AN ADAPTIVE CONTROL SCHEME USING THE GENERALIZED SECANT METHOD

Homayoon S.M. Beigi
T.J. Watson Research Center
International Business Machines
P.O. Box 704
Yorktown Heights, New York 10598

Abstract

This paper presents a solution to the adaptive-control problem by considering a general discrete-time linear time-invariant dynamic model to be controlled. A new adaptive-control scheme is presented which uses an extension to the generalized secant method of solving a set of linear equations. [1] This new method is shown to significantly outperform the self-tuning regulator in the control of two highly non-linear dynamic systems (a non-linear mass-spring-dashpot and a damped pendulum). Non-linearities of both systems have been amplified by requiring them to follow a highly non-linear and demanding trajectory.

1 Introduction

In general, most dynamic systems are too complex to be modeled properly. In most cases, an ideal control system would therefore be a system which gives the best performance for the least user-provided amount of information about the system being controlled. Adaptive-control systems have generally been developed for such applications. Two well known adaptive controllers are the Model-Reference adaptive-control and the Self tuning Regulator. [2] This paper will present a solution to the adaptive-control problem by considering a general discrete-time linear time-invariant model given by the following equation,

$$y(t+1) = \underbrace{CA}_{\hat{A}} x(t) + \underbrace{CB}_{\hat{B}} u(t) + w(t) \quad (1)$$

In equation 1 the state x is n-dimensional, the control u is m-dimensional, the output y is q-dimensional, and t is the time step. Also, A, B, and $w(t)$ are assumed to be unknown – otherwise one could determine in advance what control to use to minimize the tracking error. For simplicity, the dimension (order) of the system is assumed to be known, but generalization to just knowing an upper bound on the order is easily considered.

2 Formulation

Let us assume that the elements in u(t) in equation 1 could be arranged in such a way that all the elements which are dependent on time t are placed at the bottom portion of the vector u(t) and all the elements which signify the effects of the previous $0 \leq \tau < t$ inputs are placed on the top portion of the $u(t)$ vector. Namely,

$$u(t) = \left[\underbrace{{u^1}^T(t-1)}_{1xm_1} \,|\, \underbrace{{u^2}^T(t)}_{1xm_2} \right]^T \quad (2)$$

Therefore, equation 1 could be rewritten in the following form,

$$y(t+1) = \hat{A}x(t) + \underbrace{\left[\hat{B}^1 | \hat{B}^2\right]}_{\hat{B}} \begin{bmatrix} u^1(t-1) \\ u^2(t) \end{bmatrix} + w(t) \quad (3)$$

Rewrite equation 3 in the following form,

$$y(t+1) = \underbrace{\begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} x(t) \\ u(t) \end{bmatrix}}_{v(t)} + w(t) \quad (4)$$

Let us assume for now that there is some control $u^2(t)$ such that when applied to the real system, S, it would generate an output y_{t+1} equal to the desired value y^d_{t+1} and let us call the v vector containing that control, $\tilde{v}_t$. Therefore,

$$y^d_{t+1} = S\tilde{v}_t \tag{5}$$

Now let us assume that there exists some combination of the system matrix S and the control input $u^2(t)$ such that when placed in the v vector, attain the same result as in equation 5. Let us call this pair S_t and v_t. Therefore,

$$y^d_{t+1} = S_t v_t \tag{6}$$

There might not exist such a pair of S and v that would get the output at time $t+1$ to be equal to the desired value. However, given an S_t, one could always find some $u^2(t)$ such that when placed in v, the Frobenius norm of the difference between the actual output and the desired output at time $t+1$ is minimized. This $u^2(t)$ is given by the following equation:

$$u^2(t) = \hat{B}_t^{2^\dagger} \left(y^d(t+1) - \left[\hat{A}_t | \hat{B}_t^{\,1}\right] \begin{bmatrix} x(t) \\ u^1(t-1) \end{bmatrix} \right) \tag{7}$$

$u^2(t)$ given by equation 7 minimizes the Frobenius norm of the error, $\|y^d(t+1) - y(t+1)\|_F$. Through the application of the control given by equation 7, to the actual system, at time t, some output y_{t+1}, will be generated. Let us assume that at time t, the best estimate of the system matrix available to us is S_t. Therefore, the difference between the true system matrix S and S_t could be denoted by ΔS_t. Now let us impose the condition that S_{t+1} would generate the same output y_{t+1} as would the real system, S, with the given v_t. Therefore,

$$y_{t+1} = S_{t+1} v_t \tag{8}$$

Rewriting equation 8 in terms of S_t and the difference between S_t and S_{t+1} such that equation 8 holds,

$$y_{t+1} = (S_t + \Delta S_t)\, v_t \tag{9}$$

One way to solve equation 9 is to update the system matrix in one direction each time using the generalized secant update [1] in the following fashion,

$$S_{t+1} = S_t + \Delta S_t \tag{10}$$

where,

$$\Delta S_t = \frac{(y(t+1) - S_t v_t)\, z_t^T}{z_t^T v_t} \tag{11}$$

z_t's are secant projection vectors, chosen in the following way:
- *If* $t \geq n+m-1$, *then,* z_t *is chosen orthogonal to the previous* $n+m-1$ *vectors,* $v_{t-(n+m-1)}, \cdots, v_{t-1}$.
- *If* $t < n+m-1$, *then,* z_t *is chosen orthogonal to the available t vectors,* $v_0, \cdots, v_{t-1}$.

One possibility is to pick z_t as a linear combination of $v_0, \cdots, v_t$ which would be orthogonal to all $v_0, \cdots, v_{t-1}$. A few orthogonalization methods are available in the literature which may be used for the actual evaluation of the z vectors. These methods include the well-known Gram-Schmidt orthogonalization process [3] and a more advanced technique due to Fletcher [4] in which the number of vectors in the set are likely to be increased or decreased. Therefore, using an orthogonalization method and equations 7 and 10, in a recursive fashion, a control could be evaluated using equation 7 and after being applied to the real system, based on the response of the system, a better estimate of the system matrices could be obtained using equation 10.

3 Convergence

The set of z's picked in equation 10 has the property that,

$$\Delta S_t v_j = 0 \qquad 0 \leq t-j < n+m-1 \tag{12}$$

therefore,

$$S_{t+1} v_j = [S_{t+1} + \Delta S_{j+1} + \cdots + \Delta S_t]\, v_j \tag{13}$$

$$S_{t+1} v_j = S_{j+1} v_j \qquad 0 \leq t-j < n+m-1 \tag{14}$$

Now assume that some $s \leq n+m$ is the maximum number of linearly independent v's which could be found and thus v_s is a linear combination of all previous s linearly independent vectors. Then,

$$v_s = \alpha_0 v_0 + \cdots + \alpha_{s-1} v_{s-1} \tag{15}$$

From equation 15,

$$S_s v_s = \sum_{j=0}^{s-1} \alpha_j S_s v_j \tag{16}$$

Also,

$$S_{i+1} v_j = S_{j+1} v_j \; 0 \leq j \leq s-1 \; (from\ eq.\ 14) \tag{17}$$

$$S_{i+1} v_j = y_{j+1} \quad (from\ eq's\ 9\ and\ 10) \tag{18}$$

$$S_{i+1} v_j = S v_j \quad (from\ eq.\ 4) \tag{19}$$

Substituting equation 19 into 16,

$$S_s v_s = S \sum_{j=0}^{s-1} \alpha_j v_j \tag{20}$$

or,

$$S_s v_s = S v_s \tag{21}$$

Using equations 4 and 21 in the absence of noise,

$$\begin{aligned} y_{s+1} &= S_s v_s \\ &= S v_s \end{aligned} \tag{22}$$

Therefore, after $s \leq n+m$ time instants, the control given by equation 7 provides an output which will minimize the Frobenius norm of the error at time $s+1$ (i.e., $min||y^d(s+1) - y(s+1)||_F$).

4 Simulations and Results

The control algorithm of equations 7 and 10 was applied to controlling two very non-linear systems. System 1 was a non-linear mass-spring-dashpot with the following differential equation:

$$m\ddot{\alpha} + c\,(1 + |\alpha|)\,\dot{\alpha} + (k|\alpha|)\,\alpha = T \tag{23}$$

System 2 was a damped pendulum with the following differential equation:

$$ml^2\ddot{\alpha} + c\dot{\alpha} + mglsin\alpha = T \tag{24}$$

where α in equation 23 is a measure of distance which is the linear position of the mass m from its equilibrium position. In equation 24, α is a measure of the angle the pendulum makes with the vertical axis (equilibrium point).

Both systems were given a very demanding non-linear trajectory to follow. This trajectory is given in figure 1 and it will ensure that both systems operate in a very highly non-linear region. This is a very good test of the robustness of the algorithm. The performance of the adaptive-control algorithm of this paper is compared to that of the self-tuning regulator. In both the self-tuning regulator and the secant controller, a proportional-derivative (PD) controller was used to start up the control process. For the case of the above systems, since they are second order, the first two time steps used a PD controller which was then switched to the corresponding adaptive controller at time step 3. Both controllers started with the same initial guesses for the system matrices.

Figure 2 compares the secant controller with the self-tuning regulator when applied to equation 23. In this run, $m = 0.1kgr$, $c = 0.1N/(m/s)$ and $k = 0.1N/m$. Figures 3 make a similar comparison for the pendulum of equation 24 with $m = 1kgr$, $c = 1N/(m/s)$, and $l = 0.1m$. Table 1 shows the sum of squares of errors for the three simulation runs of figures 2 and 3. Other runs were made using different parameters for the above systems and the outcomes of the performances were very consistent with those presented here.

As a practical precaution for stability and quick convergence after evaluating the vector v_t using equation 7, a linear independence test was made by evaluating,

$$\omega_t = \frac{|z_t^T v_t|}{||z_t||_E ||v_t||_E} \tag{25}$$

If $\omega_t < \rho$, $0 < \rho < 1$ (in the simulations of this paper, $\rho = 0.00001$), then the control for that step is changed such as to make $\omega_t > \rho$. This is done using the information provided by the orthogonalization method. [3]

Fig.	Self-Tuning	Secant
2	$9.9m^2$	$6.9m^2$
3	$5.2Rad^2$	$1.8Rad^2$

Table 1: $||Error(t)||_E^2$ *for Figs. 2 & 3*

5 Conclusion

The secant adaptive-controller converges much more quickly than the self-tuning regulator and has a consistently lower overall error in the sum of squares sense (up to 65% lower). It is very robust to have been applied to such non-linear systems at highly non-linear trajectories displaying a very good performance and does not require any more computation than the self-tuning regulator. Reference [3] shows that basically, with the practical introduction of the restrictions on ω_t of equation 25, the secant controller takes a few steps to learn the parameters of the system before it greatly affects the dynamics of it. Once it realizes the system parameters, it quickly reduces the error to an amount very close to zero. This observation was done with many different systems simulated and also in the case of secant learning controller [3].

References

[1] J.G.P. Barnes, "An Algorithm for Solving Non-linear Equations Based on the Secant Method," Computer Journal, Vol. 8, pp. 66-72, 1965.

[2] K.J. Åström, "Adaptive Feedback Control," Proceedings of the IEEE, Vol. 75, No. 2, February 1987.

[3] Homayoon S.M. Beigi, "Neural Network Learning and Learning Control Through Optimization Techniques", Doctoral Thesis, Columbia University, 1991.

[4] R. Fletcher, "A Technique for Orthogonalization," J. Inst. Maths Applics, Vol. 5, pp. 162-166, 1969.

Fig. 1: Desired Trajectory

Fig. 2: Mass-Spring-Dashpot Output

Fig. 3: Pendulum Output