14)$$

Let

$$[H(s)] = \begin{bmatrix} 0 & -K & \left(s^2 J_t + K\right) \\ \left(sL_a + R\right) & sK_E & 0 \\ -K_T & \left(s^2 J_m + sD\right) & s^2 J_t \end{bmatrix}$$

$$\{Y(s)\} = \begin{bmatrix} I(s) \\ \theta_m(s) \\ \theta_l(s) \end{bmatrix}$$

$$\{U(s)\} = \begin{bmatrix} 0 \\ V(s) \\ 0 \end{bmatrix}$$

Rewrite the matrix equation (3.14) as:

$$[H(s)]\cdot\{Y(s)\} = \{U(s)\} \qquad (3.15)$$

taking $[H(s)]^{-1}$ on both side of equation (3.15):

$$\{Y(s)\} = [H(s)]^{-1}\{U(s)\} \qquad (3.16)$$

The transfer function relating the load shaft position θ_l to the input voltage V can be derived from the matrix equation (3.16) as:

$$G(s) = \frac{\theta_l(s)}{V(s)} = \frac{KK_T}{p_5 s^5 + p_4 s^4 + p_3 s^3 + p_2 s^2 + p_1 s} \qquad (3.17)$$

where

$$p_5 = J_t L_a J_m$$

$$p_4 = J_t\left(DL_a + J_m R\right)$$

$$p_3 = DJ_t R + K_E J_t K_T + KL_a\left(J_t + J_m\right)$$

$$p_2 = K\left(DL_a + R\left(J_t + J_m\right)\right)$$

$$p_1 = K\left(DR + K_E K_T\right)$$

Values for the constant parameters in (3.17) can be obtained from the DC motor data sheet and from some rudimentary measurements and calculations.

To be able to use the model (3.17) in adaptive control algorithm design described in the previous section, the s-domain transfer function (3.17) is converted into the z-domain (discrete) transfer function of the form:

$$G(z) = \frac{b_1 z^4 + b_2 z^3 + b_3 z^2 + b_4 z + b_5}{a_0 z^5 + a_1 z^4 + a_2 z^3 + a_3 z^2 + a_4 z + a_5}; \qquad (3.18)$$

or in the delay operator q^{-1} form:

$$G\left(q^{-1}\right) = \frac{b_1 + b_2 q^{-1} + b_3 q^{-2} + b_4 q^{-3} + b_5 q^{-4}}{a_0 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3} + a_4 q^{-4} + a_5 q^{-5}} \qquad (3.19)$$

where the values for b_i's and a_i's when Load J_3 engages are different from those when Load J_3 disengages.

4. Simulation

To investigate the behavior of the adaptive scheme discussed in section 2, a series of simulation are performed. Using equation (2.8), with no observer dynamic ($A_o(q^{-1})=1$), the degrees of parameter polynomials $R(q^{-1})$ and $S(q^{-1})$ are both selected to be degree $k=l=4$. Therefore, there are totally 10 controller parameters to be estimated. The *direct* adaptive algorithms with *constant* and *variable* forgetting factor are implemented in the simulation as follows, respectively:

4.1 Direct adaptive with constant forgetting

1) Use the model of the plant to calculate $y(t)$ using $u(t-1)$ as input.
2) Use $y(t-1)$ and $u(t-1)$ to update the regressor $\varphi(t-1)$.
3) Update the current estimates $\theta(t)$ using (2.14).
4) Update the current covariance matrix $P(t)$ using (2.14).
5) Use the new estimates to calculate the control signal $v(t)$ and the saturated output $u(t)$ using (2.30).
6) Increase time step.
7) Go to step 1.

4.2 Direct adaptive with variable forgetting

1) Use the model of the plant to calculate $y(t)$ using $u(t-1)$ as input.
2) Use $y(t-1)$ and $u(t-1)$ to update the regressor $\varphi(t-1)$.
3) Update the current estimates $\theta(t)$ using (2.22).
4) Update the current forgetting factor $\lambda(t)$ using (2.22).
5) Update the current covariance matrix $P(t)$ using (2.22).
6) Use the new estimates to calculate the control signal $v(t)$ and the saturated output $u(t)$ using (2.30).
7) Increase time step.
8) Go to step 1.

Note that algorithm in 4.2 is identical to 4.1 apart from step 4 in 4.2, where the forgetting factor $\lambda(t)$ is chosen.

The initial values of the parameter estimates are chosen for both adaptive schemes such that the initial controllers are proportional controllers with low gain (0.94). The initial value for the covariance matrix is chosen to be 100 times an identity matrix. Also, a perturbation signal, which is a sine wave of magnitude 0.25 and frequency 2.0 *rad/sec*, is added to the control signal $u(t)$ of both adaptive schemes. This extra signal is used to provide the additional energy content to the controllers' outputs so that they are persistently exciting within the dominating dynamics of the process. Note that the values for the magnitude and frequency of the perturbation signal are chosen based on the satisfactory *trial-and-error* simulation results.

4.3 Simulation results

The plant is required to follow a square wave with magnitude 10 and frequency of 0.005 *Hz*. The load 3 (J_3) is engaged at t = 100 seconds. The results are shown in Figure 7a, 7b, and 7c, which show the responses along with control signals of three control schemes which are the adaptive scheme with constant forgetting factor, the adaptive scheme with variable forgetting factor, and the non-adaptive scheme (Proportional-Integral-Derivative or PID), respectively.

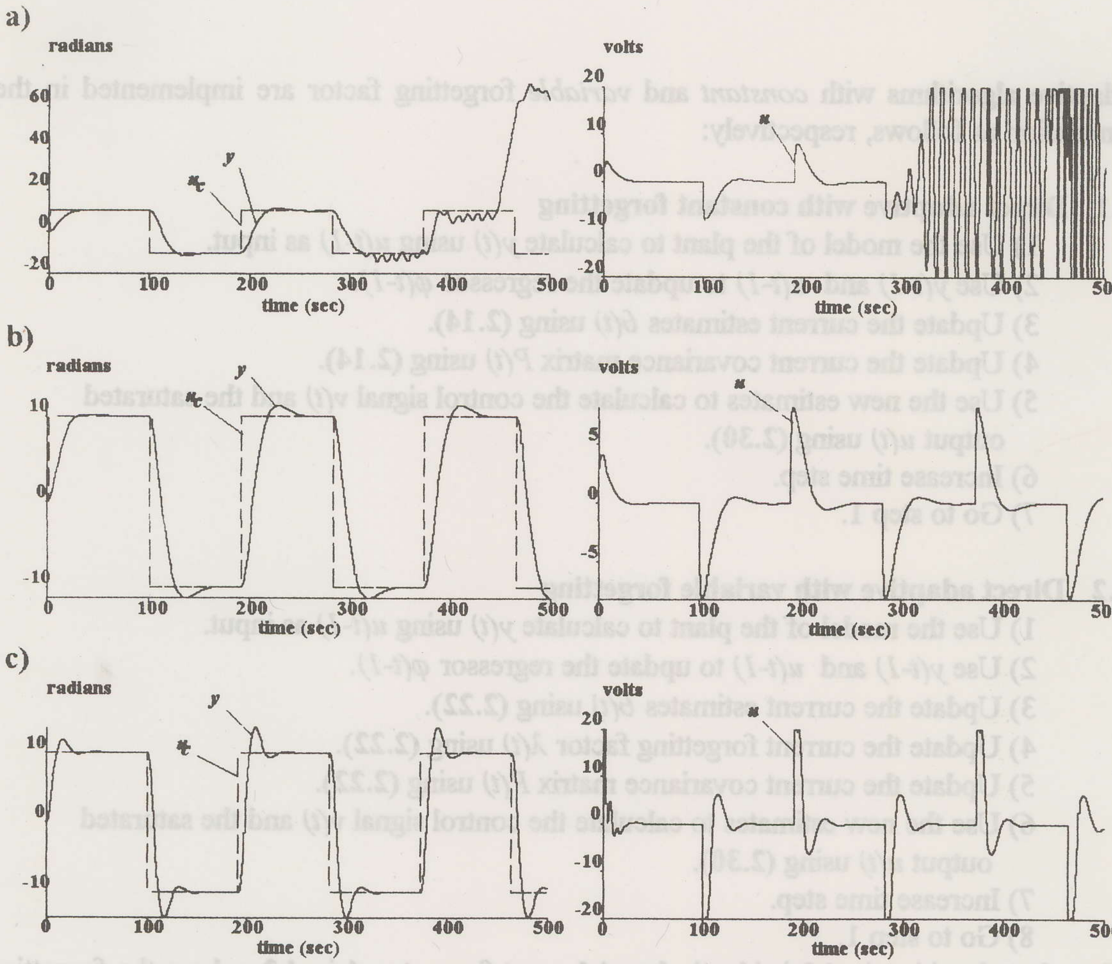

Figure 7. System responses with control signals of three control algorithms:
a) Adaptive with constant forgetting factor
b) Adaptive with variable forgetting factor
c) Non-adaptive (Proportional-Integral-Derivative or PID)
with J_3 is engaged at t = 100 seconds.

Note a *burst* in the output of the plant controlled by the adaptive controller with constant forgetting factor after approximately 300 *s.* of stable operation in Figure 7a. This is due to the *estimator windup* effect, as described in section 2. Figure 8 shows the diagonal elements p_{ii} of the covariance matrix $P(t)$ when the adaptive scheme with constant λ is used.

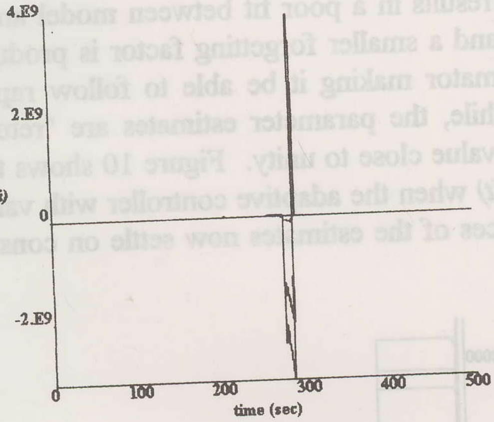

Figure 8. The diagonal elements of the covariance matrix *P* when the adaptive algorithm with *constant* forgetting factor (λ= 0.95) is used.

From the figure above, note the *blowing-up* of the covariance matrix of the estimates which leads to subsequent unstable control system. The burst in the output of the plant could be eliminated by trying another value of the forgetting factor.

Under control by the adaptive controller with variable forgetting factor, the behavior of the plant is more robust to the change in plant dynamic due to the engagement of J_3. The engagement of J_3 increases the overshoot of the response by small amount. See Figure 7b. Figure 9a and 9b show the behaviors of the forgetting factor and the prediction error of the adaptive controller with variable λ, respectively.

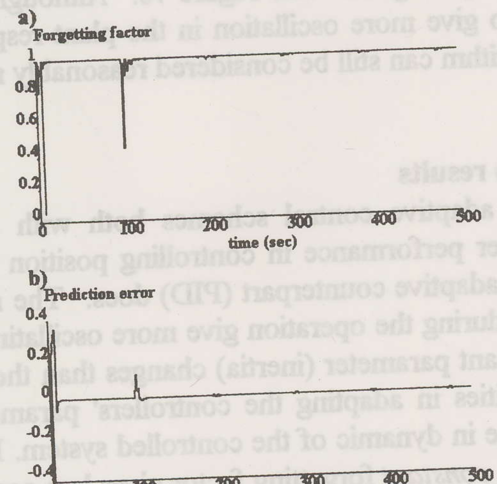

Figure 9. The behaviors of a) the forgetting factor and b) the prediction error of the adaptive controller with *variable* forgetting factor.

Under steady-state operations (prediction error is close to zero), $\lambda(t)$ is close to unity and the estimator behaves very much like an unweighted filter. This makes the estimates able to follow slow changes in the plant dynamics. During the dynamic change when J_3

engages at t = 100 *s*., this results in a poor fit between model and plant (prediction error bounces away from zero) and a smaller forgetting factor is produced. This increases the adaptation rate of the estimator making it be able to follow rapid changes in the plant dynamics. And after a while, the parameter estimates are "retuned" and the forgetting factor returns to its former value close to unity. Figure 10 shows the diagonal elements p_{ii} of the covariance matrix *P(t)* when the adaptive controller with variable forgetting factor is used. Note that the variances of the estimates now settle on constant values, and there is no estimator windup.

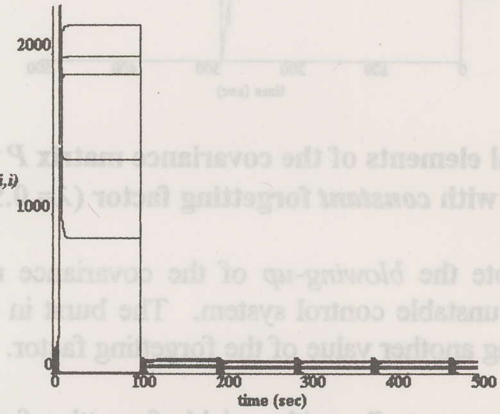

Figure 10. The diagonal elements of the covariance matrix *P* when the adaptive algorithm with *variable* forgetting factor is used.

Compared to the adaptive controller with variable forgetting factor, performance of the PID controller is considered quite good. See Figure 7c. Although the engagement of J_3 causes the PID controller to give more oscillation in the plant response than the adaptive scheme does, the PID algorithm can still be considered reasonably robust to the changes in plant dynamics.

4.4 Analysis of simulation results

From the simulations, the adaptive control schemes both with *constant* and *variable* forgetting factors show better performance in controlling position of the RVS with time varying inertia than the non-adaptive counterpart (PID) does. The non-adaptive controller whose parameters are fixed during the operation give more oscillating response around the command signal when the plant parameter (inertia) changes than the adaptive counterparts do. This is due to the abilities in adapting the controllers' parameters of both adaptive control schemes to the change in dynamic of the controlled system. However, although the adaptive control scheme with *constant* forgetting factor gives less oscillating response when there is a change in plant dynamic than the non-adaptive does, it could occasionally result in a *burst* in the plant output due to the *estimator windup*. The bursting phenomena could be eliminated if a "proper" value of the forgetting factor is used. And this fact leads to the development of the adaptive control scheme with *variable* forgetting factor. The adaptive control scheme with *variable* forgetting factor does not have the estimator windup problem because its forgetting factor is automatically adjusted to the "proper" value when the plant dynamic changes. Therefore, the adaptive scheme with *variable* forgetting factor is more robust to the change in the plant dynamic than the scheme with *constant* forgetting factor.

5. Experimentation

Figure 11 shows the schematic diagram of the experimental controlled system.

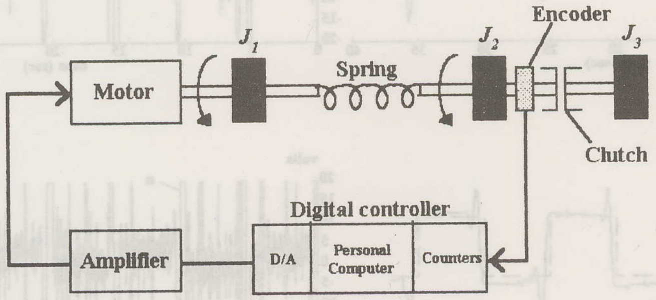

Figure 11. The schematic diagram of the experimental controlled system.

The experimental controlled system consists of three main subsystems which are:

1. Digital Controller: A personal computer equipped with multifunctional Input/Output board.
2. Position sensing subsystem: An incremental rotary optical encoder used to sense the position of the shaft.
3. Actuating subsystem: A 12-bit Digital-to-Analog Converter used to drive the servo motor through a power amplifier.

The other hardware component that is worth mentioned is the electrical clutch. This clutch engages J_3 to the rest of the system if activated and it disengages J_3 from the rest of the system if deactivated.

The adaptive (and non-adaptive) control algorithms discussed in section 2 and 4 are implemented in C programming language. For each control algorithm, the C program structure is implemented in a *modular* fashion. The initialization, Input/Output operations, and parameter estimations are implemented as separate functions called by the main program where every function routines are "glued" together. The advantage of implementing the program structure *modularly* is that it is more convenient in debugging during the program development stage because each function can be debugged and tested separately.

5.1 Experimentation results

The plant is required to follow a square wave with magnitude 90 and frequency of 0.1 *Hz*. The load 3 (J_3) is engaged at t = 15 seconds. The results are shown in Figure 12a, 12b, and 12c, which show the responses along with control signals of three control schemes which are the adaptive scheme with constant forgetting factor, the adaptive scheme with variable forgetting factor, and the non-adaptive scheme (Proportional-Integral-Derivative or PID), respectively.

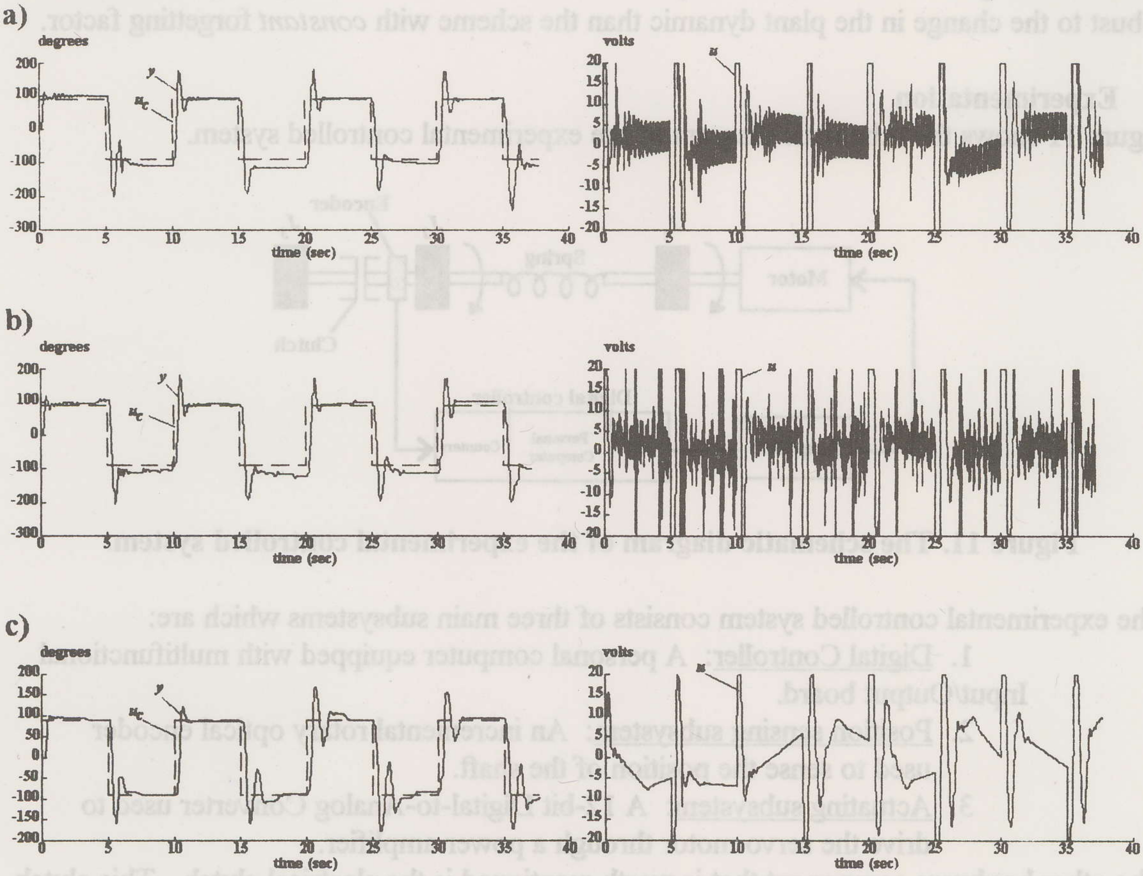

Figure 12. System responses with control signals of three control algorithms:
a) Adaptive with constant forgetting factor
b) Adaptive with variable forgetting factor
c) Non-adaptive (Proportional-Integral-Derivative or PID)
with J_3 is engaged at t = 15 seconds.

From the experimentation results above, it can be seen that the results obtained are generally support the facts found during the simulation. Both adaptive schemes are more robust to the change in the plant dynamic occurring at t = 15 *s.* than the non-adaptive scheme is. The behavior of the responses of both adaptive schemes are almost the same before and after J_3 engaging at t = 15 *s*. This fact substantiates the "well-known" characteristic of the adaptive control system in responding to the change in the plant parameters. But for the non-adaptive scheme, the behavior of its response is noticeably different before and after J_3 engaging at t = 15 *s*.

6. Conclusion

From the simulations, it is found that:

■ the adaptive control schemes both with *constant* and *variable* forgetting factors show better performance in controlling position of the RVS with time varying inertia than the non-adaptive counterpart (PID) does. This is due to the abilities in adapting the controllers' parameters of both adaptive control schemes to the change in dynamic of the controlled system.

■ the adaptive control scheme with *constant* forgetting factor could occasionally result in a *burst* in the plant output due to the *estimator windup*. The bursting phenomena could be eliminated if a "proper" value of the forgetting factor is used.

■ the adaptive control scheme with *variable* forgetting factor does not have the estimator windup problem because its forgetting factor is automatically adjusted to the "proper" value when the plant dynamic changes.

■ the adaptive scheme with *variable* forgetting factor is more robust to the change in the plant dynamic than the scheme with *constant* forgetting factor.

From the experimentation, it is found that:

■ the results obtained are generally support the facts found during the simulation.

References

[1] Shinskey, F. G. (1979). *Process Control Systems*, McGraw-Hill, New York.

[2] Åström, K. J. and B. Wittenmark (1973). " On self-tuning regulator ", *Automatica*, 9, 185.

[3] Åström, K. J., U. Borisson, L. Ljung and B. Wittenmark (1977). " Theory and application of self-tuning regulators ", *Automatica*, 13, 457.

[4] Goodwin, G. C. and R. L. Payne (1977). *Dynamic Systems Identification: Experiment Design and Data Analysis*, Academic Press, London.

[5] Oliveira, A. L., F. V. Coito, L. M. Silveira, J. M. Lemos and J. S. Marques (1986). " Microprocessor implementation of a self-tuning controller ", *IFAC Adaptive Systems in Control and Signal Processing*, 243.

[6] Rorabaugh, B. (1986). *Signal Processing Design Techniques*, TPR, Pennsylvania.

[7] Åström, K. J. and B. Wittenmark (1989). *Adaptive Control*, Addison Wesley, New York.

[8] Kulhavý, R. (1986). " Directional tracking of regression-type model parameters ", *IFAC Adaptive Systems in Control and Signal Processing*, 191.

[9] Bierman, G. J. (1977). *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York.

[10] Wellstead, P. E. and P. Zanker (1982). " Techniques for self-tuning ", *Optimal Control Applications & Methods*, 3, 305.

[11] Thaler, G. J. (1989). *Automatic Control Systems*, West, New York.

[12] Johnson, C. R. (1988). *Lectures on Adaptive Parameter Estimation*, Prentice-Hall, New Jersey.

[13] Åström, K. J. and B. Wittenmark (1990). *Computer-Controlled Systems: Theory and Design*, Prentice-Hall, New Jersey.

[14] National Instruments (1991). *LabWindows: Data Acquisition Library Reference Manual (Version 2.0)*, Data Acquisition and Control Products.

[15] National Instruments (1991). *AT-MIO-16F-5: User Manual*, Data Acquisition and Control Products.

[16] BEI Motion Systems (1990). *MX21 "Insta-Mount Series" Modular Incremental Rotary Optical Encoder: Technical Data*, Computer Products.

[17] Åström, K. J. and B. Wittenmark (1980). " Self-tuning controllers based on pole-zero placement ", *IEE Proceedings*, Vol. 127, Pt. D, No. 3, 120.

[18] Cordero, O. A. and D. Q. Mayne (1981). " Deterministic convergence of a self-tuning regulator with variable forgetting factor ", *IEE Proceedings*, Vol. 128, Pt. D, No. 1, 19.