Adaptive Control with Only Input & Output are Measurable

Student : Widya Ageng Setya Tutuko

Table of Contents

List of Figures

ABSTRACT

In this project report, adaptive control for second order system with unknown transfer function and only input-output are measurable will be studied and its performance will be examined. Based on Lyapunov function of state error and controller gain error, an adaptive law derivation will be shown in this report including the explanation of boundedness of various signals in adaptively controlled system, which will be proven to be entirely bounded. Simulation in MATLAB and Simulink are then carried out to experimentally prove the theory elaborated in Chapter I. Simulation results and discussion are presented in Chapter II which then is followed by simulation of non-step signal (sinusoid) tracking using adaptive control system, which is also proven to be stable and has marginally small tracking error. Conclusion and appendices consisting of codes and simulation entails the report .

I. Plant Model, Parameter and Adaptive Control Algorithm

A. Approximation of Plant's Transfer Function

Transfer function of the plant that needs to be equipped with adaptive control is empirically given by:

$$\frac{Y(s)}{U(s)} = \frac{b_0 s + b_1}{s^2 + a_1 s + a_2} \dots\dots\dots\dots\dots (1)$$

In concise polynomial form, we represent the transfer function as:

$$R_p y_p = k_p Z_p u \dots\dots\dots\dots\dots\dots\dots\dots (2)$$

Where $Y(s)$ and $U(s)$ are the Laplace transforms of output and input of the plant characterized by transfer function (1) above. The information given regarding plant (1) above is that the zeroes of plant are not at the RHP of s-plane and b_0 is a negative number. Thus, given that the plant is stable, b_1 must be a negative number following the information given that no zeroes are at RHP of s-plane.

In the adaptive control that will be designed later for this plant, we will only consider the case where only the input $u(t)$ and output $y(t)$ are measurable. These conditions are very relatable to the practical implementation where deploying multiple sensors to sense the intermediate signals (state variables) is not cost effective, hence adaptive control technique with only input and output being measurable is required.

Further information on the open loop plant are obtained by conducting open loop test and examine the transient and steady state response. From these responses, the plant is known to be stable, very lightly damped, and has natural frequency around 2 rad/s. The approximated model of the plant that we choose to comply with the open loop test findings is:

$$\mathrm{G_g(s)} = \frac{-1.2\mathrm{s} - 1.2}{\mathrm{s}^2 + 0.4\mathrm{s} + 4} \equiv \frac{\mathrm{K_p}\omega_\mathrm{n}^2(\mathrm{s}+1)}{\mathrm{s}^2 + 2\xi\omega_\mathrm{n} + \omega_\mathrm{n}^2} \dots\dots\dots\dots\dots\dots (3)$$

Obviously from the comparison between transfer function and the algebraic second order model of approximated plant $G_g(s)$ in equation (3) above, we notice that we have followings:

$$\omega_\mathrm{n}^2 = 4 \rightarrow \omega_\mathrm{n} = 2; \xi = \frac{0.4}{2\omega_\mathrm{n}} = 0.1; \mathrm{K_p} = \frac{-1.2}{\omega_\mathrm{n}^2} = -0.3$$

Zero of the approximated plant is at $s = -1$. This approximated plant from the information given on the open loop response is verified with MATLAB by using console codes below:

```
%% Guessed Plant to get initial value of adaptive parameter
Nat_freq = 2;
Damp_rat = 0.1;
Guessed_plant_Kp = -0.3;
Guessed_plant_Zp = (Nat_freq^2)*(Guessed_plant_Kp)*[1 1];
Guessed_plant_Rp = [1 2*Damp_rat*Nat_freq Nat_freq^2];
Guessed_plant = tf([Guessed_plant_Zp],[Guessed_plant_Rp]);
usim = [0 ones(1,1000)*10]';
tsim = (0:0.1:100)';
hold on;
lsim(Guessed_plant,usim,tsim)
grid;
axis([0 50 -9 2]);
hold off;
```

Figure 1 Code lines to simulate the open loop response to approximate the plant model and parameter

As evidenced by MATLAB verification above, we have the following step response from the approximated plant $G_g(s)$:

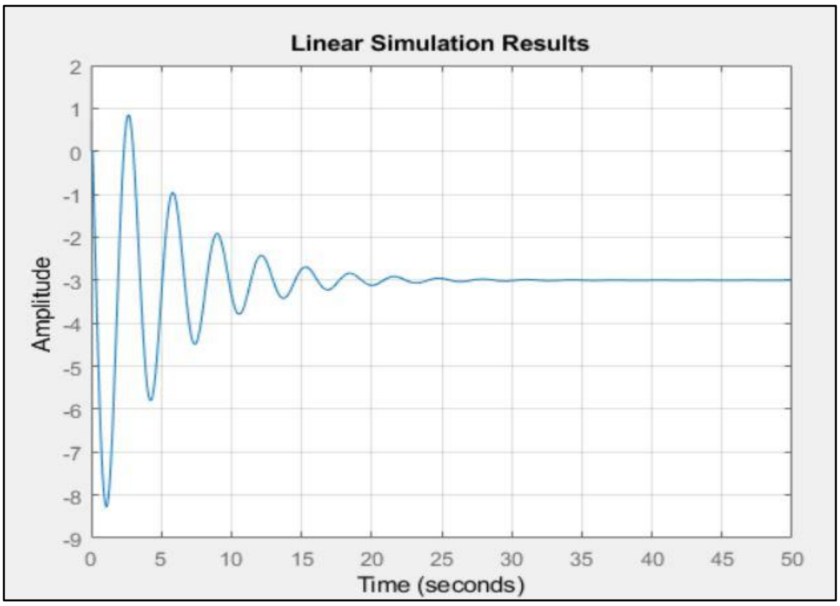


Figure 2 Open loop response of approximated plant with step input having amplitude 10 unit

The response plot in Figure 2 proved that there is a light damping and plant reached steady state within 30-35 seconds, indicates the open loop plant is eventually stable after some time.

B. Forcing True Plant and Reference Plant Selection

Other than having approximated plant model, we are also given the true plant transfer function that will be used only for the two purposes:

i). To calculate the exact parameter as well as using the true transfer function for simulation.

ii). Evaluate the performance of adaptive control that is designed based on the approximated model, comparing between real time parameter and exact parameter.

The plant has following transfer function, that we regard as 'true plant' $G(s)$:

$$G(s) = \frac{-0.5s - 1}{s^2 + 0.4s + 4.3}$$

In this exercise, we would eventually 'force' the step response of the true plant $G(s)$ whose exact transfer function assumed to be unknown (as in most real-life cases) to follow the step response from reference plant by using adaptive control law.

Now we will need to select the reference model that we would like our true plant to follow. We know that stable first order system has logarithmic response when step input is applied. Utilizing the behaviour of output response of stable first order reference model as reference response y_m and compare it with the output of adaptively controlled plant y_p to obtain referencing error e_1. Combining this output referencing error with adaptive law will result into forcing the true plant's output response to follow logarithmic response of first order reference model when step input is applied.

Another reason of using first order system as reference model is that the plant known to have relative degree one (pole polynomial has one degree more than zero polynomial). And thus, we can implement non-minimal filter T(s) (non-minimal because of having degree two and create four state variables coming from the two measurable signals: output $y(t)$ and input $u(t)$, while the plant is in second order form). By invoking non-minimal filter T(s) in the structure, we can 'transform' our true plant's model to reference model, having asymptotically equal response against step input. Mathematically, the transformation of true plant to be a reference plant is derived below:

Recall from equation (2):

$$R_p y_p = k_p Z_p u$$

We would like this equation to follow reference plant's transfer function:

$$R_m y_m = k_m r \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

By multiplying both sides of equation (2) with polynomials that we will obtain from solving Diophantine equation, we can transform $\frac{y_p}{u} = \frac{k_p Z_p}{R_p}$ from equation (2) to equalize $\frac{y}{r} = \frac{k_m}{R_m}$ from equation (4), as shown below:

Introducing Diophantine equation:

$$TR_m = ER_p + F \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Where E and T are monic polynomials, otherwise transformation will not be achieved.

Multiply equation (2) with polynomial E from Diophantine eq., we have:

$$ER_p y_p = k_p E Z_p u$$

$$(TR_m - F)y_p = k_p E \mathrm{Z_p} u \rightarrow \mathrm{T}R_m y_p = \mathrm{F} y_p + k_p E Z_p u$$

$$R_m y_p = \frac{k_p}{T}\left(\frac{F}{k_p} y_p + E Z_p u\right)$$

By letting $\mathrm{F}/k_p = \bar{F}$ and $\mathrm{EZ_p} = \bar{G}$:

$$R_m y_p = \frac{k_p}{T}(\bar{F} y_p + \bar{G} u) \ldots\ldots\ldots\ldots\ldots (6)$$

In equation (5), we will equalize degree of R_p to degree of T as well as the degree of R_m to E to transform equation (2) into equation (6) and eventually will be in the same form as equation (4).

From Diophantine equation, we can list down the degree of each polynomial in the equation:

$deg(Z_p) = m = 1$ (obtained from the known true plant's model in equation (1))

$deg(T) = deg(R_p) = n = 2$

$deg(R_m) = deg(E) = n^* = n - m = 1$

$deg(F) = n - 1 = 1$ (maximum remainder degree when $ER_p + F$ divided by T)

From knowing all these polynomial degrees, we can further breakdown equation (6) since $deg(EZ_p) = deg(\bar{G}) = deg(T) = n$, and leaving one term of input u without gain as follows:

$$\frac{\overline{G}}{T} = 1 + \frac{G_1}{T} \rightarrow R_m y_p = k_p \left(\frac{\overline{F}}{T} y_p + \frac{G_1}{T} u + u \right) \ldots\ldots\ldots (7)$$

$$\deg(\mathrm{F}) = deg(\overline{F}) = deg(G_1) = n - 1 = 1$$

As noted earlier, terms $\frac{y_p}{T}$ and $\frac{u}{T}$ are the output and input signals filtered by non-minimal filter $T(s) = s^2 + t_1 s + t_2$ and are the two of the four state variables during the implementation of adaptive control in this project scope, $\frac{y_p}{T} = \omega_y$ and $\frac{u}{T} = \omega_u$.

While the other two state variables are obtained by constructing the state space forms of the two equations $\frac{y_p}{T} = \omega_y$ and $\frac{u}{T} = \omega_u$, given below:

Recall equation (2): $R_p y_p = k_p Z_p u$, and multiply both side with filter $\frac{1}{T}$,

$$R_p \frac{y_p}{T} = k_p \frac{Z_p}{T} u \rightarrow R_p \omega_y = k_p Z_p \omega_u \rightarrow (s^2 + a_1 s + a_2)\omega_y = (b_0 s + b_1)\omega_u$$

$$s\dot{\omega_y} + a_1 \dot{\omega_y} + a_2 \omega_y = b_0 s\omega_u + b_1 \omega_u \rightarrow s\dot{\omega_y} = -a_1 s\omega_y - a_2 \omega_y + b_0 s\omega_u + b_1 \omega_u$$

As for y_p:

$$y_p = s^2 \omega_y + t_1 s\omega_y + t_2 \omega_y \rightarrow y_p = s\dot{\omega_y} + t_1 s\omega_y + t_2 \omega_y$$

$$y_p = -a_1 s\omega_y - a_2 \omega_y + b_0 s\omega_u + b_1 \omega_u + t_1 s\omega_y + t_2 \omega_y$$

$$y_p = (t_1 - a_1)s\omega_y + (t_2 - a_2)\omega_y + b_0 s\omega_u + b_1 \omega_u$$

Now we have three fundamental state equations to construct the state space form

$$s\dot{\omega_y} = -a_1 s\omega_y - a_2 \omega_y + b_0 s\omega_u + b_1 \omega_u \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

$$y_p = (t_1 - a_1)s\omega_y + (t_2 - a_2)\omega_y + b_0 s\omega_u + b_1 \omega_u \ldots\ldots\ldots\ldots\ldots (9)$$

$$u = s^2 \omega_u + t_1 s\omega_u + t_2 \omega_u \rightarrow u = s\dot{\omega_u} + t_1 s\omega_u + t_2 \omega_u \ldots\ldots\ldots\ldots (10)$$

Resulting into state space equation (11):

$$\begin{bmatrix} \dot{\omega_y} \\ s\dot{\omega_y} \\ \dot{\omega_u} \\ s\dot{\omega_u} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -a_2 & -a_1 & b_1 & b_0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -t_2 & -t_1 \end{bmatrix}}_{A_p} \begin{bmatrix} \omega_y \\ s\omega_y \\ \omega_u \\ s\omega_u \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B_p} u$$

$$y_p = \underbrace{[(t_2 - a_2) \quad (t_1 - a_1) \quad b_1 \quad b_0]}_{C_p^T} \begin{bmatrix} \omega_y \\ s\omega_y \\ \omega_u \\ s\omega_u \end{bmatrix}$$

Now, it becomes clear that non-minimal filter $T(s)$ generate four state variables in the adaptive control structure of this project.

Recalling equation (7):

$$R_m y_p = k_p \left(\frac{\overline{F}}{T} y_p + \frac{G_1}{T} u + u \right) \rightarrow R_m y_p = k_p \left(\overline{F} \omega_y + G_1 \omega_u + u \right)$$

If we choose $u = -\overline{F}\omega_y - G_1\omega_u + k^* r$, we then can reform equation (7) to be:

$$R_m y_p = k_p \left(k^* r \right) \rightarrow R_m y_p = k_{\mathrm{m}} r \ldots \ldots . (4^*)$$

Note(∗): real output plant y_p is generated from $\frac{k_m r}{R_m}$, reference input that's fed into reference model, which we ultimately would like to achieve.

We can rewrite the control law $u = -\overline{F}\omega_y - G_1\omega_u + k^* r$ into:

$$u = -(f_1 s + f_2)\omega_y - (g_1 s + g_2)\omega_u + k^* r$$

$$u = [-f_2 \quad -f_1 \quad -g_2 \quad -g_1 \quad k^*] \begin{bmatrix} \omega_y \\ s\omega_y \\ \omega_u \\ s\omega_u \\ r \end{bmatrix} = \overline{\theta}^{*T} \overline{\omega} \ldots \ldots \ldots \ldots (12)$$

We can find k^*, $\overline{F}$ and G_1 that will transform true plant's dynamic in equation (2) to follow reference plant's dynamic described in equation (4*). These parameters can only be exactly obtained if we exactly know the true plant's transfer function $G(s)$ given earlier. However, this is most likely not the case in real-life where true and exact plant's transfer function is not known. Thus, we need adaptation of these values in $\overline{\theta}^{*T}$ (over time, will be noted as real-time parameter $\overline{\theta}(t)$) and initial values $\overline{\theta}(0)$ to start the adaptation. The initial parameter $\overline{\theta}(0)$ can be obtained from deriving $k(0)$, $\overline{F}(0)$ and $G_1(0)$ using approximated plant $G_g(s)$.

Before fully designing the adaptive controller structure, we must define the reference model to generate reference response y_m. Based on the requirement of output response (reasonably fast and no steady-state error), we choose the first order:

$$\frac{y_m}{r} = \frac{k_m}{R_m} = \frac{5}{s+5} = \frac{\frac{1}{\tau}}{s+\frac{1}{\tau}}; \; where \; \tau \text{ is time constant, this case } \tau = 0.2 \text{ sec}$$

With this reference plant, the reference output will follow reference input, and equalizes 99.3% of the reference input within $5\tau = 1$ sec, as shown in below:

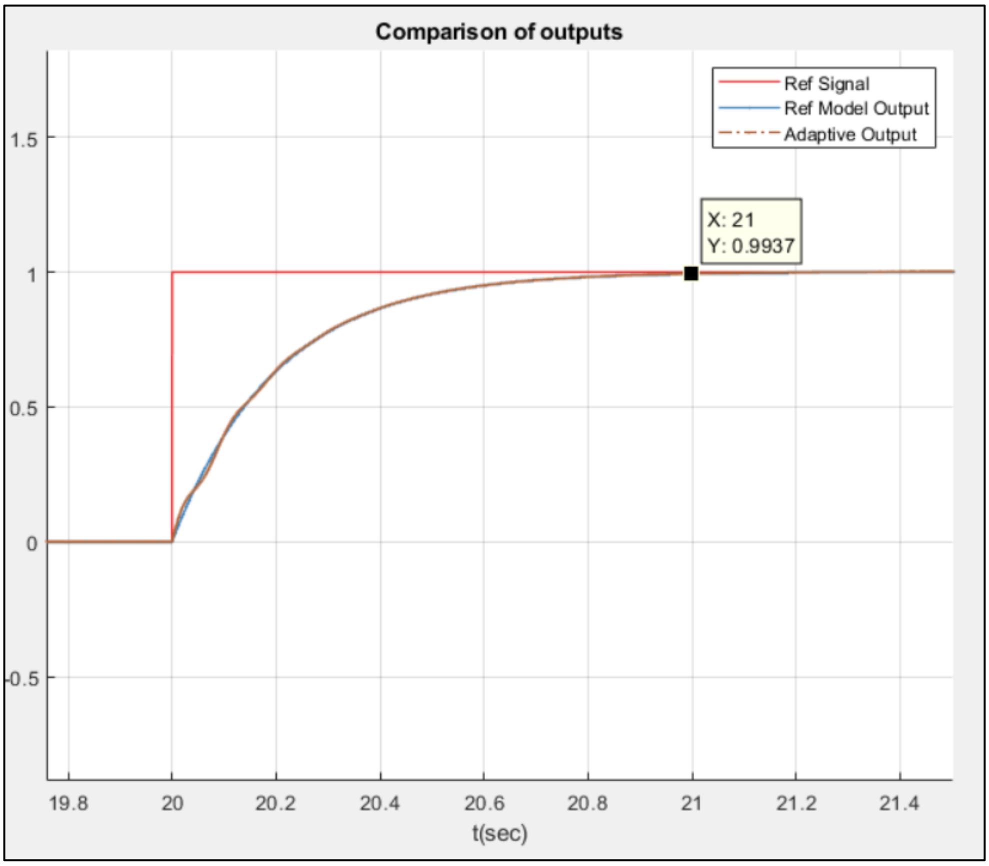


Figure 3 Reference Model output after 5tau of step reference signal being applied into reference model

C. Error Dynamics & Boundedness of Input and Output

Implementation of adaptive control ensures that the output error is tracking '0' asymptotically ($e \to 0 \; as \; t \to \infty$), and the cost function is of Lyapunov stable function after manipulating the other variables that affect the value of cost function. The mechanism can be done by adapting the parameter non-linearly and varying over time. Adaptation of these parameter leading to output error equalizing zero as t grows to very large number. Mathematical illustration to show how the parameter adaptation works will be shown in the next pages.

State space equation (11) that we formulated to equalize the output of adaptive system to the output of first order reference model is a realization of non-minimal filtering of the original plant input and output. The same non-minimal filter $T(s)$ can be used the same way to generate non-minimal state space realization for reference model:

Non-minimal state space equation of reference model (13):

$$\dot{\omega_m} = A_m \omega_m + B_m r \qquad y_m = C_m^T \omega_m \qquad \omega_m = \begin{bmatrix} \omega_{ym} \\ s\omega_{ym} \\ \omega_{um} \\ s\omega_{um} \end{bmatrix}$$

Comparing with the plant's state space equation in equation (11):

$$\dot{\omega} = A_p \omega + B_p u \qquad y_p = C_p^T \omega \qquad \omega = \begin{bmatrix} \omega_y \\ s\omega_y \\ \omega_u \\ s\omega_u \end{bmatrix}$$

With the correct adaptive law, the non-minimal state of the original plant ω will converge to non-minimal state of the reference model ω_m. These non-minimal state variables from original plant drive the input values u as shown in equation (12) in augmented configuration. Hence, the error of non-minimal states must be weighted and changes over time to generate input u such that the convergence is met.

Other than that, adaptive control signal (or input u) is a function of non-minimal states ω and adaptive parameter θ , i.e.: $u\big(\theta(t), \omega(t)\big)$. The idea of convergence on the non-minimal states is related to the convergence of adaptive parameter θ. Therefore, combining the dynamics of both to drive them to convergence will be the main idea of adaptive law implementation in this project.

Let non-minimal state error be, $e = \omega - \omega_m$, adaptive parameter error, $\Phi_\theta = \theta - \theta^*$ and adaptive reference parameter error, $\Phi_r = \theta_r - \theta_r^*$. Define a Lyapunov function (cost function) that uses all these errors quadratically:

$$V = e^T P e + \Phi^T \Gamma^{-1} \Phi; \text{where } \Phi = \begin{bmatrix} \Phi_\theta \\ \Phi_r \end{bmatrix}, \text{augmented adaptive parameter errors}$$

Derive both sides w.r.t t :

$$\dot{V} = 2e^T P \dot{e} + 2\Phi^T \Gamma^{-1} \dot{\Phi} \ldots\ldots\ldots\ldots\ldots\ldots (14)$$

From state space equations (11) and (13):

$$\dot{e} = \dot{\omega} - \dot{\omega_m} \rightarrow \dot{e} = A_p \omega + B_p u - A_m \omega_m - B_m r$$

$$u = \overline{\theta}^T \overline{\omega} = \theta^T \omega + \theta_r r = {\Phi_\theta}^T \omega + \Phi_r r + \theta^{*T} \omega + {\theta^*}_r r$$

$$\rightarrow \dot{e} = A_p \omega + B_p u - A_m \omega_m - B_m r$$

Then by substituting u in, the error equation becomes:

$$\dot{e} = A_p \omega + B_p {\Phi_\theta}^T \omega + B_p \Phi_r r + B_p \theta^{*T} \omega + B_p {\theta^*}_r r - A_m \omega_m - B_m r$$

By checking the validity of matching conditions:

$$A_m = A_p + B_p \theta^{*T} \text{ and } B_p {\theta^*}_r = B_m, \text{we reform the error equation into}$$

$$\dot{e} = A_m e + B_p {\Phi_\theta}^T \omega + B_p \Phi_r r \ldots\ldots\ldots\ldots\ldots (15)$$

The state space of error in equation (15) can be substituted into equation (14):

$$\dot{V} = 2e^T P A_m e + 2e^T P B_p {\Phi_\theta}^T \omega + 2e^T P B_p \Phi_r r + 2\Phi^T \Gamma^{-1} \dot{\Phi}$$

Now that we know state transition matrix A_m is asymptotically stable matrix as it is reference model that we set, the scalar term $2e^T P A_m e$ can be expanded and result the following time-derivative Lyapunov function:

$$\dot{V} = e^T (P A_m + A_m^T P) e + 2e^T P B_p {\Phi_\theta}^T \omega + 2e^T P B_p \Phi_r r + 2\Phi^T \Gamma^{-1} \dot{\Phi}$$

Where P is the solution of ARE equation $P A_m + A_m^T P = -Q$ following Kalman-Yakuborich Lemma with special case $\gamma = 0$, and it is positive definite matrix. Then expanded equation above becomes:

$$\dot{V} = -e^T Q e + 2e^T P B_p {\Phi_\theta}^T \omega + 2e^T P B_p \Phi_r r + 2\Phi^T \Gamma^{-1} \dot{\Phi} \ldots\ldots (16)$$

Q is also positive definite matrix since equation (15) shows that error dynamic is projected to be stable at $e = 0$ (in the long run) as noted in the stability of LTI system at equilibrium state $x = 0$ (which is state error in this application). Reconfiguring equation (16) back into augmented form, we have:

$$\dot{\mathrm{V}} = -\mathrm{e}^{\mathrm{T}}\mathrm{Qe} + 2\mathrm{e}^{\mathrm{T}}\mathrm{PB_p}\Phi^{\mathrm{T}}\begin{bmatrix}\omega\\ \mathrm{r}\end{bmatrix} + 2\Phi^{\mathrm{T}}\Gamma^{-1}\dot{\Phi}\ldots\ldots\ldots\ldots\ldots(17)$$

$\Phi^T\begin{bmatrix}\omega\\ r\end{bmatrix}$ is a scalar term, therefore it can be re-arranged to match with the third term in equation (17):

$$\dot{V} = -e^TQe + 2\Phi^T\begin{bmatrix}\omega\\ r\end{bmatrix}e^TPB_p + 2\Phi^T\Gamma^{-1}\dot{\Phi}$$

To have Lyapunov function that is only a function of non-minimal state error e, the dynamic of parameter error $\dot{\Phi}$ must be formulated such that it omits the 2nd term in the re-arranged time-derivative of Lyapunov function above.

Equalize $2\Phi^T\Gamma^{-1}\dot{\Phi}$ to negative of 2nd term:

$$2\Phi^T\Gamma^{-1}\dot{\Phi} = -2\Phi^T\begin{bmatrix}\omega\\ r\end{bmatrix}e^TPB_p \rightarrow \dot{\Phi} = -\Gamma\begin{bmatrix}\omega\\ r\end{bmatrix}e^TPB_p$$

Since $\Phi = \begin{bmatrix}\Phi_\theta\\ \Phi_r\end{bmatrix} = \begin{bmatrix}\theta - \theta^*\\ \theta_r - \theta_r^*\end{bmatrix}$ and $B_p\theta^*{}_r = B_m$:

$$\begin{bmatrix}\dot{\theta}\\ \dot{\theta}_r\end{bmatrix} = -\Gamma\begin{bmatrix}\omega\\ r\end{bmatrix}e^TP\frac{B_m}{|\theta^*{}_r|}sign(\theta^*{}_r)$$

Define $e_1 = y - y_m$ and matching condition $C_p^T = C_m^T$ from state space equation (11) and (13), the representation of e_1 can be related to non-minimal state error e , in equation below:

$$e_1 = C_m^Te = e^TC_m\ldots\ldots\ldots\ldots(18)$$

With the first order reference model being strictly positive real, then its state space triplet $\{A_m, B_m, C_m\}$, corresponding to:

$$y_m = C_m^T\omega_m, \text{ while}$$

$$\mathcal{L}\{\dot{\omega_m} = A_m\omega_\mathrm{m} + B_mr\} \xrightarrow{assume\ \omega_m(0)=0} \omega_m = (sI - A_m)^{-1}B_mr$$

Hence, $\frac{y_m}{r} = C_m^T(sI - A_m)^{-1}B_m \equiv \frac{k_m}{R_m} \equiv \frac{\frac{1}{\tau}}{s+\frac{1}{\tau}} = \frac{5}{s+5}$, is a strictly positive real transfer function. A slightly different triplet $\left\{A_m, \frac{B_m}{|\theta^*{}_r|}, C_m\right\}$ is therefore a SPR triplet too because $\frac{B_m}{|\theta^*{}_r|}$ has the same sign as B_m. Following the consequence of KY-Lemma on SPR triplet $\left\{A_m, \frac{B_m}{|\theta^*{}_r|}, C_m\right\}$, we have $P\frac{B_m}{|\theta^*{}_r|} = C_m$ since $\gamma = 0$, or in other word it is due to KY-Lemma consequence on quartet $\left\{A_m, \frac{B_m}{|\theta^*{}_r|}, C_m, \gamma = 0\right\}$.

Substituting $P\frac{B_m}{|\theta^*{}_r|} = C_m^T \rightarrow \begin{bmatrix}\dot{\theta}\\ \dot{\theta}_r\end{bmatrix} = -\Gamma\begin{bmatrix}\omega\\ r\end{bmatrix} e^T P\frac{B_m}{|\theta^*{}_r|} sign(\theta^*{}_r)$:

$$\begin{bmatrix}\dot{\theta}\\ \dot{\theta}_r\end{bmatrix} = -\Gamma\begin{bmatrix}\omega\\ r\end{bmatrix} e^T C_m sign(\theta^*{}_r)$$

Using result in equation (18), the final form of parameter dynamic equation $\begin{bmatrix}\dot{\theta}\\ \dot{\theta}_r\end{bmatrix}$ to remove the contribution of parameter error states Φ in declining rate of Lyapunov function $\dot{V}$ is given below:

$$\begin{bmatrix}\dot{\theta}\\ \dot{\theta}_r\end{bmatrix} = -sign(\theta^*{}_r)\Gamma\begin{bmatrix}\omega\\ r\end{bmatrix} e_1 \ldots\ldots\ldots\ldots\ldots (19)$$

With the dynamic equation (19) being implemented, equation (17) becomes:

$$\dot{V} = -e^T Q e + 2e^T P B_p \Phi^T \begin{bmatrix}\omega\\ r\end{bmatrix} + 2\Phi^T \Gamma^{-1}\dot{\Phi} \rightarrow \dot{V} = -e^T Q e \ldots . (20)$$

Value of equation (20) is guaranteed less than or equal to 0 since Q is positive definite matrix. Hence, the Lyapunov function above is decreasing with time and lower bounded by 0 since it is a quadratic equation $V = e^T P e + \Phi^T \Gamma^{-1} \Phi$ with Γ is a positive definite matrix and symmetric. As a result, V is then bounded between initial value and 0, leading to chain of consequences summarized below:

a. V is bounded $\rightarrow$ (e, Φ) are bounded

$$\rightarrow V\big(e(0), \Phi(0)\big) - V\big(e(t), \Phi(t)\big) = \int_0^t e^T Q e \, d\tau$$

When $t \rightarrow \infty$, $V\big(e(0), \Phi(0)\big) - V\big(e(\infty), \Phi(\infty)\big) = \int_0^\infty e^T Q e \, d\tau \leq \zeta$ due to boundedness of V.

b. (e, Φ) are bounded $\xrightarrow{e=\omega-\omega_m, \begin{bmatrix}\Phi_\theta\\ \Phi_r\end{bmatrix} = \begin{bmatrix}\theta-\theta^*\\ \theta_r-\theta_r^*\end{bmatrix}}$ (ω, θ) are bounded

$\dot{e} = A_m e + B_p {\Phi_\theta}^T \omega + B_p \Phi_r r$ is then bounded and together with $\int e^T Q e \, d\tau \leq \zeta$ concluding $\lim_{t\rightarrow\infty} e(t) = 0$. In other words, error slope $\dot{e}$ is bounded at any t indicating there is no error spike over small interval δt that distort the long term cumulative error and cumulative error overtime from $t=0$ to $t \rightarrow \infty$, is capped at ζ . Which implies an infinitesimal error δ_e to be added after some time $t > t_s$ and therefore $e(t)$ asymptotically approaches zero which, as desired, corresponds to y_p asymptotically tracking the reference model output y_m after some time (in practical point of view).

c. (e, Φ) are bounded $\xrightarrow{e_1 = C_m^T e}$ e_1 is bounded

d. e_1 *is* bounded $\xrightarrow{e_1 = y - y_m}$ y is bounded

e. (ω, θ) are bounded $\xrightarrow{u = \bar{\theta}^T \bar{\omega} = \theta^T \omega + \theta_r r}$ u is bounded

As noted above, boundedness of V affect the boundedness of input and output of the controlled plant. Therefore, adaptive law $\begin{bmatrix} \dot{\theta} \\ \dot{\theta}_r \end{bmatrix} = -sign(\theta^*{}_r)\Gamma \begin{bmatrix} \omega \\ r \end{bmatrix} e_1$ensures that output of the adaptively controlled plant asymptotically track output of reference plant in the long run with bounded plant's output y_p and bounded control signal u, and in fact ensure the boundedness of all intermediate signals in the adaptation process noted in the summary *a*-*e* above.

II. Construction of Simulation and Simulation Results

A. State Space Representation of Dynamics

Each of the transfer function blocks related to adaptation can be represented by state-space blocks in Simulink. Therefore, dynamic processes in the adaptive controller such as non-minimal filtering of input and output of original plant and parameter dynamic will be represented in state-space form.

a) Non-Minimal Filtering of Plant's Input and Output

From previous section, we have $\frac{y_p}{T} = \omega_y$ and $\frac{u}{T} = \omega_u$. Multiply both sides of the two equations with $T(s) = s^2 + t_1 s + t_2$ gives:

$$y_p = s\dot{\omega}_y + t_1 s\omega_y + t_2\omega_y \text{ , and}$$

$$u = s\dot{\omega}_u + t_1 s\omega_u + t_2\omega_u$$

For output dynamics, let's assign state space representation below as equation (21):

$$\begin{bmatrix} \dot{\omega_y} \\ s\dot{\omega_y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -t_2 & -t_1 \end{bmatrix} \begin{bmatrix} \omega_y \\ s\omega_y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} y_p$$

$$\begin{bmatrix} \omega_y \\ s\omega_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_y \\ s\omega_y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} y_p$$

$$\text{Quartet} = \{\text{A, B, C, D}\} \equiv \left\{ \begin{bmatrix} 0 & 1 \\ -t_2 & -t_1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

While for input dynamics, assigned as equation (22):

$$\begin{bmatrix} \dot{\omega_u} \\ s\dot{\omega_u} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -t_2 & -t_1 \end{bmatrix} \begin{bmatrix} \omega_u \\ s\omega_u \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$\begin{bmatrix} \omega_u \\ s\omega_u \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_u \\ s\omega_u \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

$$\text{Quartet} = \{\text{A, B, C, D}\} \equiv \left\{ \begin{bmatrix} 0 & 1 \\ -t_2 & -t_1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

These two state-space representations (21&22) represent the filtering blocks of original plant input and output in Simulink.

b) Adaptive Parameter State Space

Earlier we derived the expression of adaptive parameter dynamic to ensure the boundedness of Lyapunov function. Recall equation (19):

$$\begin{bmatrix} \dot{\theta} \\ \dot{\theta}_r \end{bmatrix} = -sign(\theta^*{}_r)\Gamma \begin{bmatrix} \omega \\ r \end{bmatrix} e_1 \xrightarrow{augmented} \dot{\overline{\theta}} = -sign(\theta^*{}_r)\Gamma\overline{\omega}e_1$$

If we let $\overline{\omega}e_1$ as the input in the state-space representation with states $\dot{\overline{\theta}}$, then the state space form will be simpler:

$$\dot{\overline{\theta}} = \begin{bmatrix} O_{4x4} & 0 \\ 0 & 0 \end{bmatrix} \overline{\theta} - sign(\theta^*{}_r)\Gamma\overline{\omega}e_1$$

$$\overline{\theta} = \begin{bmatrix} I_{4x4} & 0 \\ 0 & 1 \end{bmatrix} \overline{\theta} + \underbrace{0.\Gamma}_{O_{5x5}}\overline{\omega}e_1$$

$$\text{Quartet} = \{A, B, C, D\} \equiv \left\{ \begin{bmatrix} O_{4x4} & 0 \\ 0 & 0 \end{bmatrix}, -sign(\theta^*{}_r)\Gamma, \begin{bmatrix} I_{4x4} & 0 \\ 0 & 1 \end{bmatrix}, O_{5x5} \right\}$$

Note that $\overline{\theta}$ is adaptive parameter for four non-minimal states and reference input, it has five dimensions. Thus, matrix A (zero matrix), B, C (identity matrix) and D are of size 5-by-5. These matrices are to be constructed with consistency to generate the correct model structure in Simulink.

c) Determination of Initial Adaptive Parameter

As state-space model of the dynamic of adaptive parameter Φ has been established, the initial values for states in this model are needed for subsequence adaptation and initial control signal. Using the approximated plant that has been established, the initial parameter can be obtained through solving Diophantine's equation.

Given approximated plant: $G_g(s) = \frac{-1.2s-1.2}{s^2+0.4s+4}$

$K_g = -1.2; Z_g = s + 1; R_g = s^2 + 0.4s + 4; \frac{y_m}{r} = \frac{k_m}{R_m} = \frac{5}{s+5}$

Deriving polynomials $\overline{F}(0), G_1(0), k(0)$ to obtain initial adaptive parameter values:

$$TR_m = E(0)R_g + F(0) \rightarrow (s^2 + t_1 s + t_2)(s + 5) = E(0)R_g + F(0)$$

Note that E(0) and F(0) are of degree $n^* = 1$ and $n - 1 = 1$ respectively. Additionally, E(0) must be monic, i.e.:$E(0) = s + E_1$

$$(s^2 + t_1 s + t_2)(s + 5) = (s + E_1)(s^2 + 0.4s + 4) + (F_1 s + F_2)$$

$$s^3 + (5 + t_1)s^2 + (5t_1 + t_2)s + 5t_2 = s^3 + (E_1 + 0.4)s^2 + (4 + 0.4E_1 + F_1)s + (4E_1 + F_2)$$

$$\therefore \begin{cases} E_1 = 5 + t_1 - 0.4 \\ F_1 = 5t_1 + t_2 - 0.4E_1 - 4 \\ F_2 = 5t_2 - 4E_1 \end{cases}$$

After setting up the values of T(s) (chosen non-minimal filter polynomial), the other unknown coefficients can be obtained.

Let $T(s) = s + 40s + 4 \times 10^4$, this will yield the numerical values of polynomials E and F. Subsequently can be used to derive $\overline{F}(0)$, $G_1(0)$and k(0) as follows:

$$T(s) = s + 40s + 4 \times 10^4 \rightarrow \begin{cases} E_1 = 5 + 40 - 0.4 = 44.6 \\ F_1 = 5(40) + 4 \times 10^4 - 0.4(44.6) - 4 = 40178.16 \\ F_2 = 5(4 \times 10^4) - 4(44.6) = 199821.6 \end{cases}$$

$$\frac{\bar{G}(0)}{T} = \frac{G_1(0)}{T} + 1 \rightarrow G_1(0) = \bar{G}(0) - T = E(0)Z_g - T$$

$$G_1(0) = (s + E_1)(s + 1) - T = s^2 + (E_1 + 1)s + E_1 - s^2 - t_1 s - t_2$$

$$\therefore\ G_1(0) = (E_1 + 1 - t_1)s + (E_1 - t_2) = 5.6s - 39955.4 \equiv g_1(0)s + g_2(0)$$

While for $\overline{F}(0)$ & $k(0)$:

$$R_g y_p = k_g Z_g u \rightarrow E(0)R_g y_p = k_g E(0)Z_g u \rightarrow TR_m y_p = F(0)y_p + k_g E(0)Z_g u$$

$$R_m y_p = k_g\left(\frac{\overline{F}(0)}{T}y_p + \frac{\overline{G}(0)}{T}u\right) \rightarrow R_m y_p = k_g\left(\frac{\overline{F}(0)}{T}y_p + \frac{G_1(0)}{T}u + u\right)$$

$$R_m y_p = k_g\left(\overline{F}(0)\omega_y + G_1(0)\omega_u + u\right) = k_m r$$

$$u = -\overline{F}(0)\omega_y - G_1(0)\omega_u + k(0)r \rightarrow k_m r = k_g k(0) r$$

$$\therefore \begin{cases} \overline{F}(0) = f_1(0)s + f_2(0) = \dfrac{F(0)}{k_g} = \dfrac{40178.16s + 199821.6}{-1.2} = -33481.8s - 166518 \\ k(0) = \dfrac{k_m}{k_g} = \dfrac{5}{-1.2} = -4.1667 \end{cases}$$

With all these polynomials in place, the initial adaptive parameter $\overline{\theta}(0)$ are obtained:

$$\overline{\theta}(0)^T = [-f_2(0) \quad -f_1(0) \quad -g_2(0) \quad -g_1(0) \quad k(0)]$$

$$\rightarrow \overline{\theta}(0)^T = [166518 \quad 33481.8 \quad 39955.4 \quad -5.6 \quad -4.1667]$$

Together with the quartet of state-space of adaptive parameter $\{A, B, C, D\} \equiv \left\{\begin{bmatrix} O_{4x4} & 0 \\ 0 & 0 \end{bmatrix}, -sign(\theta^*{}_r)\Gamma, \begin{bmatrix} I_{4x4} & 0 \\ 0 & 1 \end{bmatrix}, O_{5x5}\right\}$, a Simulink state-space block can be created.

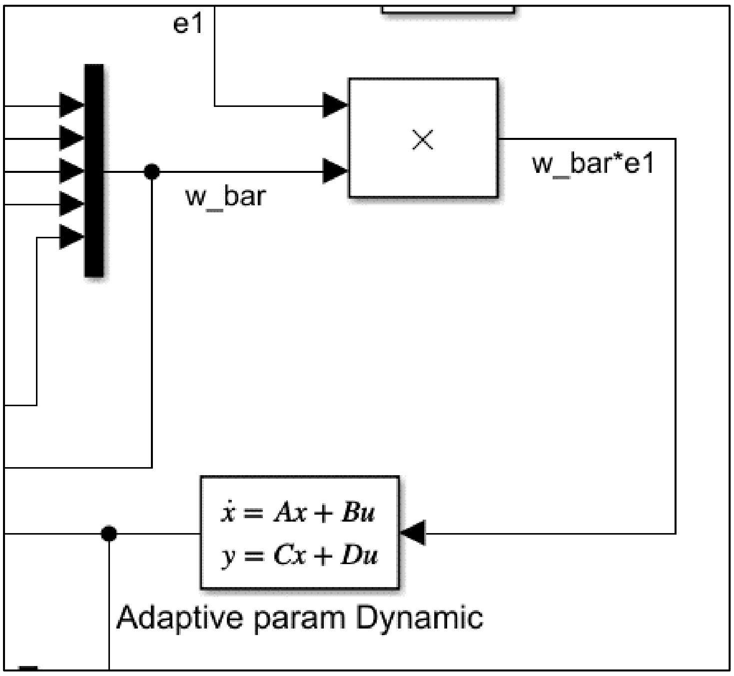


Figure 4 Simulink block implementing adaptive parameter dynamic in state space form

B. Simulation Model in Simulink & Results

With all the state space quartets for ω_u, ω_y and $\overline{\theta}$ established, recalled below:

Quartet for ω_u and ω_y: $\{A, B, C, D\} \equiv \left\{\begin{bmatrix} 0 & 1 \\ -t_2 & -t_1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right\}$

Quartet for $\overline{\theta}$: $\{A, B, C, D\} \equiv \left\{\begin{bmatrix} O_{4x4} & 0 \\ 0 & 0 \end{bmatrix}, -sign(\theta^*{}_r)\Gamma, \begin{bmatrix} I_{4x4} & 0 \\ 0 & 1 \end{bmatrix}, O_{5x5}\right\}$

Simulink model is then constructed and given in Figure 5 below:

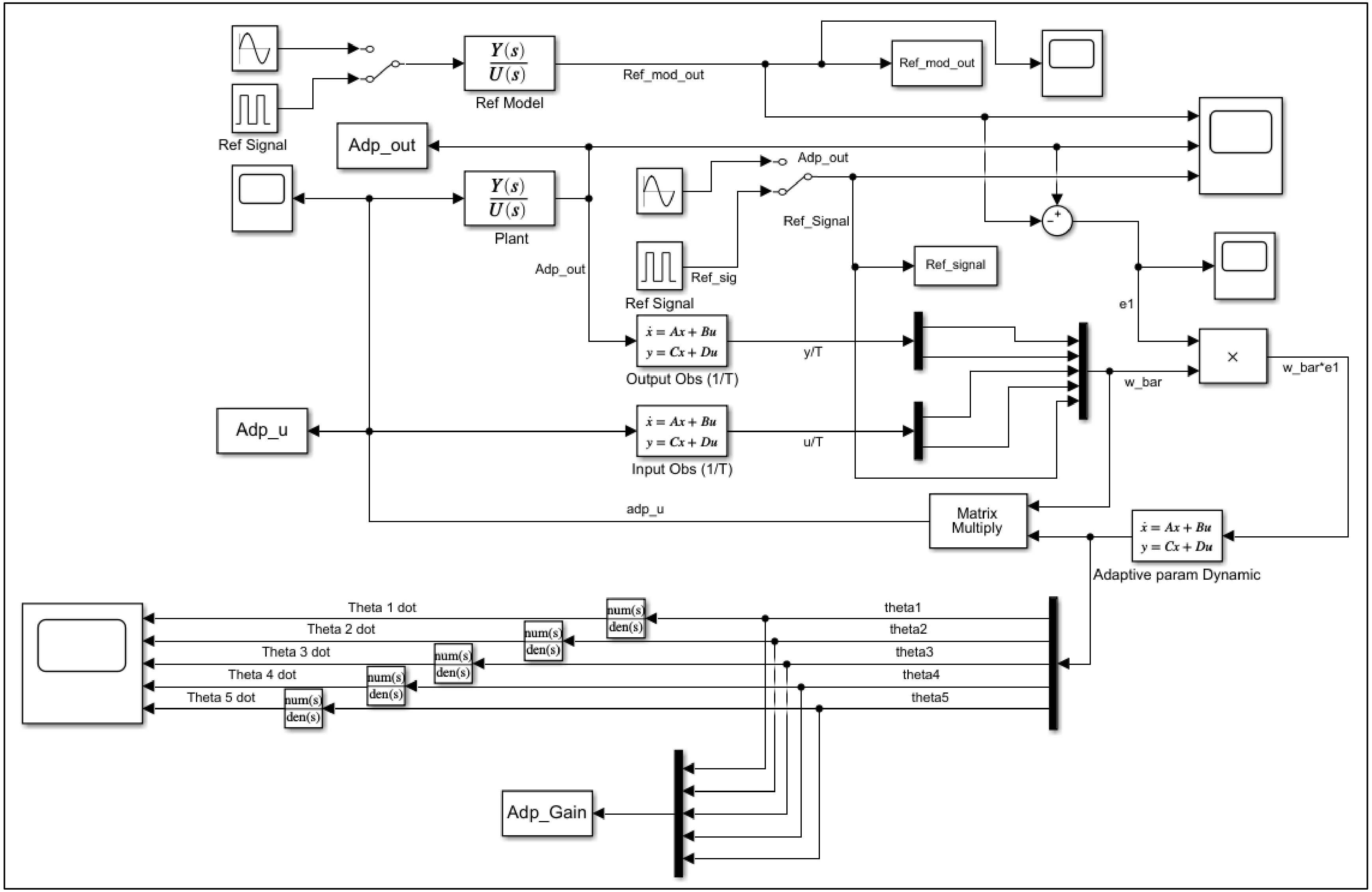


Figure 5 Simulink Model Structure for Adaptive Controller Implementation

The simulation is limited to the first 100 seconds. The reference signal generated is square wave $r(t) = 0.5 + 0.5\mathrm{sgn}(\sin(2\pi 0.05t))$, with period of 20 seconds and 50% duty cycle as noted in the wave equation. Following figures are showing various signals obtained after simulation:

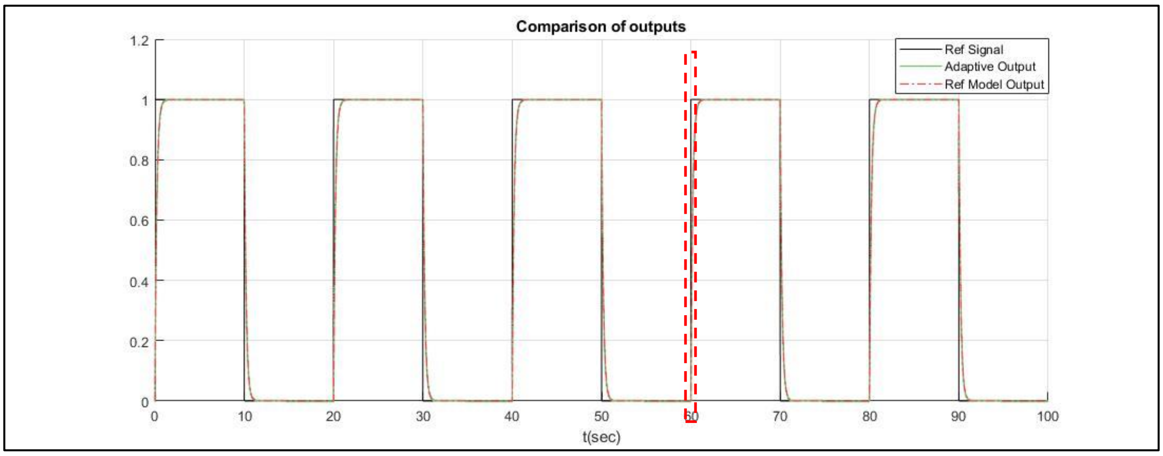


Figure 6 Comparison of reference signal $r(t)$, output of reference model $y_m(t)$ and plant's output with adaptive controller $y_p(t)$.

Transient response as noted by dotted box in Figure 6 is given below:

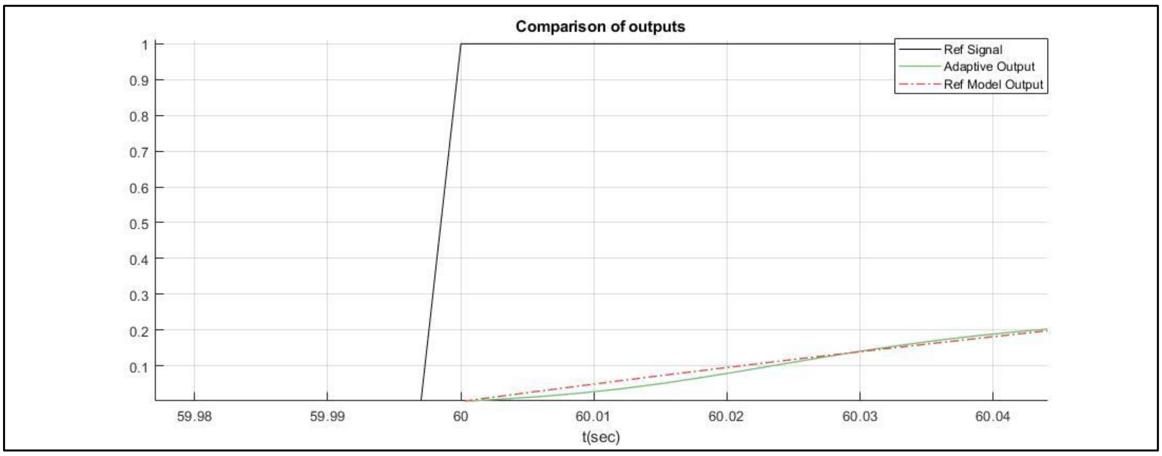


Figure 7 Transient response of adaptively controlled system around t=60s when the adaptive parameters are in steady state.

Observe that plant's output has lightly damped tracking towards reference model output and asymptotically equalizes reference model output as well as reference signal.

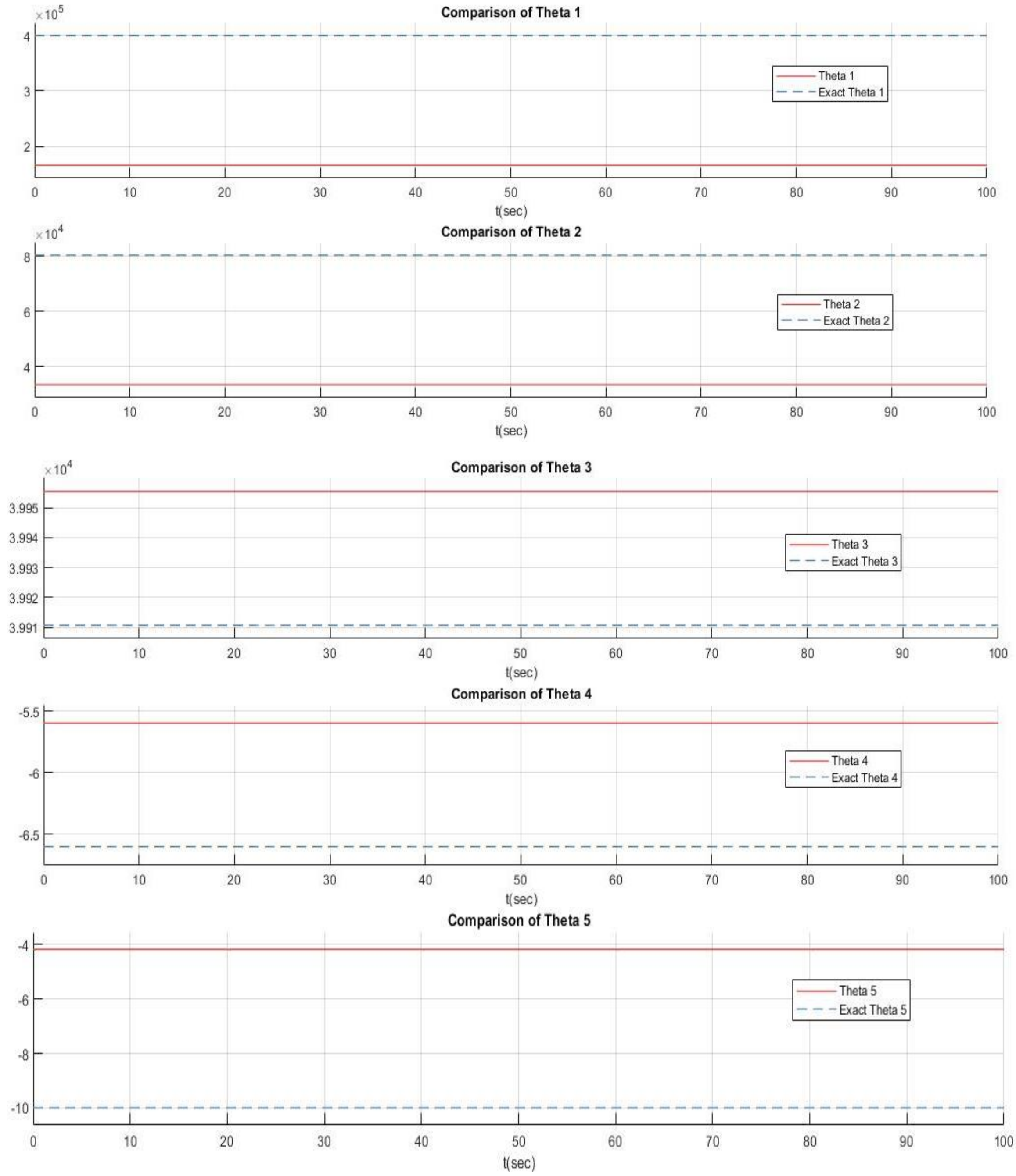


Figure 8 Adaptive Parameter comparison with those calculated from true plant (exact adaptive parameter).

It is noted that adaptive parameter evolution is not significant starting from the initial time. The initial adaptive parameters calculated earlier are maintained over time. The reason of parameter stagnation is due to the closeness of output response y_p of controlled plant against the output of reference model y_m. As a result, size of e_1 is too small to drive a significant change in the changing rate of parameter, as reflected in equation $\begin{bmatrix} \dot{\theta} \\ \dot{\theta}_r \end{bmatrix} = -sign(\theta^*{}_r)\Gamma \begin{bmatrix} \omega \\ r \end{bmatrix} e_1$. Figures shown below display the movement of e_1 and $\begin{bmatrix} \omega \\ r \end{bmatrix}$ over time.

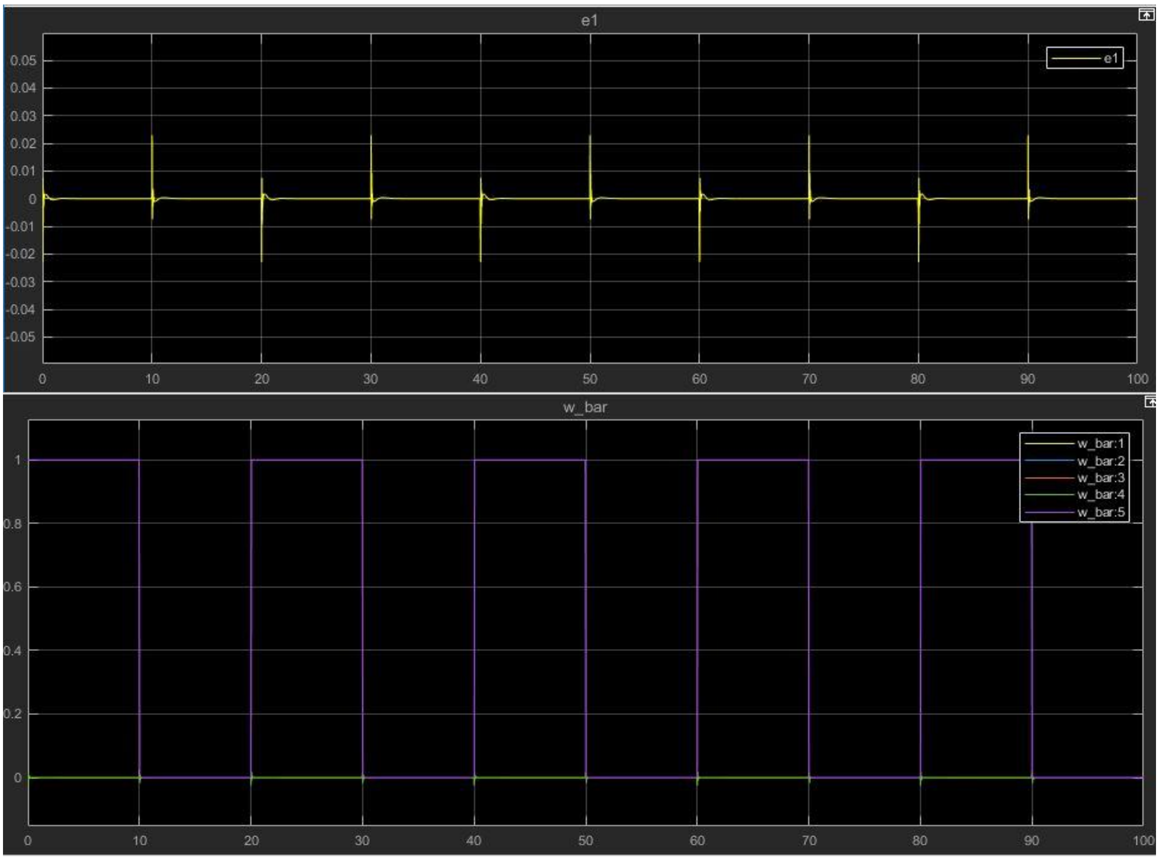


Figure 9 Output error e_1 and non-minimal states $\overline{\omega}$ are too small to drive changes in adaptive parameter $\dot{\overline{\theta}}$.

C. Selection of Observer Polynomial $T(s)$ and Adaptation Gain Γ

Note that non-minimal states ω are generated by observer plant T(s) from controlled plant input u and output y_p. $\frac{1}{T(s)}$ can be thought as a second order transfer function to generate non-minimal states as its outputs.

Consider the relationships between u, ω_u, ω_y and y_p:

$$\begin{bmatrix}\omega_y\\ \omega_u\end{bmatrix} = \frac{1}{T(s)}\begin{bmatrix}y_p\\ u\end{bmatrix} = \frac{1}{s^2+t_1 s+t_2}\begin{bmatrix}y_p\\ u\end{bmatrix}$$

These values are then passed into system of parameter adaptation with dynamic that has been found earlier:

$$\begin{bmatrix}\dot{\theta}\\ \dot{\theta}_r\end{bmatrix} = -sign(\theta^*{}_r)\Gamma\overline{\omega}e_1$$

Whose cumulative value over time $\begin{bmatrix}\theta\\ \theta_r\end{bmatrix} = \int\begin{bmatrix}\dot{\theta}\\ \dot{\theta}_r\end{bmatrix}$ drives the size of control signal $u = \begin{bmatrix}\theta\\ \theta_r\end{bmatrix}^T \overline{\omega}$, with multiplicative contribution from augmented states $\overline{\omega}(\omega_u, \omega_y, r)$.

Recall the plant's transfer function, equation (6):

$$R_p y_p = k_p Z_p u \rightarrow R_p y_p = k_p Z_p \begin{bmatrix} \theta \\ \theta_r \end{bmatrix}^T \overline{\omega} \ldots\ldots\ldots\ldots (23)$$

Since both $\overline{\omega}$ and $\begin{bmatrix} \theta \\ \theta_r \end{bmatrix}$ always have T at the denumerator, transfer function from non-minimal states $\overline{\omega}$ to y_p are affected by the roots of R_p and T. Thus, T must be chosen from pools of stable polynomial (having no roots at RHP). And its roots are set in such a way that $\frac{1}{T(s)}$ response is much faster than $\frac{1}{\mathrm{R_p(s)}}$. In other words, roots of $\mathrm{T(s)}$ are set far to the left of roots of $\mathrm{R_p(s)}$ to ensure that the dynamic of $\mathrm{R_p(s)}$ is more dominant than dynamic in $\mathrm{T(s)}$ in a term that includes $\frac{1}{\mathrm{R_p(s)}T(s)}$, which is implicitly in the plant's transfer function, equation (23).

Intuitively, setting up poles of $\mathrm{T(s)}$ to be on the far left of poles of $\mathrm{R_p(s)}$ let the non-minimal ω states converge faster to temporary steady state values (it's temporary because of assuming the reference signal is in oscillating fashion with proper period). This leads to reduction in oscillation time of adaptation parameter according to equation $\begin{bmatrix} \dot{\theta} \\ \dot{\theta}_r \end{bmatrix} = -sign(\theta^*{}_r)\Gamma\overline{\omega}e_1$.

However, $\mathrm{R_p(s)}$ are unknown in this case which will set difficulty to set the roots of $\mathrm{T(s)}$. Setting up too close will cause $\mathrm{R_p(s)}$ becoming not dominant, while setting up too large causes the initial adaptive parameter too high as Diophantine's solution depends on coefficients of $\mathrm{T(s)}$ as noted earlier:

$$\therefore \begin{cases} \mathrm{E_1} = 5 + \mathrm{t_1} - 0.4 \\ F_1 = 5t_1 + t_2 - 0.4E_1 - 4 \\ F_2 = 5t_2 - 4E_1 \end{cases}$$

If those solution polynomials are made of large values, then the initial parameter:

$$\overline{\theta}(0)^T = [-f_2(0) \quad -f_1(0) \quad -g_2(0) \quad -g_1(0) \quad k(0)]$$

will be consequently large too. At the eventual, this causes larger size of control signal following equation $u = \begin{bmatrix} \theta \\ \theta_r \end{bmatrix}^T \overline{\omega}$.

One strategy to determine the location of the roots of $\mathrm{T(s)}$ is by using approximated plant's poles to compare.

Recall the transfer function of approximated plant:

$$\mathrm{G_g(s)} = \frac{-1.2\mathrm{s} - 1.2}{\mathrm{s}^2 + 0.4\mathrm{s} + 4}$$

Its poles are located at: $\mathrm{s}_{g1,2} = -\zeta\omega_{gn} \pm \omega_{gn}\sqrt{\zeta^2 - 1} \cong -0.2 \pm \mathrm{j}2;\ \omega_{\mathrm{gn}} = 2;\ \zeta = 0.1;$

Taking the approximated plant's poles as reference to set up the roots of $\mathrm{T(s)}$ will enable us to verify the effectiveness of adaptive control for unknown true plant $\mathrm{G(s)}$.

Various trials below contrast the dynamic of non-minimal states $\bar{\omega}$, adaptive parameter $\begin{bmatrix} \theta \\ \theta_r \end{bmatrix}$, control signal u and corresponding output error $\mathrm{e}_1 = \mathrm{y_p} - y_m$ with respect to the location of roots of $\mathrm{T(s)}$.

a) Roots of $\mathrm{T(s)}$ set at $5\mathrm{s}_{g1,2} \rightarrow \mathrm{T(s)} = \mathrm{s}^2 + 2\mathrm{s} + 100$

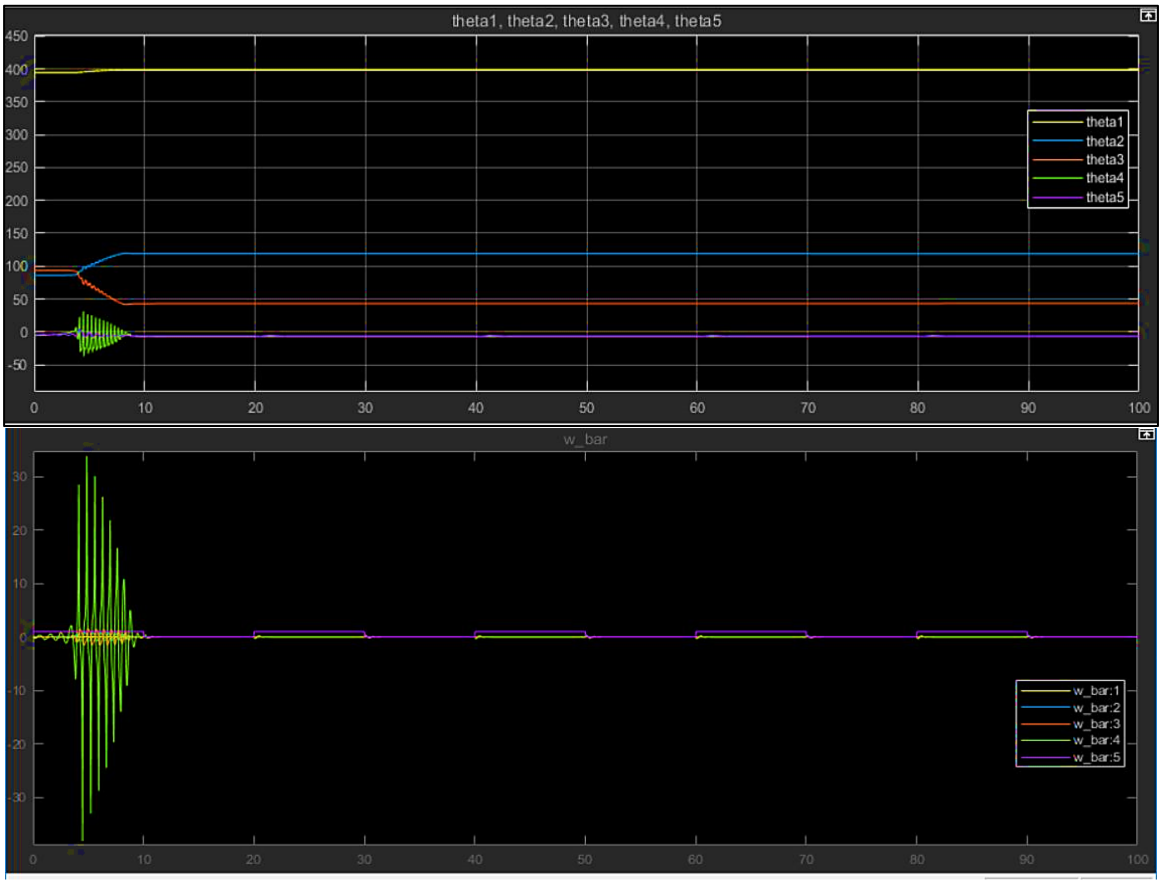


Figure 10 Adaptive parameter (top) and non-minimal states (bottom) for $5S_{g1,2}$.

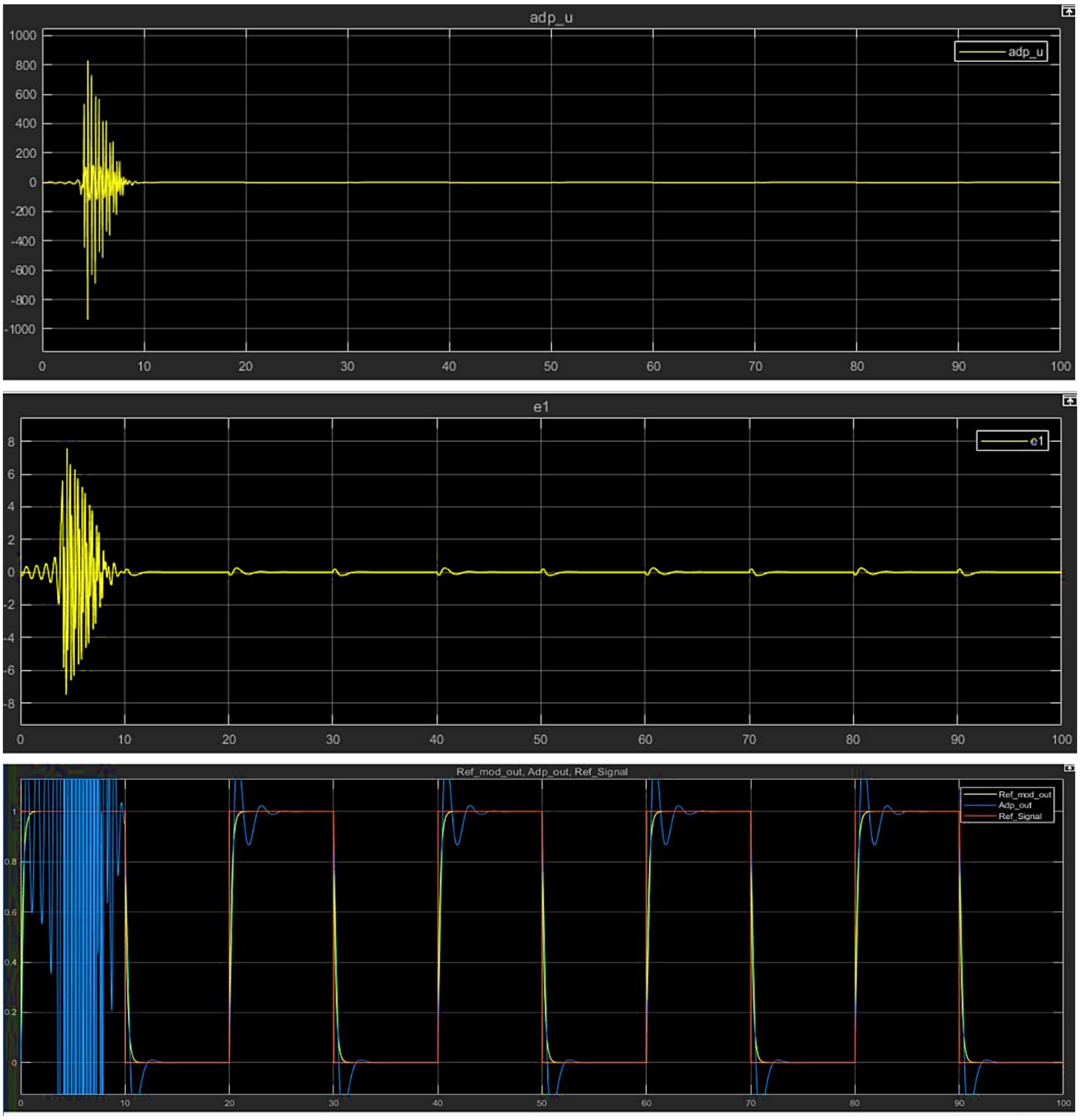


Figure 11 Control signal (top), output error e_1 and output y_p for $5S_{g1,2}$.

When $\mathrm{T}(s) = s^2 + 2s + 100$, the initial adaptive parameters are:

$$\bar{\theta}(0)^T = [394.6667 \;\; 86.1333 \;\; 93.4000 \;\; -5.6000 \;\; -4.1667]$$

Notice that output plant significantly oscillated in the first 10 seconds, the range of runtime in which the adaptive parameter is still adapting as shown in Figure 10 (top). The noticeably long adaptation is due to the location of roots of $\mathrm{T}(s)$ are not far enough (too slow) to let the plant's poles become dominant in the whole dynamic. This leads to long adaptation transient in non-minimal states $\bar{\omega}$ as shown in Figure 10 bottom. To reduce the transient time/adaptation time of adaptive parameter, roots of $\mathrm{T}(s)$ need to be set further away.

b) Roots of $T(s)$ set at $50s_{g1,2} \rightarrow T(s) = s^2 + 20s + 10000$

In this case, the initial adaptive parameters are:

$$\overline{\theta}(0)^T = [41585.0 \quad 8405.1 \quad 9975.4 \quad -5.6 \quad -4.1667]$$

Obviously much larger than previous case $5s_{g1,2}$, as suggested in page 22.

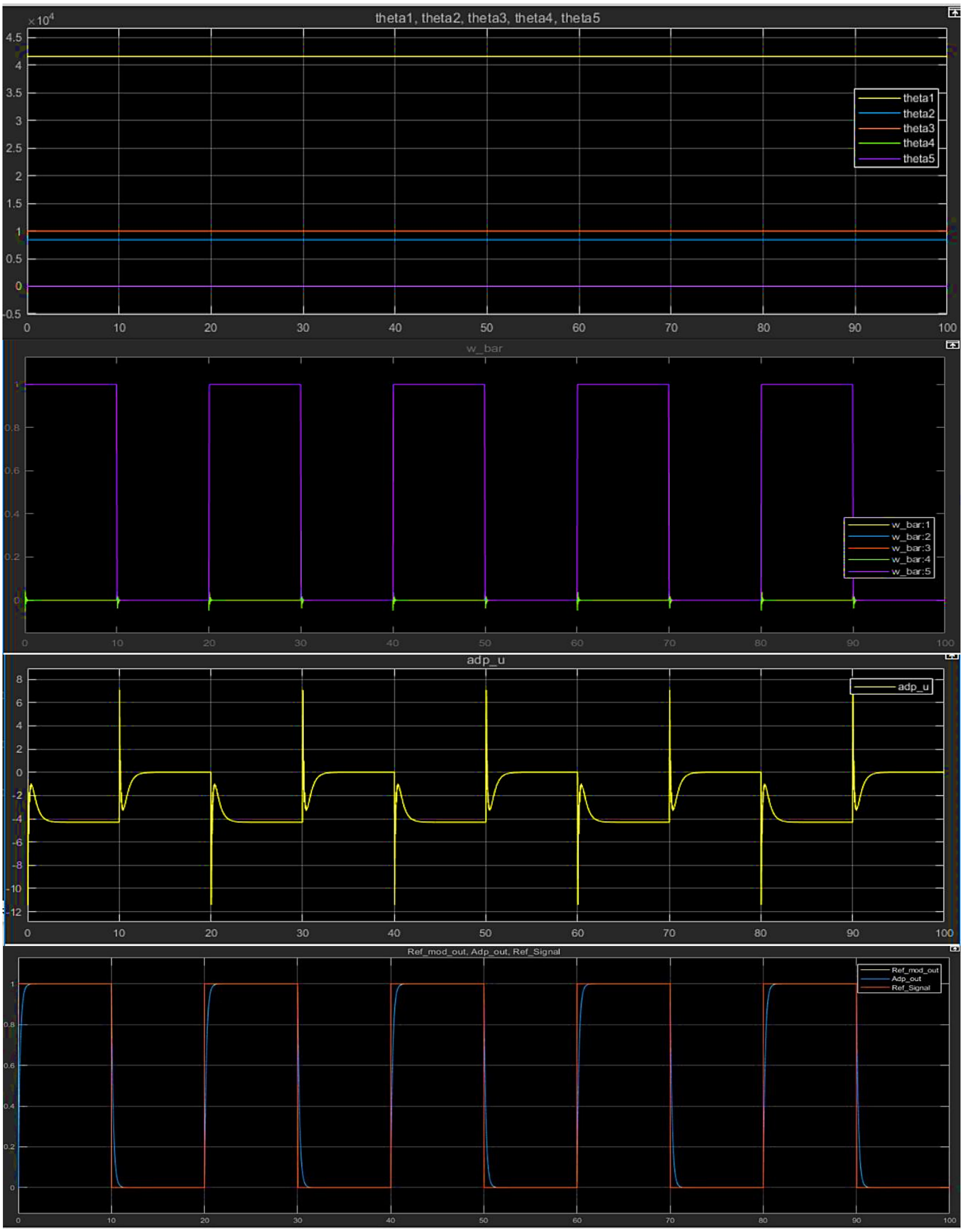


Figure 12 Top to bottom: Adaptive parameter $\overline{\theta}$, non-minimal states $\overline{\omega}$, control signal and output for $50s_{g1,2}$

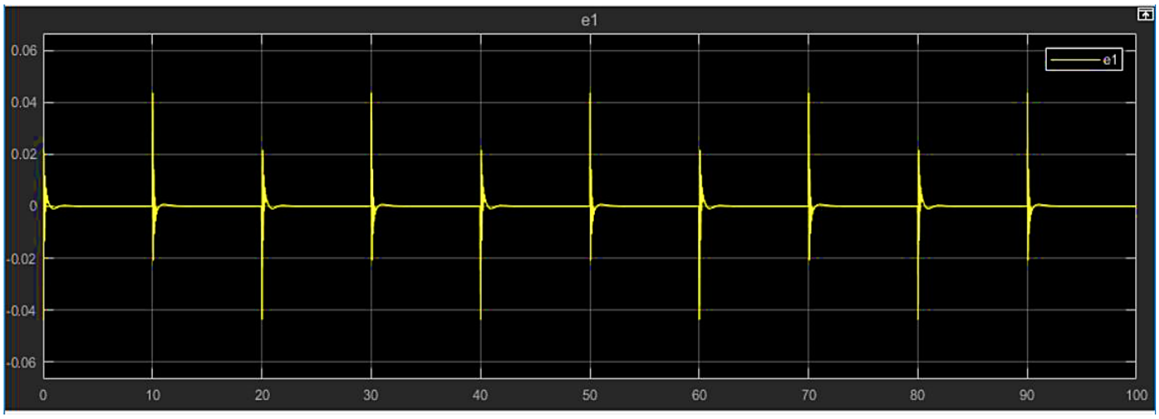


Figure 13 Output error e_1 for case $50s_{g1,2}$.

Parameter adaptation takes much shorter, in fact it is unnoticeable. Fast adaptation adjusts the adaptive parameter to steady state faster and hence results in control signal having less transient compared to case $5s_{g1,2}$. It is noted that control signal has range around [-11,7] in a regular form (periodic form), this follows in the shape of output y_p (bottom) in Figure 12, where it now asymptotically tracks reference signal. To know if larger poles to even lead to better overall result, case $500s_{g1,2}$ is examined below.

c) Roots of $T(s)$ set at $500s_{g1,2}$ $\rightarrow T(s) = s^2 + 200s + 1000000$

At this setting, the initial adaptive parameter is extremely large compared to previous two cases:

$$\bar{\theta}(0)^T = [4165984.67 \quad 834095.133 \quad 999795.4 \quad -5.6 \quad -4.16666667]$$

Also observe that the last two parameters ($g_1(0)$ & $k(0)$) are always the same for all cases. Note that $g_1(0) = (E_1 + 1 - t_1)$ while $E_1 = 5 + t_1 - 0.4$ from Diophantine equation for the specific approximated plant discussed in this report. Substitute E_1 into $g_1(0)$, the term t_1 vanishes and leaving 6-0.4=5.6. For parameter $k(0)$, it is obvious to see that $k(0) = \frac{k_m}{k_g}$ does not involve any terms in observer polynomial T(s). Hence, changing $T(s)$ will only affect the first three parameters, and in the case where $T(s) = s^2 + 200s + 1000000$, the results are presented below:

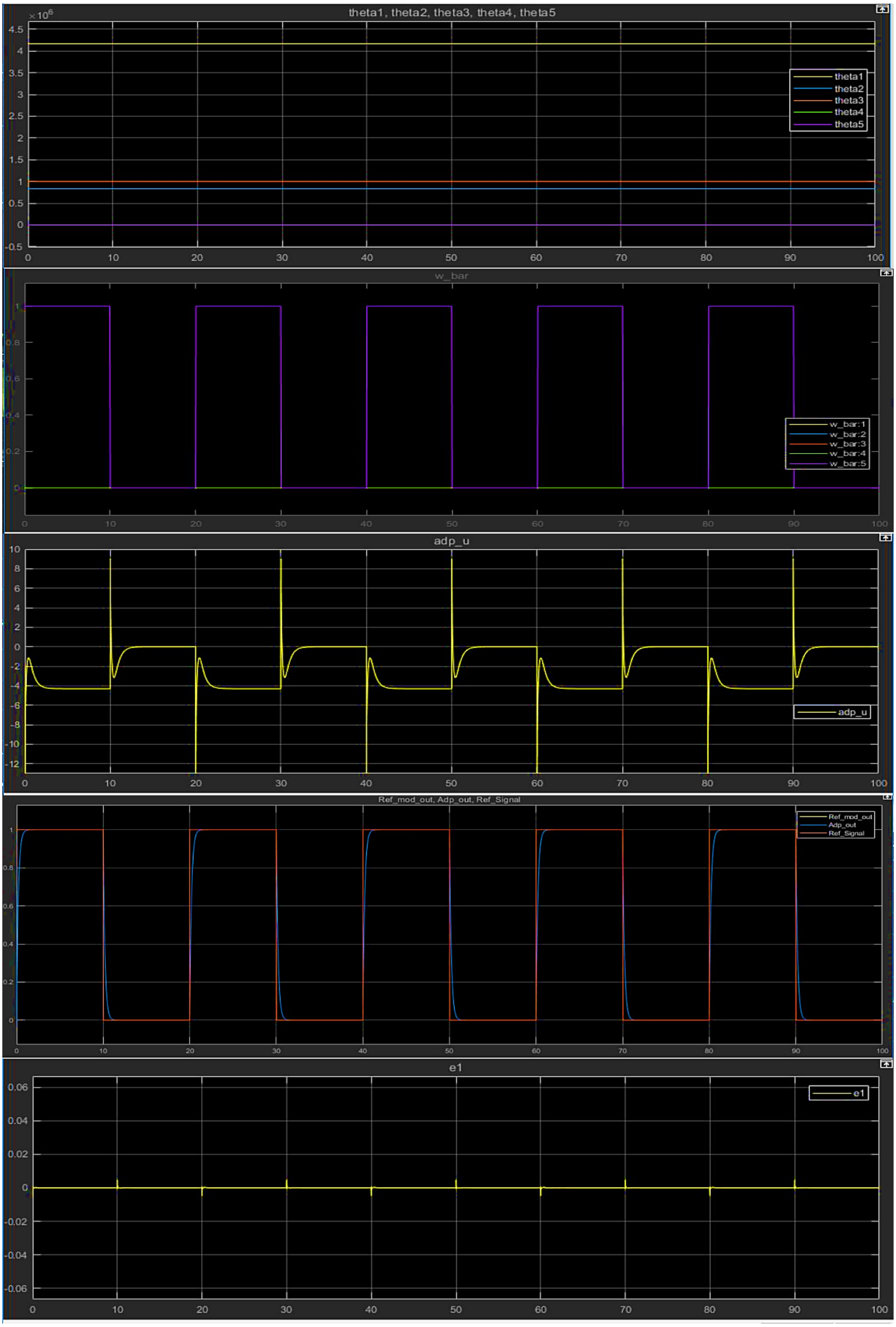


Figure 14 Top to bottom: Adaptive parameter $\overline{\theta}$, non-minimal states $\overline{\omega}$, control signal, output y_p and output error e_1 for $500S_{g1,2}$.

On top of larger initial adaptive parameter, control signal u has grown larger taking values in range [-13,9] with similar period form, which has been noted in page 22. But it helps in terms of reduction of output error e_1 as shown in last entry of Figure 14.

All these observations indicate that there is a trade-off between the size of control signal u, adaptive parameter $\overline{\theta}$ and output error e_1r. The trade-off composition is affected by many factors and one of them is the roots of T(s). Placing the roots of T(s) to be too close with the plant's poles causes longer adaptation time and higher output error mainly due to the dynamic in non-minimal states $\overline{\omega}$ that has longer adaptation transient. However, the control signal required is lower once adaptive parameter has converged.

On the other side, setting the roots of polynomial observer T(s) much further away from the original plant's poles allow the plant's poles to be dominant in a way that convergence (steady state) in non-minimal states $\overline{\omega}$ is faster, resulting in reduction of adaptation transient time and adaptive parameter achieves relative convergence faster. In addition, the output error e_1 is observed lower if the root location is set further. The setback of setting such T is that the initial adaptive parameter is large, and results in higher control signal being generated.

The setting of T that is considered the best one in this project is $T(s) = s + 40s + 4 \times 10^4$, which has been used in calculation earlier in part II.A.

D. Selection of Adaptive Gain Γ

Similar to observer polynomial, value of adaptive gain result in different response or the dynamic of various signals in the system. To check the effect of adaptive gain, a polynomial observer has to be chosen and fixed, therefore the setting that has been established earlier will be used, that is $T(s) = s + 40s + 4 \times 10^4$.

a) Adaptive Gains Set At $\Gamma = 10I_{5x5}$

Adaptive gain Γ does not affect the initial adaptive parameter values as they are calculated based on approximated plant's transfer function and observer polynomial as noted earlier. At this point, it is known fact that adaptive gain Γ affects the size of change rate of adaptive parameter as shown in equation (19).

After simulation, figures below present the movement of various signals in the system.

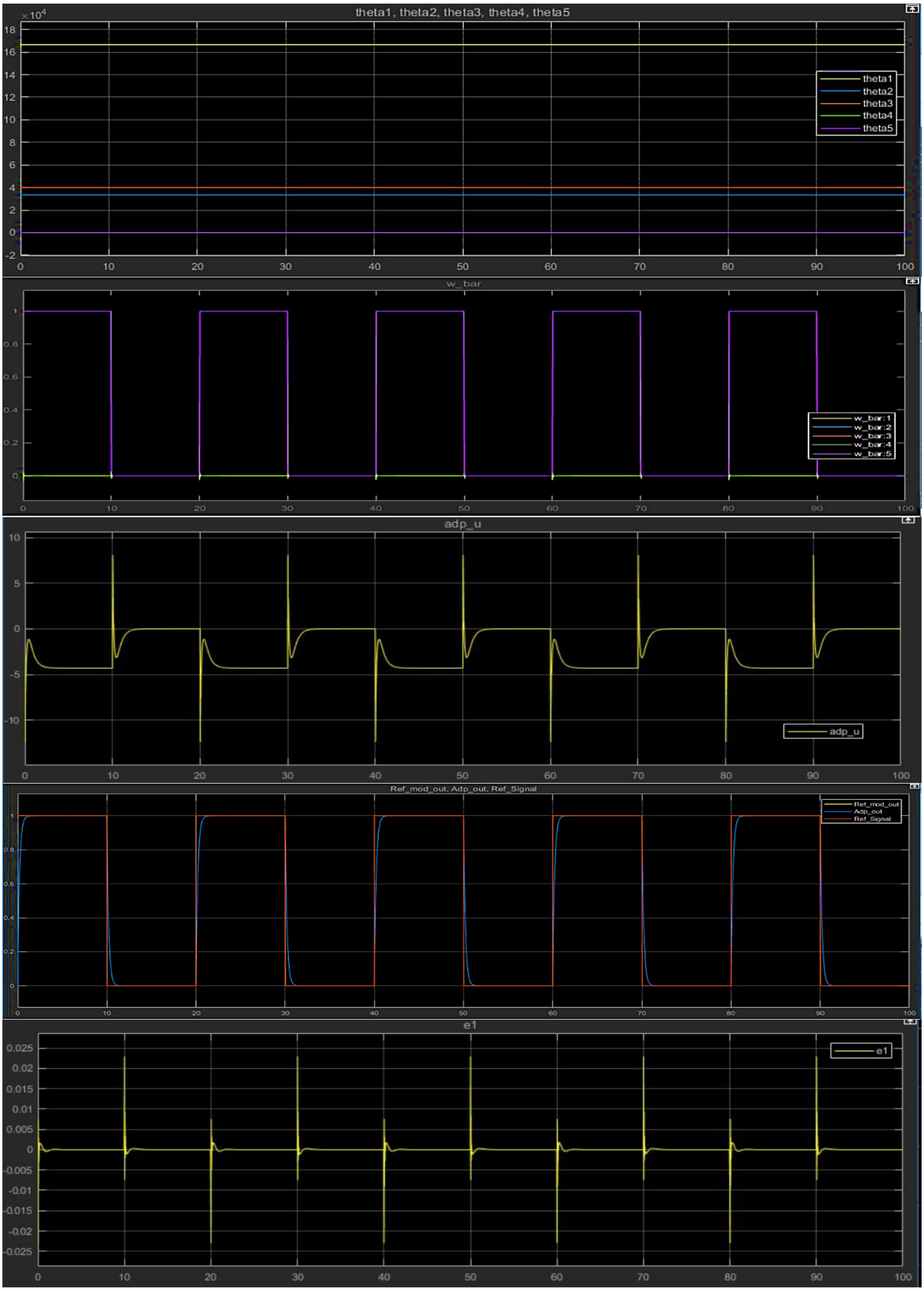


Figure 15 Top to bottom: Adaptive parameter $\overline{\theta}$, non-minimal states $\overline{\omega}$, control signal, output y_p and output error e_1 for $\Gamma = 10I_{5x5}$.

b) Adaptive Gains Set At $\Gamma = 1000I_{5x5}$

Results:

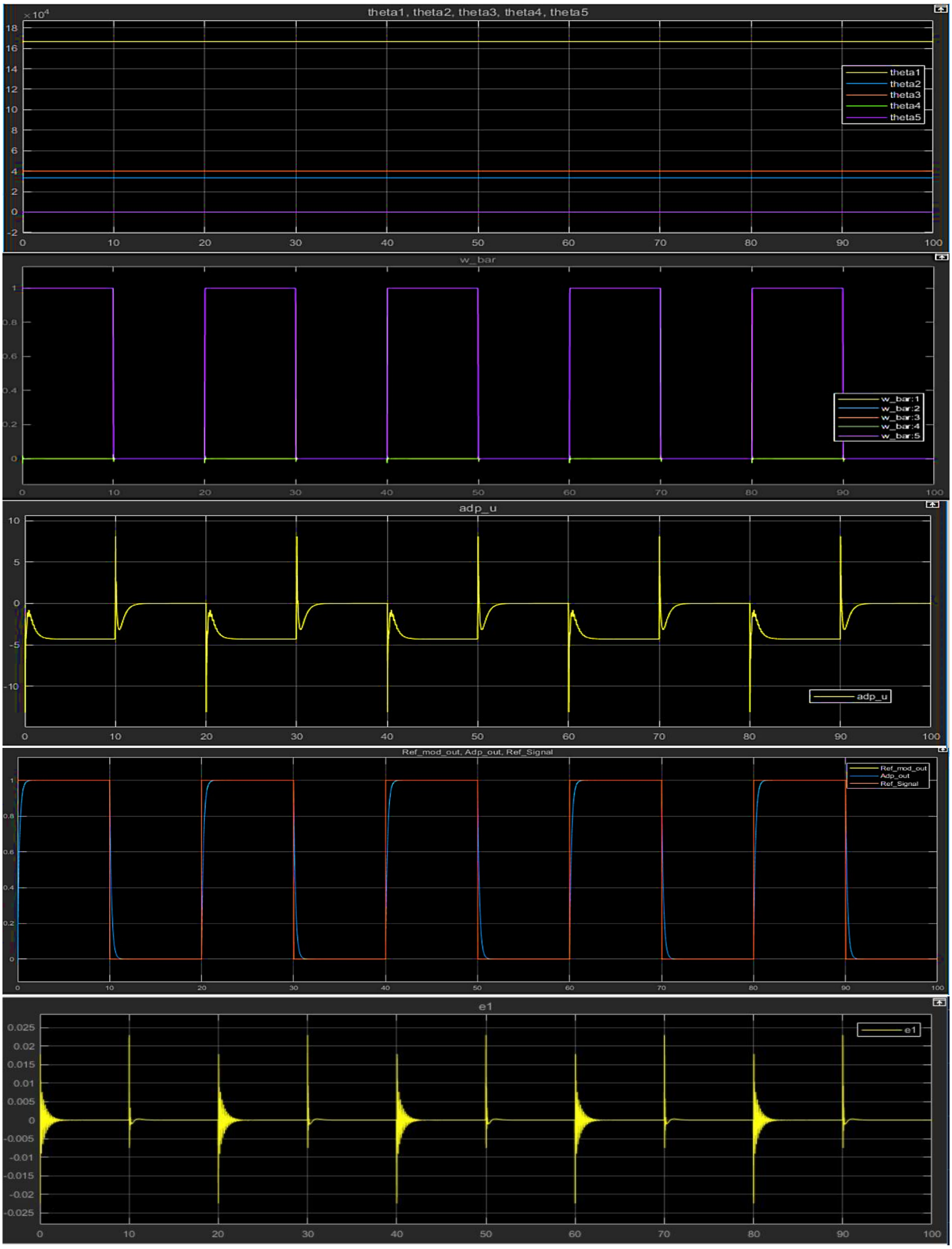


Figure 16 Top to bottom: Adaptive parameter $\overline{\theta}$, non-minimal states $\overline{\omega}$, control signal, output y_p and output error e_1 for $\Gamma = 1000I_{5x5}$.

Note that the output error e_1 starts degrading indicated by more oscillation.

c) Adaptive Gains Set At $\Gamma = 0.1 I_{5x5}$

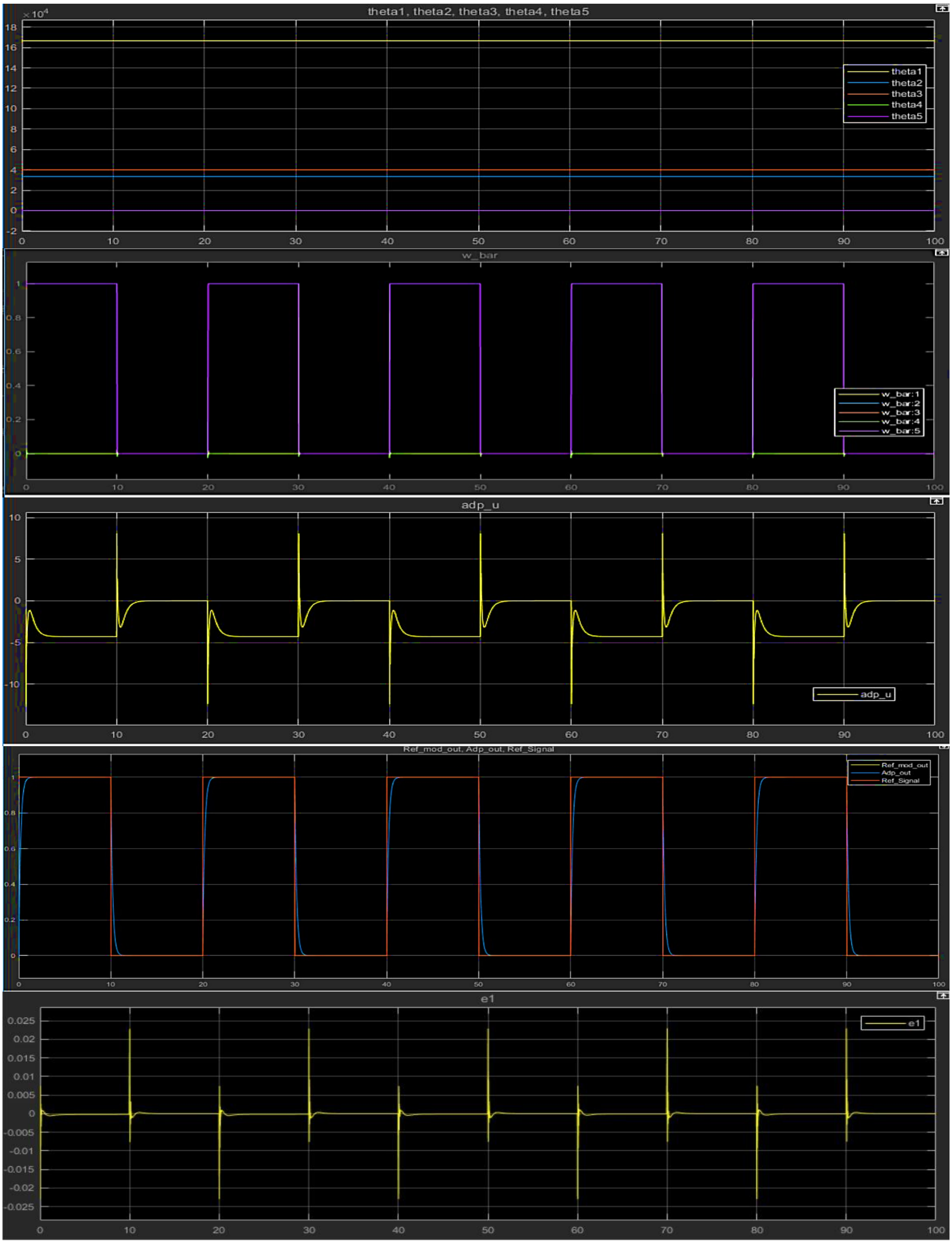


Figure 17 Top to bottom: Adaptive parameter $\overline{\theta}$, non-minimal states $\overline{\omega}$, control signal, output y_p and output error e_1 for $\Gamma = 0.1 I_{5x5}$.

These signals when $\Gamma = 0.1 I_{5x5}$ doesn't differ significantly from the case when $\Gamma = 10 I_{5x5}$. However, output error e_1 is showing more oscillation during increasing edge of reference signal when adaptive gain Γ is set high ($1000 I_{5x5}$), which is unfavourable in the system.

III. Simulation with Sinusoidal Reference Signal

The reference model that is used throughout previous simulation is first order which is mostly used to track a step signal. Thus, it makes sense to employ first order system as reference model to implement adaptive controller when the reference signal is step signal. In this particular case, a sinusoidal reference signal is to be simulated using the adaptive system that has been established, with polynomial observer $T(s) = s^2 + 40s + 40000$, and $\Gamma = 10$.

Result of simulations are presented in collection of figures below:

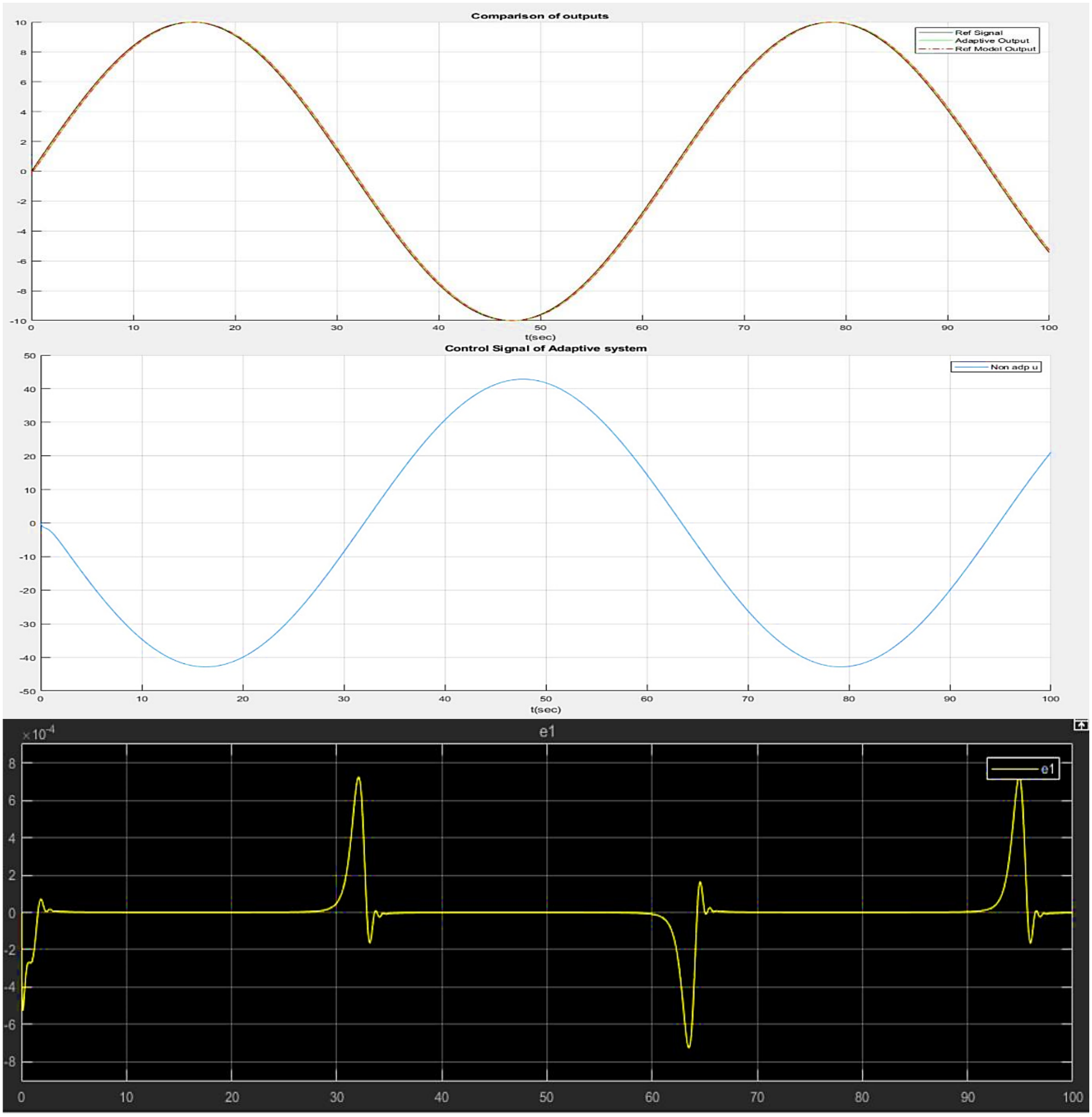


Figure 18 Top to bottom: Output, control signal and output error

The output error for the reference signal 10sin(0.1t) is small but when the output waveform is inspected at magnification, there is delay observed between reference signal and output of plant with adaptive controller, as shown below:

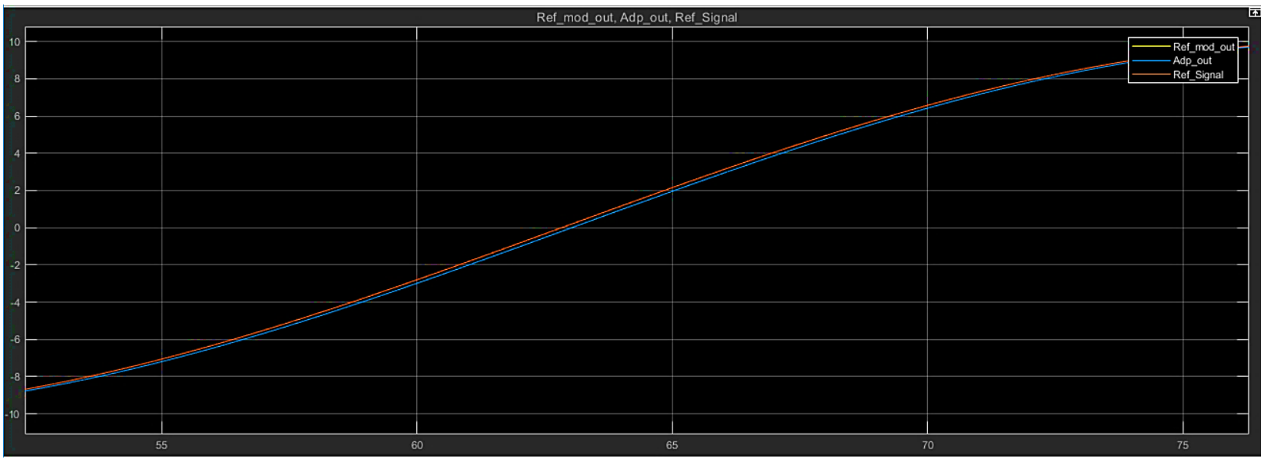


Figure 19 Delay between reference sinusoid and adaptively controlled plant output.

The delay is understandable because the reference model used is a first order system. There is always microscopic delay between them so long as the first order system being used as reference model. One might decrease the time constant of first order reference model to better track the sinusoidal reference signal (lessen the delay), as long as the delay is acceptable and within specification then choosing first order system as reference model is viable.

Other than that, it is observed that adaptive parameter in this sinusoidal tracking is subtly similar with those in step signal tracking (square wave as reference input with appropriately long period). The values of these adaptive parameter can be seen in figures below:

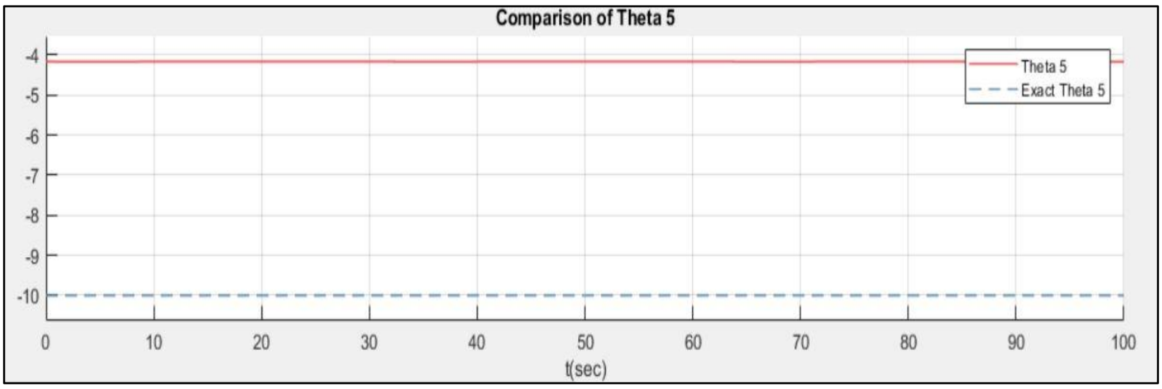


Figure 20 Adaptive parameter for reference input, stays at -4.167

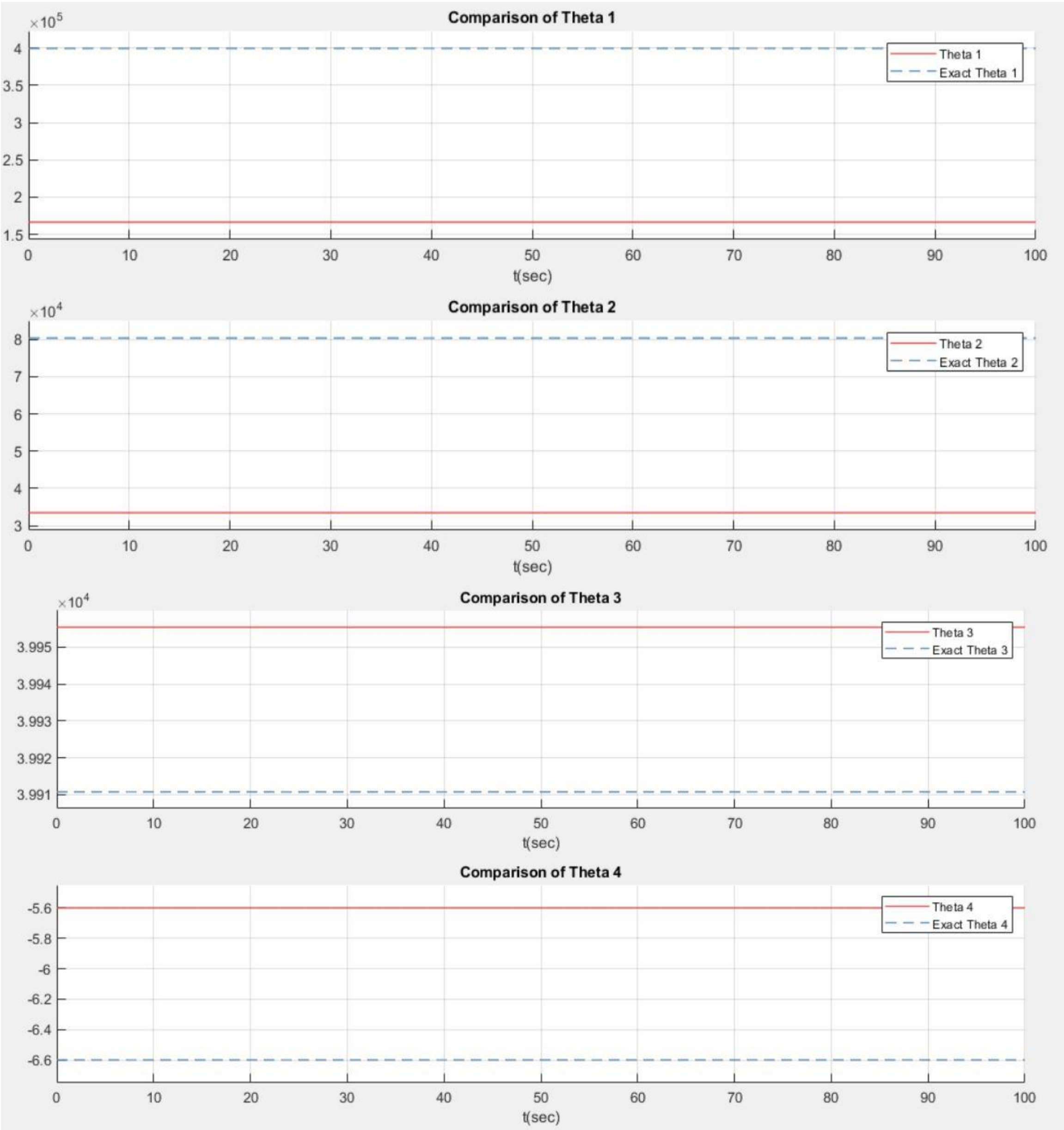


Figure 21 Comparison of adaptive parameter θ_1 until θ_4, which corresponds to $[-g_2 - g_1 - f_2 - f_1]$.

The adaptive parameter $\overline{\theta}$ are not affected by the sinusoidal reference signal. The initial adaptive parameter $\overline{\theta}(0)$ calculated based on approximated plant and polynomial observer is maintained over time. Hence, adaptive control implemented in the plant works for both step and sinusoidal signals with the same adaptive parameter and same initial adaptive parameter values. However, there is slight delay observed between reference sinusoid and output plant, which its acceptance is solely dependent on the minimal performance specification defined, if any.

IV. Conclusion

Implementation of adaptive control in the true system which has unknown form $G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s+b_1}{s^2+a_1 s+a_2}$ noted in previous chapter without having knowledge of exact transfer function of the plant is possible. The information given about the true plant's zero location, damping level and natural frequency assisted the approximation of plant's transfer function $G_g(s)$ that is in the second order form. The approximation is then used to calculate initial augmented adaptive parameter (controller gain) $\bar{\theta}(0)$. Using the dynamic adaptation law of this controller gain, given as $\dot{\bar{\theta}} = -\text{sign}(\theta^*{}_r)\Gamma\bar{\omega}e_1$, which is equivalent to $\dot{\bar{\theta}} = -\text{sign}\left(k_p\right)\Gamma\bar{\omega}e_1$ since $k_m > 0$ and $\theta^*{}_r = \frac{k_m}{k_p}$ and sign of k_p is known from the zero location that is given (zero is at LHP, thus plant's DC gain sign is negative). One may consider setting the roots of polynomial non-minimal observer T(s) to be on far left from the approximated plant's poles in the complex plane to ensure the plant's poles are more dominant than the poles of observer and reduce the duration of adaptation with the cost of control signal getting larger. On the other hand, letting Γ value to be too large affect the output error causing it to be more 'undamped' during transition of step reference.

Overall, the adaptive controlled system is able to deal with uncertainty in the true plant's transfer function. An approximated plant derived from the characteristic of open-loop output response can be utilized to ignite adaptation of controller gain whose dynamic depends on non-minimal observer with only input and output being measurable.

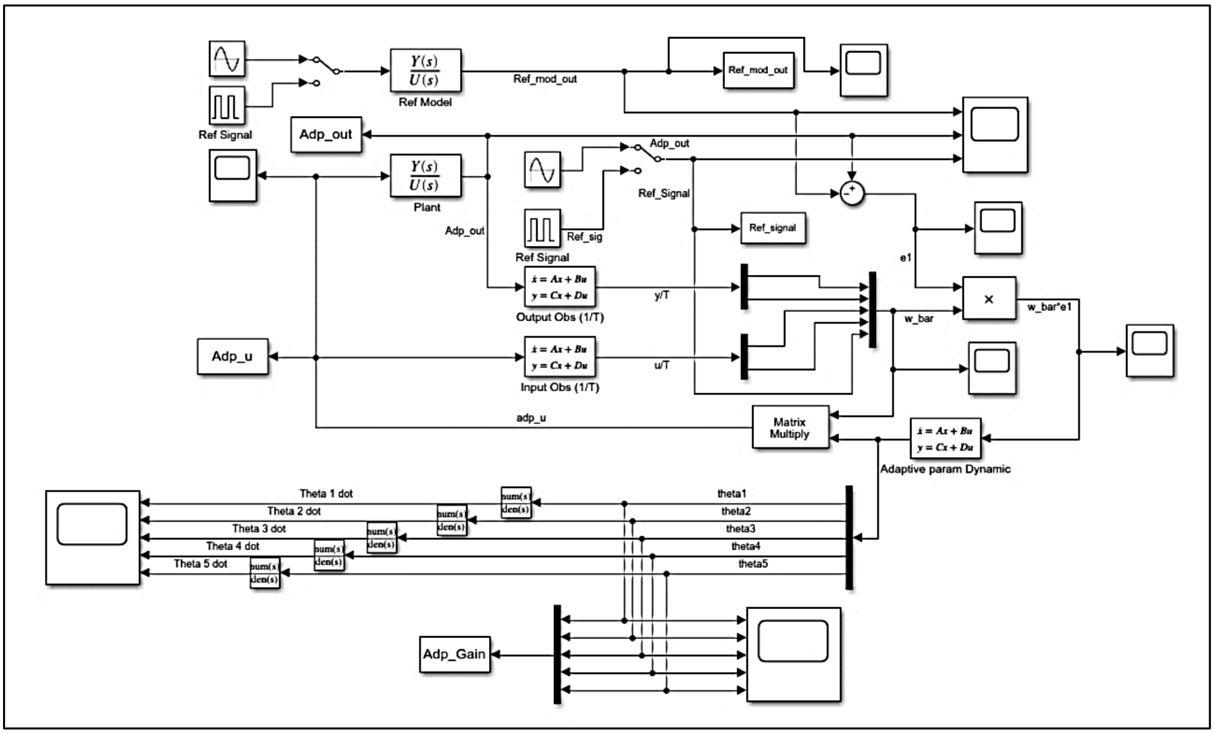
V. Appendix: Code and Model

A. Adaptive Controller Code

```
%---------------------------------------------------------
% Adaptive Control, where Plant parameter are unknown
%---------------------------------------------------------
clear;clc;

%% Known parameter of True Plant
True_plant_Zp = [1 2];
True_plant_Kp = -0.5;
True_plant_Rp = [1 0.4 4.3];
True_plant = tf([True_plant_Kp*True_plant_Zp],[True_plant_Rp]);

%% Reference Model
time_const = 0.2;
Ref_plant_Rm = [1 1/time_const];
Ref_plant_Km = 1/time_const;
Ref_plant = tf(Ref_plant_Km,[Ref_plant_Rm]);

%% Guessed Plant to get initial value of adaptive parameter
Nat_freq = 2;
Damp_rat = 0.1;
Guessed_plant_Kp = -0.3*(Nat_freq^2);
Guessed_plant_Zp = [1 1];
Guessed_plant_Rp = [1 2*Damp_rat*Nat_freq Nat_freq^2];
Guessed_plant = tf(Guessed_plant_Zp*(Guessed_plant_Kp),Guessed_plant_Rp);
tsim = (0:0.01:100);
usim = [0 ones(1,size(tsim,2)-1)*10]';
figure;
hold on;
lsim(Guessed_plant,usim,tsim);
grid;
axis([0 50 -9 2]);
hold off;
roots(Guessed_plant_Rp)

%% Diophantine Solution & Adaptive Gains
% Choosing Parameter for Non Minimal filter T
push_root = 100; %to set the roots of T, multiple away from plant's poles
dampw = push_root*Nat_freq;
dampT = 0.1;
t1 = 2*dampT*dampw;
t2 = dampw^2;
T_filt = [1 t1 t2];
roots(T_filt)

% Finding polynom E,F,F_bar,G_bar,G1,and Kr of guessed plant
E = [1 1/time_const + t1 - Guessed_plant_Rp(2)];
F = [t2 + (1/time_const)*t1 - Guessed_plant_Rp(3) -
Guessed_plant_Rp(2)*E(2) (1/time_const)*t2 - Guessed_plant_Rp(3)*E(2)];
G_bar = conv(E,Guessed_plant_Zp);
F_bar = F/Guessed_plant_Kp;
G1 = G_bar - T_filt;
G1 = G1(2:3);

% Initial Adaptive Parameter
Kr = Ref_plant_Km/Guessed_plant_Kp;
```

```
Theta_init_bar = [(-1)*fliplr(F_bar) (-1)*fliplr(G1) Kr];

%% State Space of Non Minimal filter T & Gamma Setting
% Input Filter State Space, Filter output = [wu, swu]
AT_filt_U = [0 1;-t2 -t1];
BT_filt_U = [0;1];
CT_filt_U = [1 0; 0 1];
DT_filt_U = [0;0];

% Output Filter State Space, Filter output = [wy, swy]
AT_filt_Y = [0 1;-t2 -t1];
BT_filt_Y = [0;1];
CT_filt_Y = [1 0; 0 1];
DT_filt_Y = [0;0];

% Gamma setting
Gamma = 10*[1 0 0 0 0;0 1 0 0 0;0 0 1 0 0;0 0 0 1 0;0 0 0 0 1];

%% Plant's dynamic through non-minimal states
nm_A = [0 1 0 0;
    (-1*Guessed_plant_Rp(3)) (-1*Guessed_plant_Rp(1))
Guessed_plant_Zp(2)*Guessed_plant_Kp Guessed_plant_Zp(1)*Guessed_plant_Kp;
    0 0 0 1;
    0 0 -1*T_filt(3) -1*T_filt(2)];
nm_B = [0 0 0 1]';
nm_C = [T_filt(3)-Guessed_plant_Rp(3) T_filt(2)-Guessed_plant_Rp(2)
Guessed_plant_Zp(2)*Guessed_plant_Kp
Guessed_plant_Zp(1)*Guessed_plant_Kp]';
syms s;
non_minimal_tf = (nm_C'/(s*eye(4)-nm_A))*nm_B;
[nonum, noden] = numden(non_minimal_tf);

%% Dynamic of Adaptive Parameter (State Transition Matrices)

Atheta = zeros(5,5);
Btheta = -1*sign(Guessed_plant_Kp)*Gamma;
Ctheta = eye(5);
Dtheta = zeros(5,5);

%% Getting exact adaptive parameter for comparison purposes

T_filt_exact = [1 t1 t2];
E_exact = [1 1/time_const + t1 - True_plant_Rp(2)];
F_exact = [t2 + (1/time_const)*t1 - True_plant_Rp(3) -
True_plant_Rp(2)*E(2) (1/time_const)*t2 - True_plant_Rp(3)*E(2)];

% Theta star vector ,ie : exact [-f2,-f1,-g2,-g1]
G_bar_exact = conv(E_exact,True_plant_Zp);
F_bar_exact = F/True_plant_Kp;
G1_exact = G_bar_exact - T_filt_exact;
G1_exact = G1_exact(2:3);

% Non-adaptive reference gain
Kstar_R_exact = Ref_plant_Km/True_plant_Kp;

% Theta bar star
Theta_star_bar = [(-1)*fliplr(F_bar_exact) (-1)*fliplr(G1_exact)
Kstar_R_exact];
```

B. Adaptive Control Simulation



File Model:


Adaptive_Simulation.slx


Adaptive_Simulation.slxc

Note : Use save as, and import to matlab workspace to simulate.

C. Plot for Various Signals Obtained During Simulation

```
%-------------------------------------------------------
% Adaptive Control, Plotting output of Simulink
%-------------------------------------------------------
clc;
%% Plotting output (comparison between : reference model, non-adaptive
model (or
% we call it exact parameter model), and reference signal)
RGB = [0.3467 0.5360 0.6907;0.9153 0.2816 0.2878;0.4416 0.7490 0.4322];
figure;
hold on;
grid;
plot(Ref_signal.Time, Ref_signal.Data,'-','Color','k','Linewidth',0.5);
plot(Adp_out.Time, Adp_out.Data,'-
x','Color',RGB(3,:),'LineWidth',1,'MarkerSize',1);
plot(Ref_signal.Time, Ref_mod_out.Data,'-
.','Color',RGB(2,:),'LineWidth',1,'MarkerSize',2);
hold off;
legend('Ref Signal','Adaptive Output','Ref Model Output');
title('Comparison of outputs');
xlabel('t(sec)');

%% Plotting of control signal from adaptive model
figure;
hold on;
grid;
plot(Ref_signal.Time, Adp_u.Data,'Color',RGB(1,:));
hold off;
legend('Adp u');
title('Control Signal of Adaptive system');
xlabel('t(sec)');

%% Plotting of Adaptive gains and its comparison against exact adaptive
gains
figure;

% Plotting theta 1, ie: -f2
subplot(2,1,1);grid;hold on;
plot(Adp_Gain.Time, Adp_Gain.Data(:,1),'-','Color',RGB(2,:),'Linewidth',1);
plot(Adp_Gain.Time, Theta_star_bar(1)*ones(size(Adp_Gain.Data(:,1),1),1),'-
-','Color',RGB(1,:),'LineWidth',1,'MarkerSize',1);
xlabel('t(sec)');
title('Comparison of Theta 1');
legend('Theta 1','Exact Theta 1');
range = max(max(Adp_Gain.Data(:,1),Theta_star_bar(1))) -
min(min(Adp_Gain.Data(:,1),Theta_star_bar(1))) + 0.5;
axis([Adp_Gain.Time(1) Adp_Gain.Time(end)
(min(min(Adp_Gain.Data(:,1),Theta_star_bar(1)))-0.1*range)
(max(max(Adp_Gain.Data(:,1),Theta_star_bar(1)))+0.1*range)]);
hold off;

% Plotting theta 1, ie: -f1
subplot(2,1,2);grid;hold on;
plot(Adp_Gain.Time, Adp_Gain.Data(:,2),'-','Color',RGB(2,:),'Linewidth',1);
plot(Adp_Gain.Time, Theta_star_bar(2)*ones(size(Adp_Gain.Data(:,2),1),1),'-
-','Color',RGB(1,:),'LineWidth',1,'MarkerSize',1);
xlabel('t(sec)');
title('Comparison of Theta 2');
legend('Theta 2','Exact Theta 2');
```

```
range = max(max(Adp_Gain.Data(:,2),Theta_star_bar(2))) -
min(min(Adp_Gain.Data(:,2),Theta_star_bar(2))) + 0.5;
axis([Adp_Gain.Time(1) Adp_Gain.Time(end)
min(min(Adp_Gain.Data(:,2),Theta_star_bar(2))) - 0.1*range
max(max(Adp_Gain.Data(:,2),Theta_star_bar(2))) + 0.1*range]);
hold off;

figure;

% Plotting theta 3, ie: -g2
subplot(2,1,1);grid;hold on;
plot(Adp_Gain.Time, Adp_Gain.Data(:,3),'-','Color',RGB(2,:),'Linewidth',1);
plot(Adp_Gain.Time, Theta_star_bar(3)*ones(size(Adp_Gain.Data(:,3),1),1),'-
-','Color',RGB(1,:),'LineWidth',1,'MarkerSize',1);
xlabel('t(sec)');
title('Comparison of Theta 3');
legend('Theta 3','Exact Theta 3');
range = max(max(Adp_Gain.Data(:,3),Theta_star_bar(3))) -
min(min(Adp_Gain.Data(:,3),Theta_star_bar(3))) + 0.5;
axis([Adp_Gain.Time(1) Adp_Gain.Time(end)
min(min(Adp_Gain.Data(:,3),Theta_star_bar(3))) - 0.1*range
max(max(Adp_Gain.Data(:,3),Theta_star_bar(3))) + 0.1*range]);
hold off;

% Plotting theta 4, ie: -g1
subplot(2,1,2);grid;hold on;
plot(Adp_Gain.Time, Adp_Gain.Data(:,4),'-','Color',RGB(2,:),'Linewidth',1);
plot(Adp_Gain.Time, Theta_star_bar(4)*ones(size(Adp_Gain.Data(:,4),1),1),'-
-','Color',RGB(1,:),'LineWidth',1,'MarkerSize',3);
xlabel('t(sec)');
title('Comparison of Theta 4');
legend('Theta 4','Exact Theta 4');
range = max(max(Adp_Gain.Data(:,4),Theta_star_bar(4))) -
min(min(Adp_Gain.Data(:,4),Theta_star_bar(4))) + 0.5;
axis([Adp_Gain.Time(1) Adp_Gain.Time(end)
min(min(Adp_Gain.Data(:,4),Theta_star_bar(4))) - 0.1*range
max(max(Adp_Gain.Data(:,4),Theta_star_bar(4))) + 0.1*range]);
hold off;

% Plotting theta 5, ie: K
figure;hold on;grid;
plot(Adp_Gain.Time, Adp_Gain.Data(:,5),'-','Color',RGB(2,:),'Linewidth',1);
plot(Adp_Gain.Time, Theta_star_bar(5)*ones(size(Adp_Gain.Data(:,5),1),1),'-
-','Color',RGB(1,:),'LineWidth',1,'MarkerSize',1);
xlabel('t(sec)');
title('Comparison of Theta 5');
legend('Theta 5','Exact Theta 5');
range = max(max(Adp_Gain.Data(:,5),Theta_star_bar(5))) -
min(min(Adp_Gain.Data(:,5),Theta_star_bar(5))) + 0.5;
axis([Adp_Gain.Time(1) Adp_Gain.Time(end)
min(min(Adp_Gain.Data(:,5),Theta_star_bar(5))) - 0.1*range
max(max(Adp_Gain.Data(:,5),Theta_star_bar(5))) + 0.1*range]);
hold off;
```