AC 2011-987: APPLICATION OF MICRO COMPUTERS IN DESIGN OF SELF-TUNING CONTROLLERS

Alireza Rahrooh, Daytona State College

ALIREZA RAHROOH Alireza Rahrooh is an Associate Professor of Electrical Engineering Technology at Dayton State College. He received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the Univ. of Akron, in 1979, 1986, and 1990, respectively. His research interests include digital simulation, nonlinear dynamics, chaos, control theory, system identification and adaptive control. He is a member of ASEE, IEEE, Eta Kappa Nu, and Tau Beta Pi.

Robert De la Coromoto Koeneke, Daytona State College

Robert Koeneke is an Assistant Professor of Electrical Engineering Technology at Daytona State College. He received his B.S. in Electronics Engineering from Universidad Simon Bolivar in 1977 and his M.S. in Computer Science from Santa Clara University in 1982. His research interest include embedded systems, digital programmable devices and computer communications. He is a member of IEEE and ACM.

Walter W. Buchanan, Texas A&M University

Walt Buchanan is the J.R. Thompson Chair Professor at Texas A&M University. He is a Fellow and served on the Board of Directors of both ASEE and NSPE, is Past-Chair of the Professional Engineers in Higher Education of NSPE, is a Past President of the Massachusetts Society of Professional Engineers, and is a registered P.E. in six states. He is a past member of the Executive Committee of TAC of ABET. Buchanan is on the editorial board of the Journal of Engineering Technology, has authored over 170 publications, and has been a principal investigator for NSF.

Application of Micro Computers in Design of Self-tuning Controllers

ABSTRACT

The speed and accuracy of microprocessors has extensively changed the way control systems are designed. Process controllers can be "taught" to adjust themselves without any operator intervention. These self-tuning controllers are programmed to provide a stable system response under various disturbance conditions.

This paper presents a fluid level system to be modeled and controlled utilizing A Self-tuning controller to improve the output response to a step input. The digital controller will provide the required output with variations in a single plant parameter. A fully adaptive controller will then be implemented using PC Matlab to allow for any of the plant parameters to vary and still maintain a suitable output. This concept can be used in Senior Design Project Course as well as in Master Programs in developing nations with limited resources.

The popularity of the PID controller and the increased use of microprocessors have led to a digital version of the algorithm for use in computer control applications. The first part of this paper will look at the output response of the specified plant to a step input. Some of the plant parameters will be adjusted to obtain the best results. The system response is improved by adding a PID[1] controller. The next part will show how a self-tuning controller will allow any of the plant parameters to vary without greatly changing the system response. The least-squares algorithm[2] will be used to update the controller values during every sampling period.

I. INTRODUCTION

The most widely used industrial process controller is the PID controller[3]. It is a combination of three distinct components and is used in closed loop feedback systems. In most cases, the input is the error signal, which is the difference between the system set point value and the system output. The controller output signal is Proportional to: the error, the Integral of the error, and the Derivative of the error. The PID has the following form[3]:

$$u(s) = K[1 + \frac{1}{T_i s} + T_d s] \qquad (1)$$

where K is the proportional gain, T_i is the integral time, and T_d is the derivative time. There are times when the derivative portion of the PID controller is not needed for satisfactory system control. A PI controller is capable to provide satisfactory control for first order systems. However, higher order systems are controlled via PID controller. The system to be controlled in this paper is third order so PID control will be used.

The popularity of the PID controller and the increased use of microprocessors have led to a digital version of the algorithm for use in computer control applications. The first part of this paper will look at the output response of the specified plant to a step input. Some of the plant parameters will be adjusted to obtain the best results. The next part will show how the system response is improved by adding a PID controller. A digital PID controller will be used so that the controller parameters can be adjusted on-line to account for variations in one of the plant values. Pole placement technique will be used in the design. One of the plant parameters that can be externally adjusted will be varied. It will be shown that the system response will remain the same over the entire range of adjustment. The last part will show how an adaptive controller will allow any of the plant parameters to vary without greatly changing the system response. The least-squares algorithm[2] will be used to update the controller values during every sampling period.

II. SYSTEM DESCRIPTION

The system is intended to maintain a specified fluid level in a tank is shown in Figure-1. The tank has an outlet that is controlled by a valve. This valve can be adjusted at any time while the system is operational and can be set at any position. No matter what the outlet flow requirements might be, it is important that the tank maintain a certain level. The level can be set from 0 to a maximum height of 0.5 meters by adjusting the input voltage.

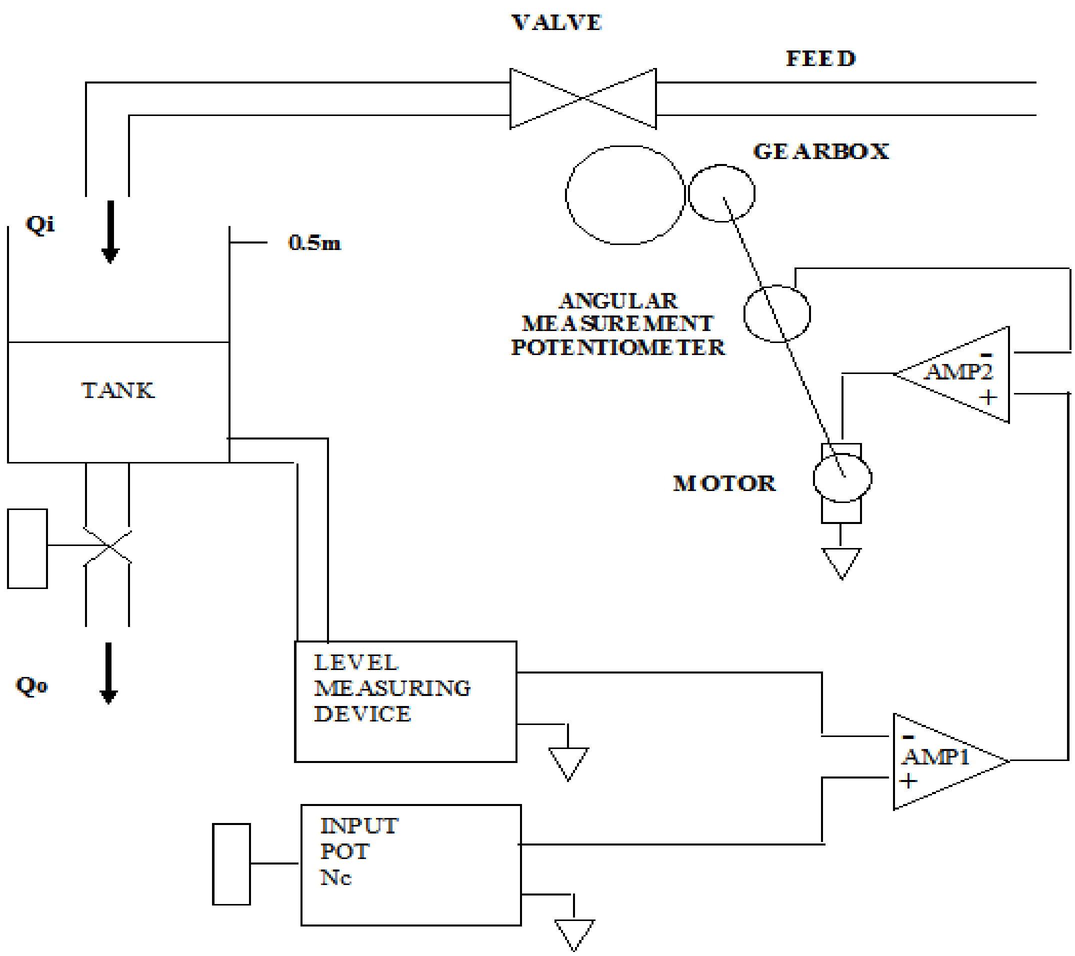

Figure-1. Fluid level tank system

A. System Block Diagram

The transfer function block diagram for the system is shown in Figure-2. The input signal is multiplied by 20 to produce the system input voltage V_p. Summing junction INSERT compares the input voltage to the voltage provided by a depth measuring device at the bottom of the tank. The difference is the error voltage and is amplified by A_1. This linear amp has an adjustable gain that will be calculated. The next summing junction INSERT compares the signal from A_1 with the signal produced by an angular measurement device attached to a motor shaft. The motor shaft is connected to a gearbox that will open and close a valve. This valve allows fluid to fill the tank. As the tank fills to the desired level, the error voltage decreases. Eventually, the angular displacement pot will output a voltage greater than the A_1 voltage causing the armature fed DC motor to reverse and close the valve. The linear amplifier feeding the motor also has an adjustable gain that will be calculated. The gearbox ratio is 20:1 so it has a gain of 0.05. The flow rate of the valve is proportional to INSERT and has a gain of 0.1. The capacitance C of the tank is equal to its cross-sectional area which is $0.25m^2$.

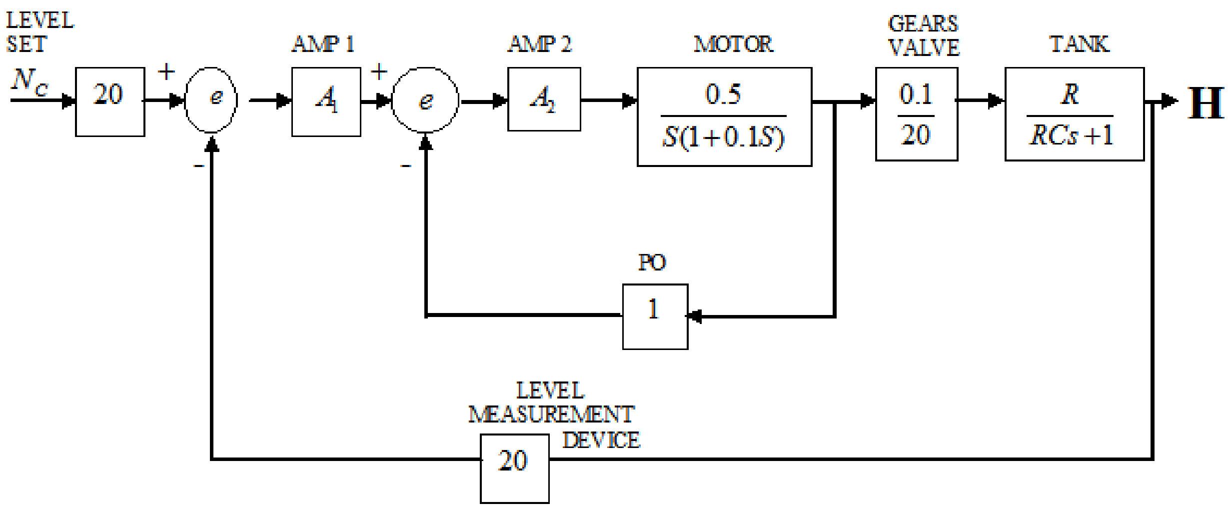

Figure-2. System Block Diagram

B. Tank Transfer Function

The transfer function for the tank is determined with the outlet valve fully closed and fully open. It will be shown that the tank without an outlet will be modeled as an integrator acting on the input flow rate. The output will be the height of the fluid in the tank with the inverse of the tank area as the gain. The minimum tank outlet resistance will be calculated using the following formula and the maximum circular outlet diameter of 8cm.

Area of outlet: $A_o = \pi(\frac{0.08}{2})^2 = 0.0050m^2$ (2)

Flow out: $Q_o = 0.0050\sqrt{2gh} = 0.0157\frac{m^3}{s}$ (3)

where g = 9.81m/s^2 and h = 0.5m tank level.
Equation (3) is for turbulent flow. The hydraulic resistance for the outlet under laminar flow is shown below.

Hydraulic resistance: $R = \frac{h}{Q_o} = \frac{0.05}{0.0157} = 32.05$ (4)

It is assumed that the flow through the outlet is laminar, therefore the hydraulic resistance will be constant for any specified outlet opening. The resistance calculated in (4) is for a fully open valve. As the valve is closed, the resistance will increase and reach infinity when the valve is fully closed. In an actual system, R would have to be determined by measuring the discharge flow rate at a specific fluid level. Since the outlet is controlled by a valve, the area A_o can be adjusted to achieve any desired resistance value.

The tank system will be linear based on the laminar flow assumption. An incremental change in the amount of volume in the tank will cause an incremental change in the height since the walls of the tank are rigid. Over a small period of time, the following differential equation results:

$$C(dh) = (Q_i - Q_o)dt \tag{5}$$

Where Q_i is the flow into the tank. Combining equations (4) and (5) results in:

$$RC\frac{dh}{dt} + h = RQ_i \tag{6}$$

The transfer function of the tank will be:

$$\frac{H(s)}{Q_i(s)} = \frac{R}{RCs+1} \tag{7}$$

In the case where the outlet valve is completely closed, Q_o will be zero. Then equation (5) becomes:

$$C(dh) = Q_i(dt) or C\frac{dh}{dt} = Q_i \tag{8}$$

The transfer function below shows that the tank acts as an integrator with the gain equal to the inverse of the tank capacitance.

$$\frac{H(s)}{Q_i(s)} = \frac{1}{Cs} \tag{9}$$

The closed-loop transfer function for the block diagram of Figure-2 is:

$$T(s) = \frac{2A_1A_2}{S^3 + (10 + \frac{4}{R})S^2 + (5A_2 + \frac{40}{R})S + 20\frac{A_2}{R} + 2A_1A_2} \tag{10}$$

III. Response Specifications

The system response will be required to meet certain design criteria. The tank height is 0.55 m so the maximum overshoot will be 10%. Otherwise the fluid level will exceed the tank limitations when the setpoint Nc is set to 0.5. The settling time will be set at 4 seconds. For a third order system, the following is the characteristic equation.

$$(S+A)(S^2+BS+C)=S^3+(A+B)S^2+(AB+C)S+AC=0 \quad (11)$$

where: A is a pole on the real axis, $B=2\zeta\omega_{n^1} and C=\omega_{n^2}$

ζ is the damping ration and ω_n is the undamped natural frequency of the system. The percent overshoot is calculated by the following:

$$PO=100e^{-\zeta\pi/\sqrt{1-\zeta^2}}=10\% \quad (12)$$

which leads to $\zeta\approx$ 0.6. The settling time will be the time when the response is within 2% of its final value. It is calculated by:

$$e^{-\zeta\varpi_n t_s}\le 0.02 \Rightarrow t_s=\frac{4}{\zeta\varpi_n}=4 \quad (13)$$

Using ζ, ϖ_n = 1.67. Then B = 2 and C = 2.78. The system block diagram shown in Figure-2 is reduced to the single block shown in Figure-3. The coefficients from equation (11) are then equated to the coefficients shown for the single block for like powers of s.

$$A+B=10+\frac{4}{R} \quad (14)$$

$$AB+C=5A_2+\frac{40}{R}\Rightarrow A_2=0.2(AB+C-\frac{40}{R}) \quad (15)$$

$$AC=20\frac{A_2}{R}+20A_2A_1\Rightarrow A_1=\frac{AC}{20A_2}-\frac{1}{R} \quad (16)$$

The value for the outlet resistance R in the equations above, will vary with the tank outlet valve opening. The largest outlet opening will be R = 32. As the valve is closed, the resistance will increase to infinity. Using the values determined for B and C, the value for A will vary from 8.125 to 8 as the valve closes. A pole on the real axis will occur at -A and complex conjugate poles will occur at:

$$-\zeta\varpi_n \pm j\varpi_n\sqrt{\zeta^2-1}=-1.000\pm j1.333 \quad (17)$$

The dominant poles of a system should be at least five times closer to the imaginary axis than any other poles. The real part of the complex poles is approximately eight times closer to the imaginary axis than the pole at -A. The complex poles will therefore be the dominant poles and the pole at -A will have little effect on the system response. The percent overshoot and the settling time should be very close to what was specified.

System Response

Equations (15) and (16) determine the values for the two amplifier gains. Both gains are dependent on R. As the outlet valve is turned, the gains will have to be adjusted to maintain the system specifications. The input was 0.5 and the two amplifier gains were adjusted for each different R value according to Table-I. As the value of R increased, the response approached the set point value.

The fluid in the tank never reaches the specified level when the outlet resistance equals 32. This is due to the steady-state error. For a unity feedback system that has no poles at the origin, the steady-state error for a step input is:

$$e_{ss} = \lim_{s \to o} \frac{s(\frac{A}{s})}{1 + G(s)} \tag{18}$$

For a step input with 0.5 height, the steady state error becomes:

$$e_{ss} = \frac{0.5}{1 + \frac{2A_1A_2}{\frac{20A_2}{R}}} = \frac{0.5}{1 + \frac{A_1R}{10}} \tag{19}$$

It is clear from the above equation that as the outlet resistance increases, the steady-state error will approach zero.

R	A_1	A_2	Final	Max	e_{ss}
32	2.86	3.55	0.451	0.492	0.049
64	2.91	3.66	0.475	0.518	0.025
1000	2.96	3.75	0.498	0.544	0.002

Table-I. Amplifier Gains and Output Response

The Final column in Table-I is the fluid level after 8 seconds when the system has reached a steady-state condition. The difference between this level and the desired level is shown in the e_{ss} column. The Max column is the overshoot level.

The only way to decrease the steady-state error when R = 32, is to increase the value of A_1. When A_1 = 4.2, the steady-state error is 0.035. But the overshoot is 0.552. As A_1 becomes greater that 4.2, the overshoot increases to a level that goes beyond the tank limit. Also, if A_1 = 4.2, A_2 = 3.55 and the valve is closed so that R = 1000, the overshoot would be 0.598. The system could not tolerate that much overshoot. This condition could only be prevented if the amplifier gains were adjusted as the valve was closed.

IV. SELF-TUNING CONTROLLER

There are many occasions when the plant parameters are either unknown or they can change over time. In either case, an adaptive controller will be able to provide control of the system output. Figure-3 shows how the system will be designed to incorporate an on-line parameter estimator. The estimator is responsible for estimating the plant parameters during every sampling period. It uses past inputs and outputs of the plant as well as previous estimates. The appropriate controller parameters are then derived from the estimated values.

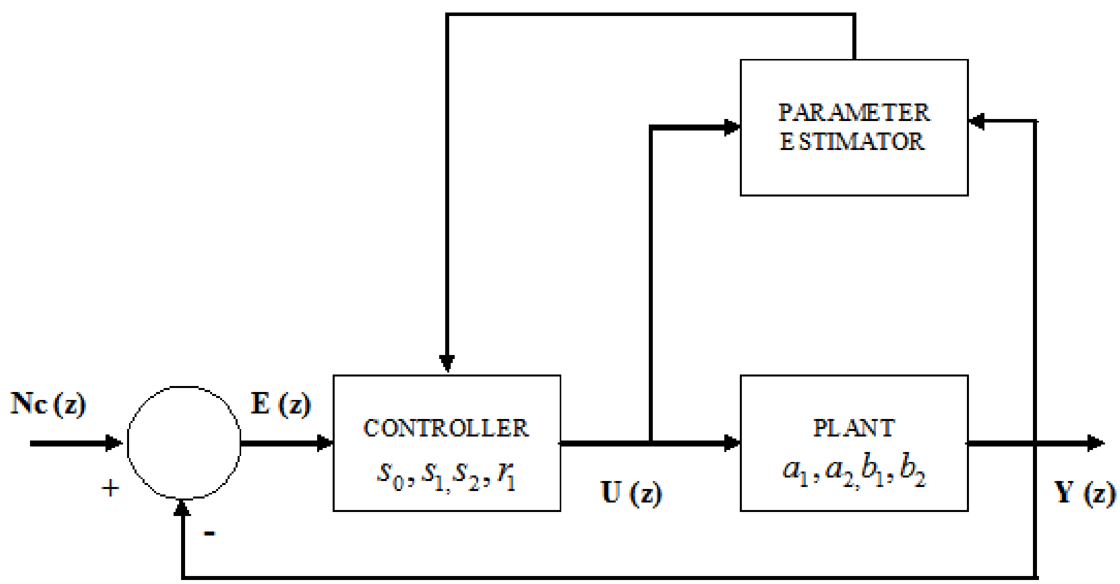

Figure-3. Self-tuning Controller Block Diagram

A. DARMA Model

There are various mathematical modeling techniques that use curve fitting algorithms to determine the parameters of a process model that best fit the available input/output data. Models that will completely describe the response of a system are called deterministic dynamical systems[2], and systems that have a random component in the output response are called stochastic[2]. The choice of a model is an important step in the prediction of a process. Some models are better suited to certain applications. The Deterministic Autoregressive Moving-Average or DARMA model[2] is one type of Model-Based Predictive Control and will be used to model the linear system in Part I.

The plant transfer function can be expressed as:

$$H_p(z) = \frac{Y(z)}{U(z)} = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} \tag{20}$$

Where U(z) is the output of the controller and the input to the plant. Equation (20) can be written as:

$$(1 + a_1 z^{-1} + a_2 z^{-2})Y(z) = (b_1 z^{-1} + b_2 z^{-2})U(z) \tag{21}$$

Since z^{-n} is a time delay of magnitude n, this leads to the following DARMA model.

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) + b_1 u(t-1) + b_2 u(t-2) \tag{22}$$

It shows that the present output is a combination of the past two outputs and the past two inputs to the plant. A higher order system would use a corresponding number of past inputs and outputs. The y terms on the left in equation (21) are the autoagressive component and the u terms are the moving-average component. The model can be expressed in the following form:

$$y(t) = \theta_o^T \phi(t-1); t \geq 0 \quad (23)$$

Where,

$$\begin{aligned} \phi(t-1) &= [-y(t-1), -y(t-2), u(t-1), u(t-2)] \\ \theta_o^T &= [\alpha_1, \alpha_2, b_1, b_2] \end{aligned} \quad (24)$$

The output of the controller is defined by:

$$U(z) = \frac{s_o z^2 + s_1 z + s_2}{(z-1)(z+r_1)} E(z) \quad (25)$$

or

$$[1 + (r_1 - 1)z^{-1} - r_1 z^{-2}]U(z) = [s_o + s_1 z^{-1} + s_2 z^{-2}]E(z) \quad (26)$$

Then:

$$u(t) = (1 - r_1)u(t-1) + r_1 u(t-2) + s_o e(t) + s_1 e(t-1) + s_2 e(t-2) \quad (27)$$

The controller output is a combination of the past two controller outputs plus the current error signal and the past two error values. The controller parameters will be a function of the plant parameters.

B. Least Squares Estimation Algorithm

The response of the modeled plant depends on the values associated with the model parameters a_1, a_2, b_1, and b_2. These parameter values may be unknown or time varying. In either case, the values will have to be estimated. This process, called parameter estimation, is a key component of adaptive control.
A popular class of on-line parameter estimation algorithms is of the form:

$$\hat{\theta}(t) = f(\hat{\theta}(t-1), D(t), t) \quad (28)$$

The current parameter estimate is a function of the last computed estimate. It is also a function of D(t) which is the past and present input/output data available at time t. The left side of the algorithm is an algebraic function which is specific to a certain type of estimation algorithm. A widely used form of equation (25) is:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + M(t-1)\phi(t-d)\bar{e}(t) \quad (29)$$

Where:

$\hat{\theta}(t)$ is the parameter estimate

$M(t-1)$ is the algorithm gain, $\phi\ (t-d)$ is input-output regression vector, and $\bar{e}(t)$ is the modeling error. Equation (26) can have many different forms depending upon the desired objectives of the algorithm.

The least-squares estimation algorithm will be presented here and has the following form:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-20\phi(T-1)}{1+\phi(t-1)^T P(t-2)\phi(t-1)}[y(t) - \phi(t-1)^T \hat{\theta}(t-1)] \qquad (30)$$

$$P(t-1) = P(t-2) - \frac{P(t-2)\phi(t-1)\phi(t-1)^T P(t-2)}{1+\phi(t-1)^T P(t-2)\phi(t-1)} \qquad (31)$$

for $t \geq 1, \hat{\theta}$ is given and P(-1) = any positive definite matrix

This algorithm will use the two previous system outputs and plant inputs to estimate the plant parameters a_1, a_2, b_1, and b_2. The controller parameters are functions of these values so as they are estimated, the controller is being tuned. The system characteristic equation and thus the response of the system will remain the same as was shown earlier.

The MATLAB software[4] was used to determine the output through 80 sampling times. The output is shown in Figure-4 for the input set-point = 0.5 and R = 32.

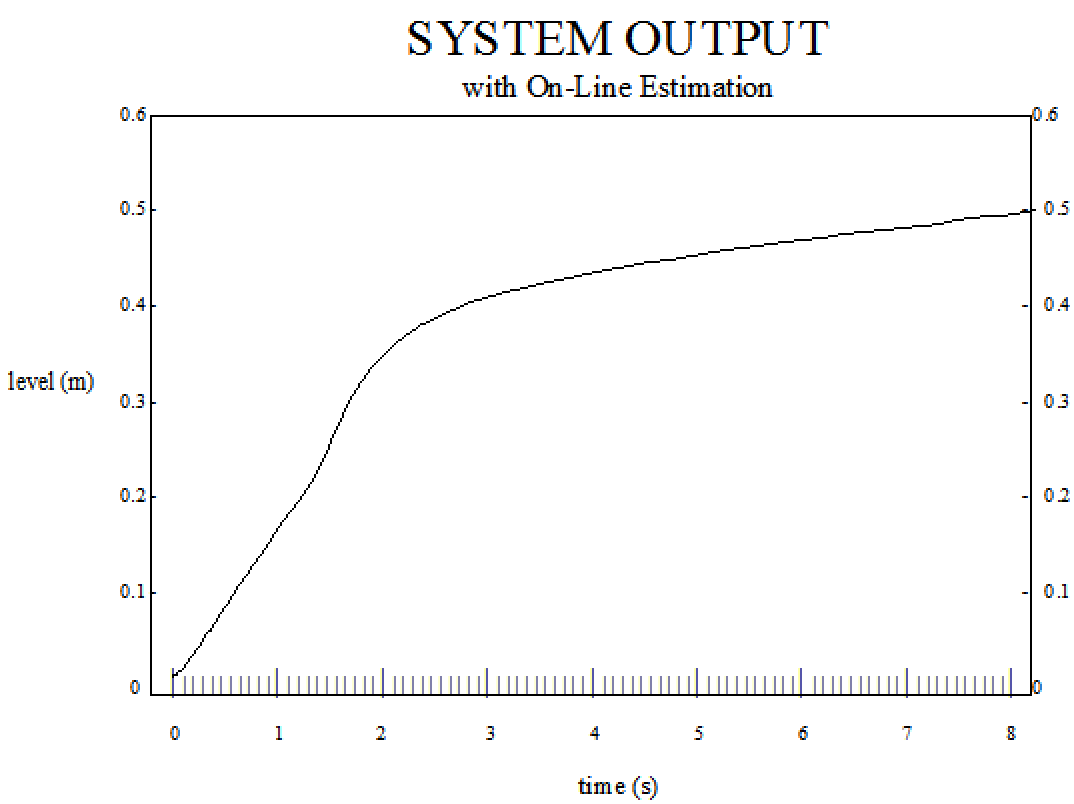

Figure-4. Response with Self-tuning controller

V. CONCLUSION

Model-based control is sensitive to modeling and parameter errors. We developed a solution to this problem by augmenting the standard controller with an adaptation mechanism. The proposed design incorporates an on-line identifier to eliminate parameter errors and individual joint controllers to compensate for unmodeled dynamics. This approach is particularly appealing because it retains the basic structure of model-based control systems.

References

1. G. S. Virk, Digital Computer Control Systems, McGraw-Hill Inc., New York, NY, 1991, pp 17-41.

2. G. C. Goodwin, K. S. Sin, Adaptive Filtering, Prediction and Control, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984, pp 7-62.

3. K. J. Aström, T. Hägglund, Automatic Tuning of PID Controllers, Instrument Society of America, Research Triangle Park, NC, 1988, pp 88-91.

4. PC-MATLAB User Guide, Version 3.1-PC, The MathWorks, Inc., Sherborn, MA, 1987.

MATLAB Computer Program

```
echo off

pt2=eye(4)
n=0.5
yt=0.0
et=n-yt

a1=-1.214
a2=0.368
b1=-0.404
b2=0.146
d1=-2.529
d2=2.273
d3=-0.844
d4=0.110

r1n=d2-a2+a1-(b1/b2)*(d3+a2-(b1/b2)*d4)-(b2/b1)*(d1-a1+1)
r1d=a1-1-(b1/b2)*(a2-a1+(b1/b2)*a2)-(b2/b1)
r1=r1n/r1d
s1=(d3-a2*r1+a2+a1*r1-(b1/b2)*(d4+a2*r1))/b2
s2=(d4+a2*r1)/b2
```

```
s0=(d1-a1-r1+1)/b1

ut1=0
ut2=0
yt1=0
yt2=0
et1=0
et2=0

ut=(1-r1)*ut1+r1*ut2+s0*et+s1*et1+s2*et2

theta1=[a1;a2;b1;b2]
phi1=[-yt1;-yt2;ut1;ut2]
pt1=pt2-(pt2*phi1*phi1'*pt2)/(1+phi1'*pt2*phi1)
theta=theta1+(pt2*phi1)*(yt-phi1'*theta1)/(1+phi1'*pt2*phi1)
k=0
p1t=[p1t,yt]
tm=[tm,k]


for zz=1:20
  pt2=pt1
  theta1=theta
  yt2=yt1
  yt1=yt
  ut2=ut1
  ut1=ut
  et2=et1
  et1=et

  phi1=[-yt1;-yt2;ut1;ut2]
  pt1=pt2-(pt2*phi1*phi1*pt2)/(1+phi1;*pt2*phi1)
  theta=theta1+(pt2*phi1)*(yt-phi1'*theta1)/(1+phi1'*pt2*phi1)
  yt=phi1'*theta
  et=n-yt
  ut=(1-r1)*ut1+r1*ut2+s0*et+s1*et1+s2*et2

  r1n=d2-a2+a1-(b1/b2)*(d3+a2-(b1/b2)*d4)-(b2/b1)*(d1-a1+1)
  r1d=a1-1-(b1/b2)*(a2-a1+(b1/b2)*a2)-(b2/b1)
  r1=r1n/r1d
  s1=(d3-a2*r1+a2+a1*r1-(b1/b2)*(d4+a2*r1))/b2
  s2=(d4+a2*r1)/b2
  s0=(d1-a1-r1+1)/b1
  k=k+1
  p1t=[p1t,yt]
  tm=[tm,k]
```

```
end

diary input.txt
theta
theta=input('enter [a1;a2;b1;b2] ')
diary off

for zz=1:60
  pt2=pt1
  theta1=theta
  yt2=yt1
  yt1=yt
  ut2=ut1
  ut1=ut
  et2=et1
  et1=et

  phi1=[-yt1;-yt2;ut1;ut2]
  pt1=pt2-(pt2*phi1*phi1'*pt2)/(1+phi1'*pt2*phi1)
  theta=theta1+(pt2*phi1)*(yt-phi1'*theta1)/(1+phi1'*pt2*phi1)
  yt=phi1'*theta
  et=n-yt
  ut=(1-r1)*ut1+r1*ut2+s0*et+s1*et1+s2*et2

  r1n=d2-a2+a1-(b1/b2)*(d3+a2-(b1/b2)*d4)-(b2/b1)*(d1-a1+1)
  r1d=a1-1-(b1/b2)*(a2-a1+(b1/b2)*a2)-(b2/b1)
  r1=r1n/r1d
  s1=(d3-a2*r1+a2+a1*r1-(b1/b2)*(d4+a2*r1))/b2
  s2=(d4+a2*r1)/b2
  s0=(d1-a1-r1+1)/b1
  k=k+1
  p1t=[p1t,yt]
  tm=[tm,k]
end

diary
p1t
diary
plot(tm,p1t)
```