Simulation & Modeling

Online Parameter Estimation

Petridis Konstantinos 9403

April 10, 2022

Abstract

This report exhibits the versatility and adaptability of **Online Parameter Estimation** methods. Different types of **models**, their sensitivity to noise and their estimation accuracy are also topics of interest, which are presented in detail in corresponding sections. The testing outcomes and the visual proof provided by graphical illustrations, demonstrate the effectiveness of parameter estimation, performed in real time.

1 Introduction

In previous report and simulations, we covered the case of **Offline Parameter Estimation**, which happens after the input-output measurements are gathered. Based on this philosophy, our estimator is able to minimize the **square error** across all the collected values, estimating the unknown parameters as the solution to the optimization problem, using **Least Squares**. However, in real systems, the the ability to approximate the unknown parameters in **real time** is often required. This means that, parameters and states of the model are computed when new data is available during the operation of the physical system. The two methods of **Online Parameter Estimation** which we focus on, are the **Gradient Descent** and **Lyapunov** method. The latter, is applied both on **Parallel Model (P)** and **Series-Parallel Model (SP)** configurations.

2 Gradient Descent

2.1 Theoretical Analysis

The **differential equation** of the system is given by

$$\dot{y} = -ay + bu \quad (1)$$

with the initial condition $y(0) = 0$. In order to transform the system to its linear parametric form, the following operations are applied. Let $a_m > 0$ be an arbitrarily chosen parameter.

$$\dot{y} = a_m y - a_m y - ay + bu \iff$$

$$\dot{y} = (a_m - a)y - a_m y + bu \iff$$

$$(s + a_m)y = (a_m - a)y + bu \iff$$

$$y = \frac{1}{s + a_m}\Big[(a_m - a)y + bu\Big]$$

which is decomposed to

$$\theta^* = \begin{bmatrix} a_m - a & b \end{bmatrix} \quad (2)$$

$$\phi = \begin{bmatrix} \frac{1}{s + a_m}y & \frac{1}{s + a_m}u \end{bmatrix} \quad (3)$$

Combining $(2), (3)$ we get the desired formulation of the system.

$$y = \theta^{*T}\phi \quad (4)$$

which is the **linear parametric** form of the **real system**. The corresponding form of the model should be

$$\hat{y} = \hat{\theta}^T \phi \quad (5)$$

The next step is to compute the **error** between the system and the model.

$$e = y - \hat{y}$$

Using (4), (5) we get

$$e = -\tilde{\theta}^T \phi, \quad \tilde{\theta} = \hat{\theta} - \theta^* \quad (6)$$

Intuitively we choose our objective function to be

$$K(\hat{\theta}) = \frac{1}{2} e^2 \quad (7)$$

which is convex, so the minimum is both unique and global. Minimizing this function with respect to $\hat{\theta}$, we manage to minimize the output error between the system and the model. In order to achieve this, we apply the **Gradient Descent** method, which is described by

$$\dot{\hat{\theta}} = -\gamma \nabla K(\hat{\theta}) \quad (8)$$

The slope of the objective function K, can be directly computed, using (6), (7), as

$$\nabla K(\hat{\theta}) = \nabla \Big[\frac{1}{2} (y - \hat{\theta}^T \phi)^2 \Big] \iff$$

$$\nabla K(\hat{\theta}) = -\phi(y - \hat{\theta}^T \phi) \iff$$

$$\nabla K(\hat{\theta}) = -e\phi \quad (9)$$

So substituting (9) in (8) results in

$$\dot{\hat{\theta}} = -\gamma e \phi$$

or in tabular form

$$\begin{bmatrix} \dot{\hat{\theta}}_1 \\ \dot{\hat{\theta}}_2 \end{bmatrix} = -\gamma e \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \quad (10)$$

with

$$\phi_1 = \frac{1}{s + a_m} y, \quad \phi_2 = \frac{1}{s + a_m} u \quad (11)$$

Applying **Inverse Laplace Transformation** to (11)

$$\dot{\phi}_1 = -a_m \phi_1 + y, \quad \phi_1(0) = 0$$

$$\dot{\phi}_2 = -a_m \phi_2 + y, \quad \phi_2(0) = 0$$

Eventually, all the necessary equations for the **Gradient Descent Method** to work are obtained.

2.2 Simulation

2.2.1 u=3cos(2t)

This section presents the outcome of the simulations, conducted on **system (1)**, which was modeled as described previously. The desired behavior seems to have been achieved in a satisfying degree, as we can easily observe that both the parameters approach the real values and simultaneously the error between the model and the system is lead to a negligible value. *Figure 1.1.1* particularly illustrates the phenomenon of the estimated parameters converging gradually to the real values, while *Figure 1.1.2* depicts the precise **tracking** of the real output, performed by the model. Our claims regarding the accuracy are verified by *Figure 1.1.3* and *1.1.4*. The error, decreases constantly throughout the time span of the process, eventually being almost completely eliminated (or at least not significantly noticeable).

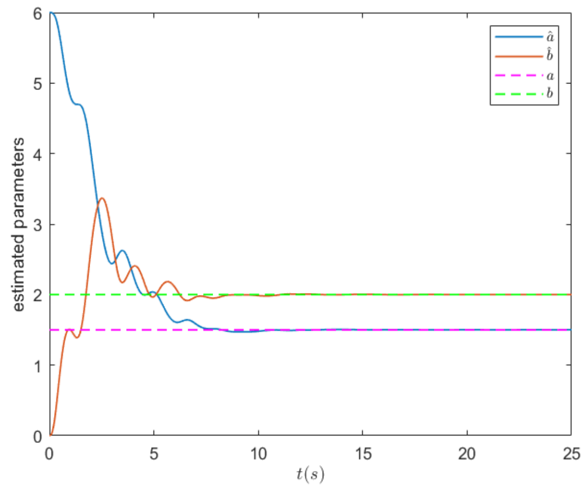

Figure 1.1.1: Estimated parameters

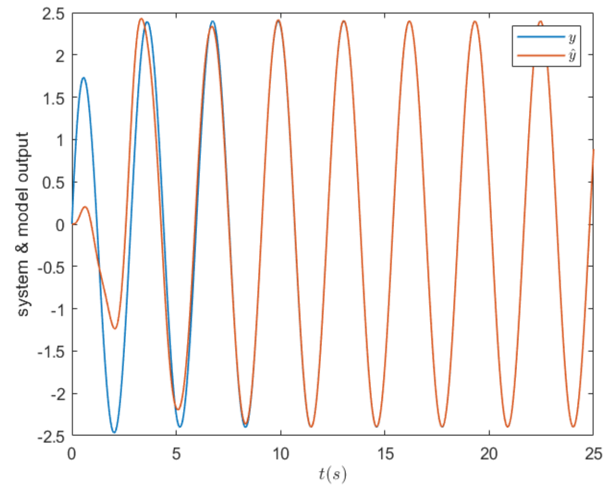

Figure 1.1.2: System & model output

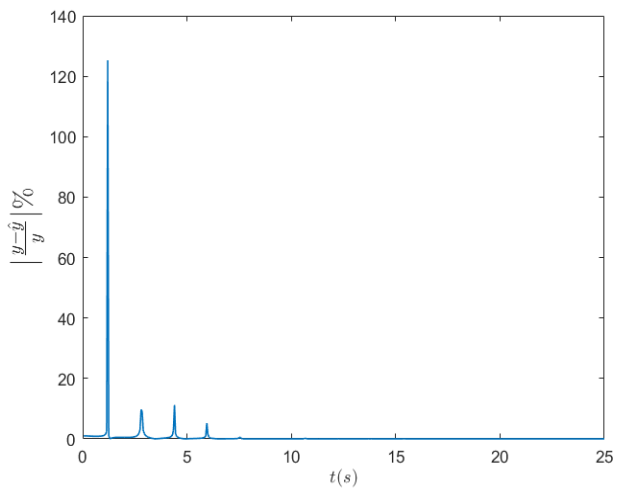

Figure 1.1.3: Percentage error

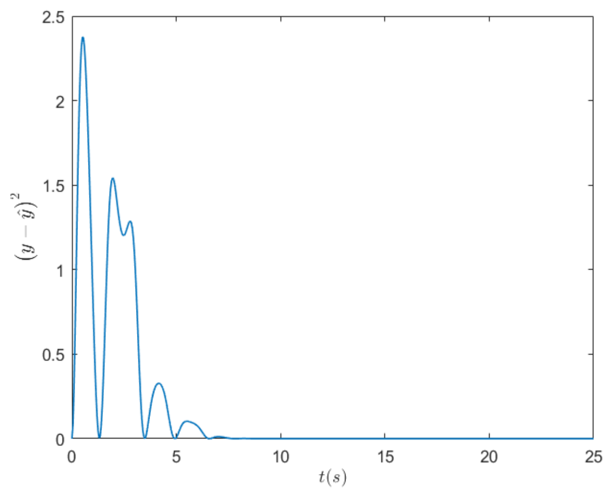

Figure 1.1.4: Mean square error

2.2.2 u=3

If we activate the system using a constant input, the results indicate a strange behavior. Specifically, the model itself seems to accurately track the system's output and decrease the error to almost zero, as depicted in *Figure 1.2.2* and *1.2.3*. However, the deviation of the estimated parameters from the actual values is very considerable, as demonstrated in *Figure 1.2.1*. This happens because, in order for both to occur (zero error and parameters converging to the real values), some conditions concerning the input, should be satisfied. The transfer function can be obtained by (1) as

$$y = H(s)u, \quad H(s) = \frac{b}{s+a}$$

For matrix H it is directly proved that $det(H) \neq 0$, which shows that, the system's input should at least contain one distinct and non-zero frequency. This condition however is not satisfied for $u = 3$, and that's the reason why the odd behavior occurs.

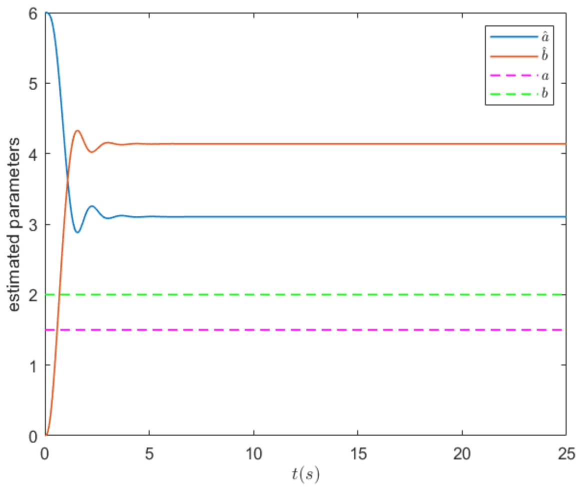

Figure 1.2.1: Estimated parameters

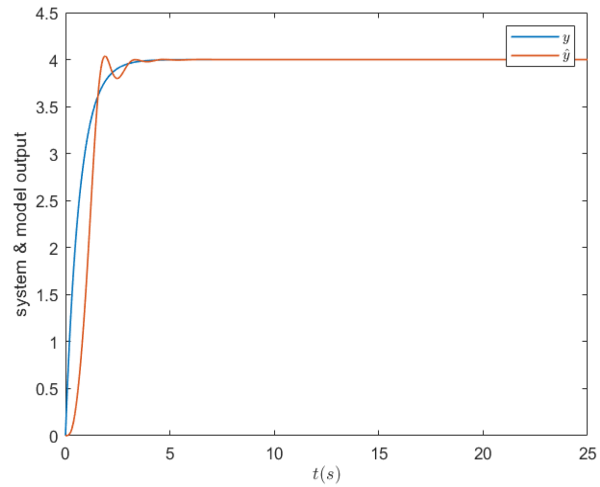

Figure 1.2.2: System & model output

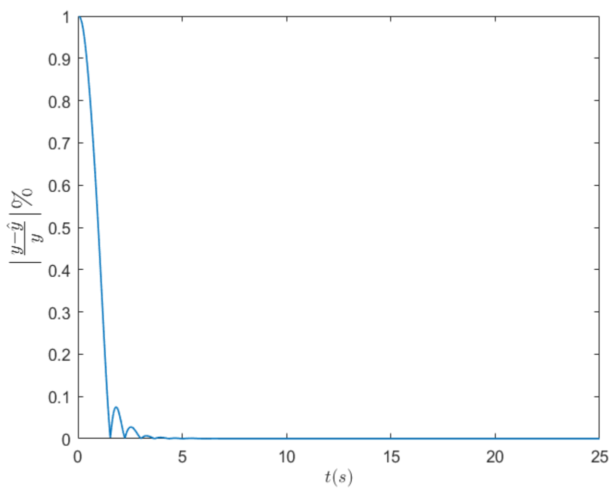

Figure 1.2.3: Percentage error

3 Lyapunov Method

In the following section, a similar analysis is going to be exhibited, now using the **Lyapunov Method** to guarantee that the error between the system and model is lead to zero. Once again we focus on **system (1)**, but we apply two different types of modeling configuration.

3.1 Parallel Model

For the **Parallel Model** configuration (P), we design our model as

$$\dot{\hat{y}} = -\hat{\theta}_1\hat{y} + \hat{\theta}_2 u, \quad \hat{\theta}_1 = \hat{a}, \hat{\theta}_2 = \hat{b}$$

3.1.1 Theoretical Analysis

Given that $e = y - \hat{y}$ and following the obvious operations, presented previously, we get

$$\dot{e} = -\theta_1^* e + \tilde{\theta}_1\hat{y} - \tilde{\theta}_2 u \quad (2.1)$$

where

$$\tilde{\theta}_1 = \hat{\theta}_1 - \theta_1^*$$

$$\tilde{\theta}_2 = \hat{\theta}_2 - \theta_2^*$$

$$e = y - \hat{y}$$

Now we choose the **Lyapunov Function**

$$V(\hat{\theta}) = \frac{1}{2}e^2 + \frac{1}{2\gamma_1}\tilde{\theta_1}^2 + \frac{1}{2\gamma_2}\tilde{\theta_2}^2 \quad (2.2)$$

which satisfies all the necessary conditions. Differentiating (2.2) results in

$$\dot{V}(\hat{\theta}) = e\dot{e} + \frac{1}{\gamma_1}\tilde{\theta}_1\dot{\hat{\theta}}_1 + \frac{1}{\gamma_2}\tilde{\theta}_2\dot{\hat{\theta}}_2$$

Substituting (2.1)

$$\dot{V}(\hat{\theta}) = -\theta_1^* e^2 + \tilde{\theta}_1\hat{y}e - \tilde{\theta}_2 ue + \frac{1}{\gamma_1}\tilde{\theta}_1\dot{\hat{\theta}}_1 + \frac{1}{\gamma_2}\tilde{\theta}_2\dot{\hat{\theta}}_2 \quad (2.3)$$

In order to eliminate the terms of indefinite sign, we choose

$$\dot{\hat{\theta}}_1 = -\gamma_1\hat{y}e \quad (2.4)$$

$$\dot{\hat{\theta}}_2 = \gamma_2 ue \quad (2.5)$$

which leaves us with the desired

$$\dot{V}(\hat{\theta}) = -\theta_1^* e^2 \leq 0$$

If our mathematical reasoning is valid, using $(2.4), (2.5)$ in our simulation, will lead to the desired outcome.

3.1.2 Simulation

Once again, the experimental outcome demonstrates the validity of our theoretical analysis. *Figure 2.1* and *2.2* depict the behavior, both of the estimated parameters and the output of the system and the model respectively. The error decreases and reaches a satisfyingly low level, as shown in *Figure 2.3* and *2.4*.

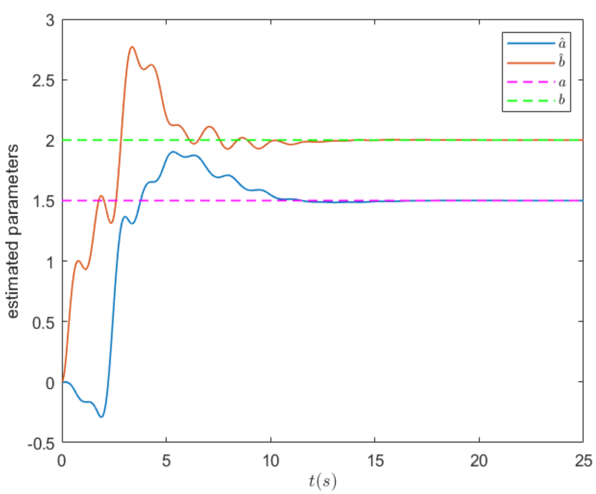

Figure 2.1: Estimated parameters

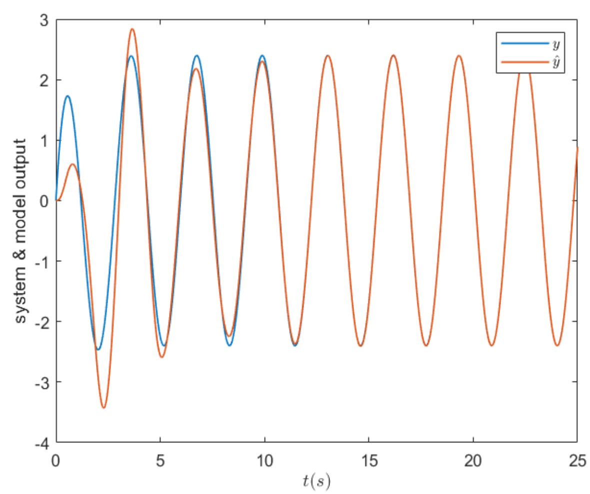

Figure 2.2: System & model output

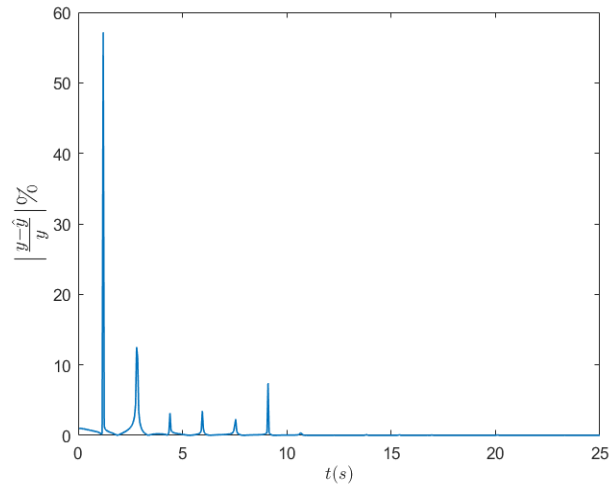

Figure 2.3: Percentage error

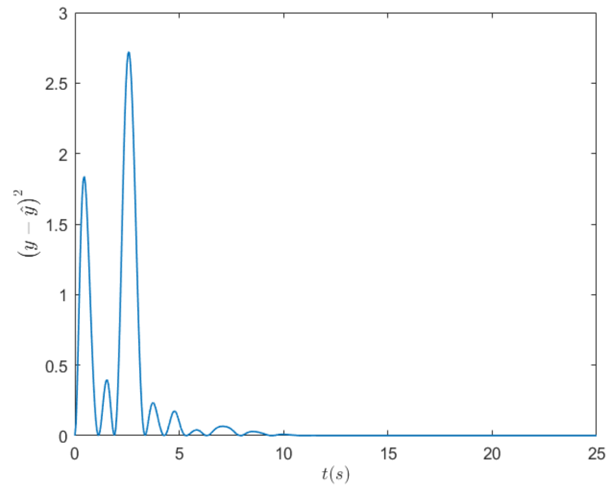

Figure 2.4: Mean square error

3.2 Series-Parallel Model

For the **Series-Parallel Model** configuration (SP), the design is the following:

$$\dot{\hat{y}} = -\hat{\theta}_1 y + \hat{\theta}_2 u + \theta_m (y - \hat{y}), \quad \hat{\theta}_1 = \hat{a}, \hat{\theta}_2 = \hat{b}$$

3.2.1 Theoretical Analysis

Repeating the procedure, as presented in the **Parallel Model** configuration but for the new model, we obtain:

$$\dot{e} = \tilde{\theta}_1 y - \tilde{\theta}_2 u - \theta_m e \quad (3.1)$$

where

$$\tilde{\theta}_1 = \hat{\theta}_1 - \theta_1^*$$

$$\tilde{\theta}_2 = \hat{\theta}_2 - \theta_2^*$$

$$e = y - \hat{y}$$

and choosing the same **Lyapunov Function**

$$V(\hat{\theta}) = \frac{1}{2}e^2 + \frac{1}{2\gamma_1}\tilde{\theta_1}^2 + \frac{1}{2\gamma_2}\tilde{\theta_2}^2$$

we get, after differentiation,

$$\dot{V}(\hat{\theta}) = e\dot{e} + \frac{1}{\gamma_1}\tilde{\theta}_1\dot{\hat{\theta}}_1 + \frac{1}{\gamma_2}\tilde{\theta}_2\dot{\hat{\theta}}_2 \iff$$

$$\dot{V}(\hat{\theta}) = -\theta_m e^2 + \tilde{\theta}_1 ye - \tilde{\theta}_2 ue + \frac{1}{\gamma_1}\tilde{\theta}_1\dot{\hat{\theta}}_1 + \frac{1}{\gamma_2}\tilde{\theta}_2\dot{\hat{\theta}}_2 \quad (3.2)$$

The indefinite sign term elimination is applied, choosing

$$\dot{\hat{\theta}}_1 = -\gamma_1 ye \quad (3.3)$$

$$\dot{\hat{\theta}}_2 = \gamma_2 ue \quad (3.4)$$

so (3.2) becomes

$$\dot{V}(\hat{\theta}) = -\theta_m e^2 \leq 0$$

3.2.2 Simulation

The results for the **Series-Parallel Model** configuration are almost identical to those of the **Parallel Model** configuration as shown in the figures below.

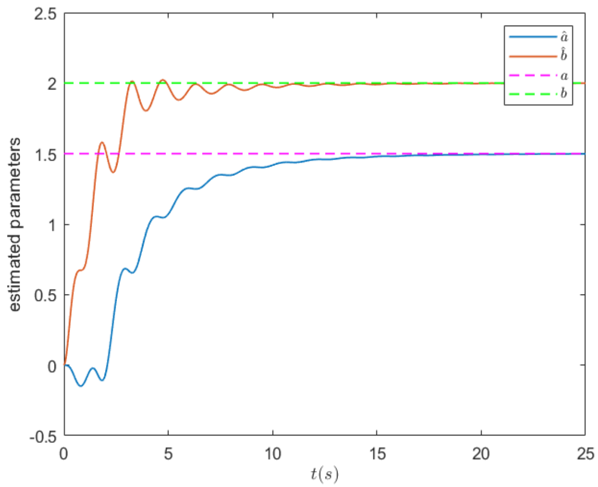

Figure 3.1: Estimated parameters

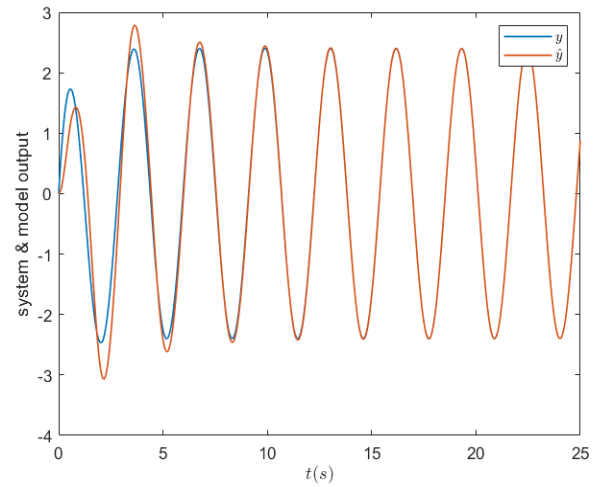

Figure 3.2: System & model output

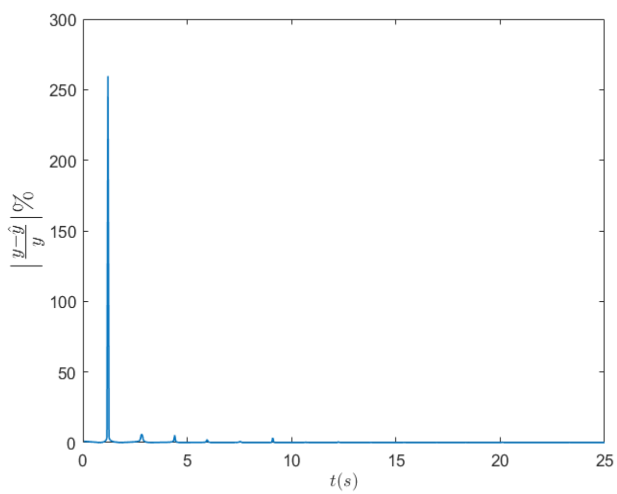

Figure 3.3: Percentage error

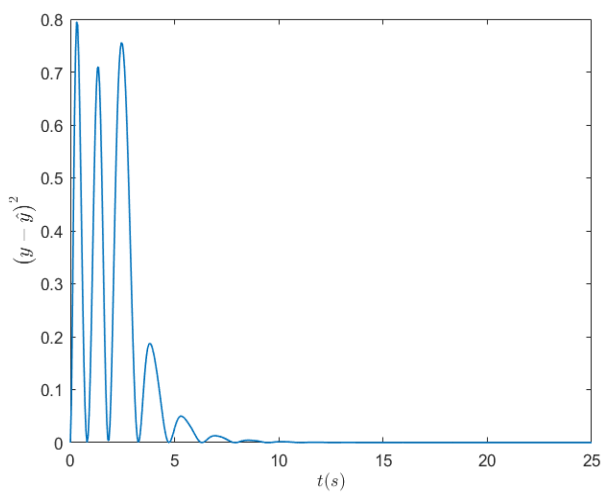

Figure 3.4: Mean square error

3.3 Noise Sensitivity

Although both techniques for modeling seem to have similar degree of accuracy and overall behavior in an ideal environment, the performance under realistic conditions, such as noise, needs to be taken into consideration. For this purpose, we add noise which is described by

$$n(t) = n_0 \sin(2\pi f t)$$

3.3.1 $\mathbf{n}_0 = 0.15, f = 20Hz$

At first we assign the initial values of 0.15 and $20Hz$ to the noise coefficient and the frequency respectively.

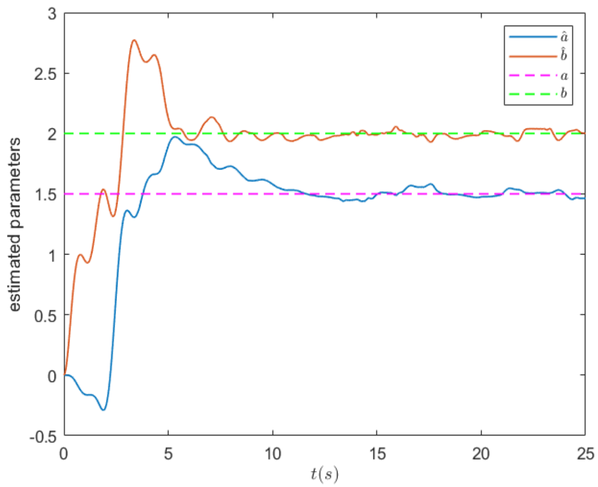

Figure 4.1.1: Estimated parameters with noise (P)

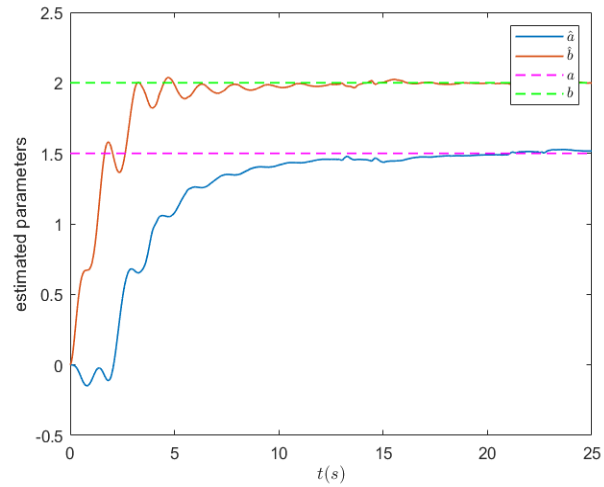

Figure 4.1.2: Estimated parameters with noise (SP)

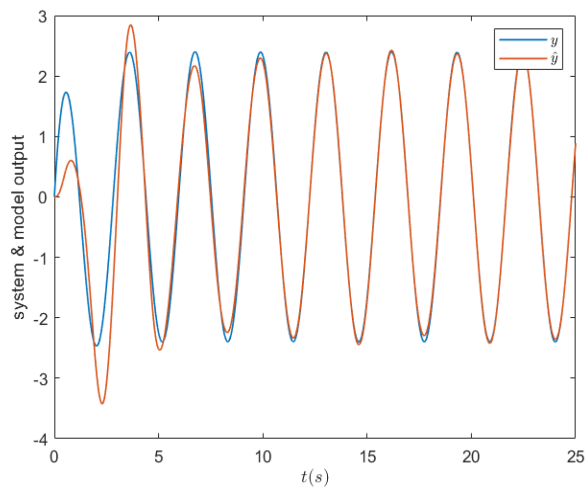

Figure 4.1.3: Output with noise (P)

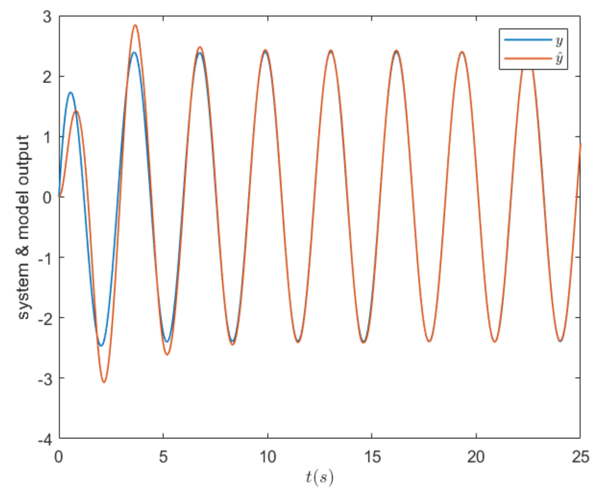

Figure 4.1.4: Output with noise (SP)

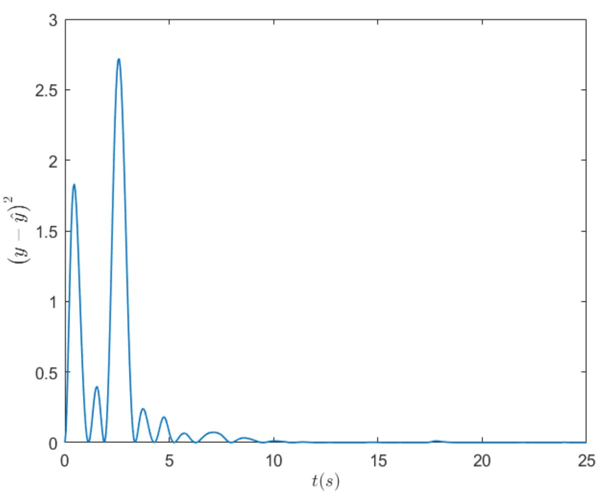

Figure 4.1.5: Mean square error with noise (P)

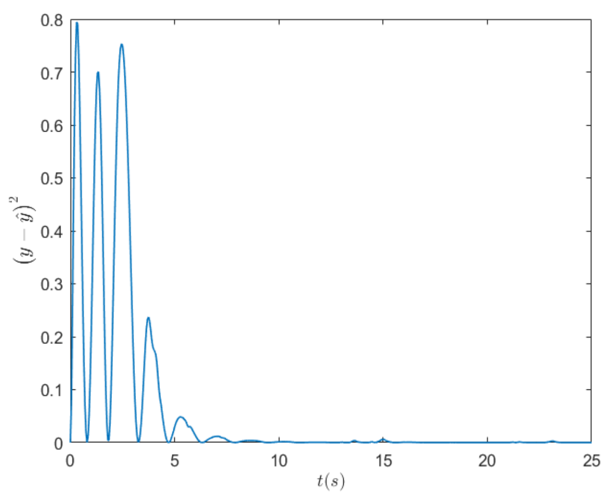

Figure 4.1.6: Mean square error with noise (SP)

3.3.2 $\mathbf{n}_0 = 15, f = 20Hz$

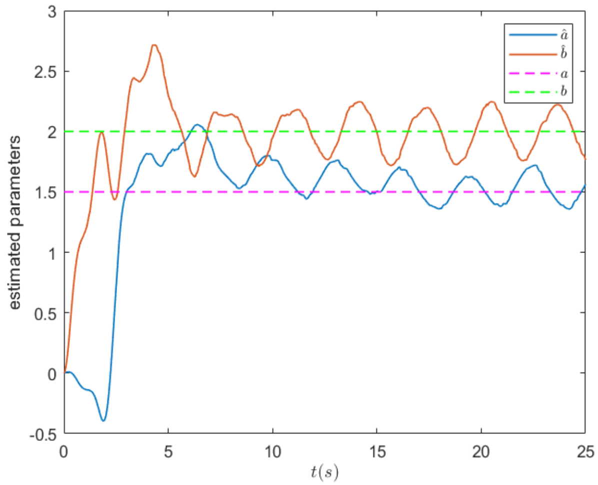

Figure 4.2.1: Estimated parameters with noise (P)

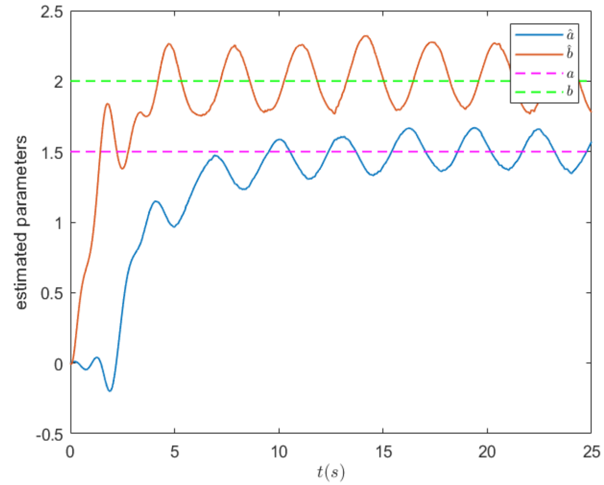

Figure 4.2.2: Estimated parameters with noise (SP)

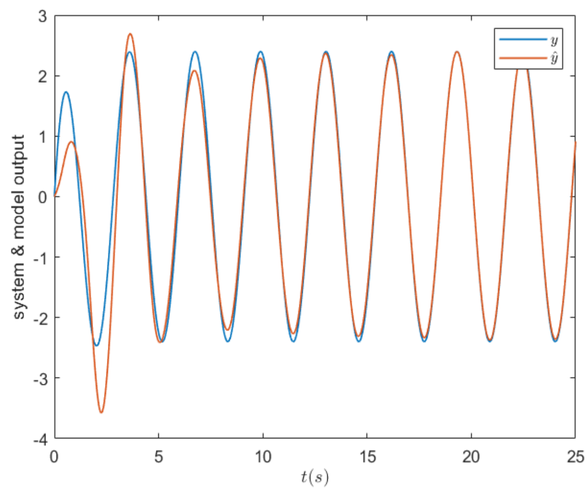

Figure 4.2.3: Output with noise (P)

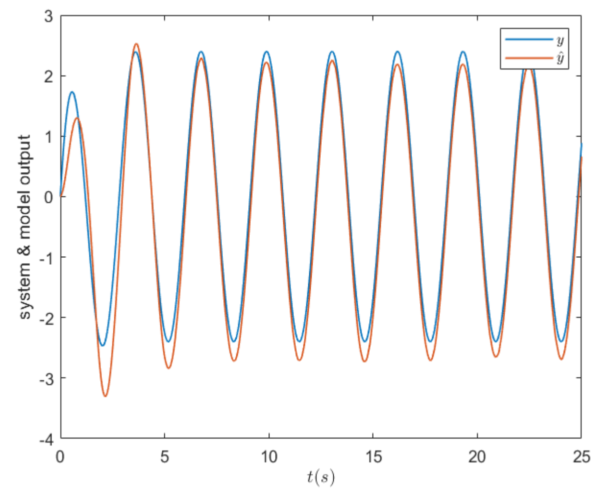

Figure 4.2.4: Output with noise (SP)

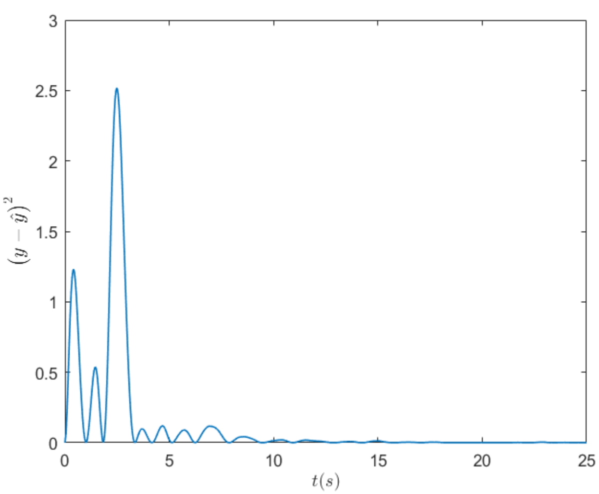

Figure 4.2.5: Mean square error with noise (P)

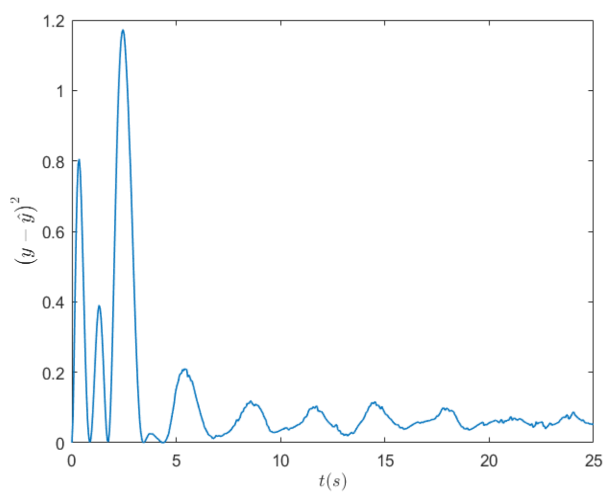

Figure 4.2.6: Mean square error with noise (SP)

3.3.3 $\mathbf{n}_0 = 15, f = 2000Hz$

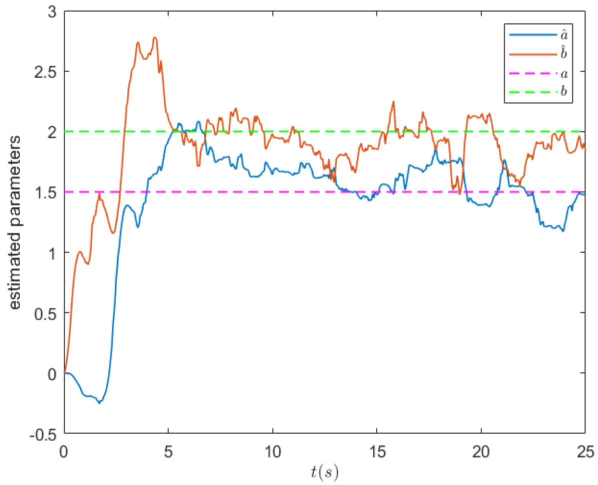

Figure 4.3.1: Estimated parameters with noise (P)

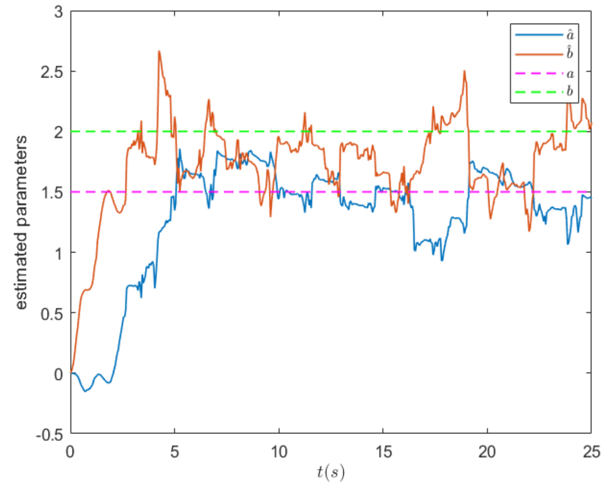

Figure 4.3.2: Estimated parameters with noise (SP)

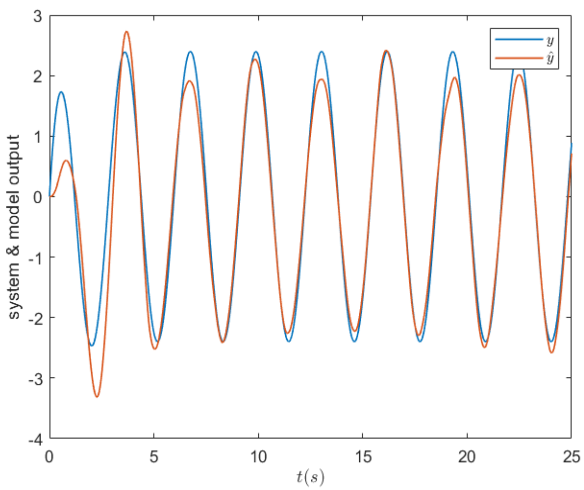

Figure 4.3.3: Output with noise (P)

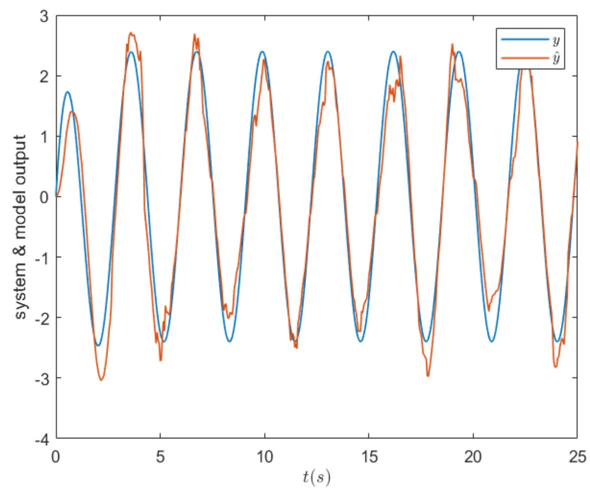

Figure 4.3.4: Output with noise (SP)

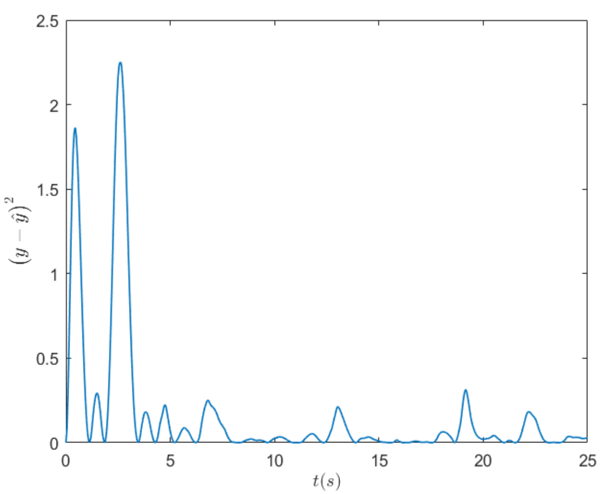

Figure 4.3.5: Mean square error with noise (P)

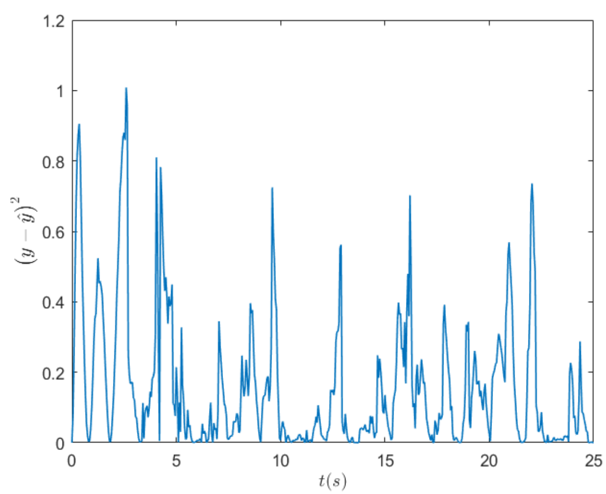

Figure 4.3.6: Mean square error with noise (SP)

3.3.4 Conclusion

What we notice is that, for weak noise, the two models behave similarly, still managing to approach the real values. However, when we increase the coefficient n_0, the **Parallel Model** continues to successfully stabilize the output, retaining a minimal mean square error, while the **Series-Parallel Model** doesn't manage to decrease it, therefore functions with significantly less precision. If we increase the frequency f, the outcome is the same. The **Parallel Model** once again handles the error quite effectively, entrapping it in a small range of values, while the **Series-Parallel Model** fails to diminish it and ends up oscillating in a wide numerical range. This outcome is nevertheless consistent with our expectations, since in **Parallel** configuration, the noise is hidden inside the model but in power of one. On the contrast, in **Series-Parallel** modeling, the noise appears in power of two, making the latter, more sensitive to noise.

3.4 Two Dimensional System

The new system is:

$$\dot{x} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} x + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u, \quad x(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

or

$$\dot{x} = Ax + Bu$$

This time, the **Parallel Model** is

$$\dot{x} = \begin{bmatrix} \hat{a_{11}} & \hat{a_{12}} \\ \hat{a_{21}} & \hat{a_{22}} \end{bmatrix} \hat{x} + \begin{bmatrix} \hat{b_1} \\ \hat{b_2} \end{bmatrix} u$$

thus

$$\dot{x} = \hat{A}\hat{x} + \hat{B}u$$

Following the previous mathematical process, now applied to matrices, we obtain:

$$\dot{e} = Ae - \tilde{A}\hat{x} - \tilde{B}u$$

with

$$e = x - \hat{x}$$

$$\tilde{A} = \hat{A} - A$$
$$\tilde{B} = \hat{B} - B$$

Our **Lyapunov Function** is similarly chosen as

$$V(\hat{\theta}) = \frac{1}{2}e^T e + \frac{1}{2\gamma_1}\tilde{A}^T\tilde{A} + \frac{1}{2\gamma_2}\tilde{B}^T\tilde{B} \iff$$

$$\dot{V}(\hat{\theta}) = e^T Ae - \tilde{A}^T e\hat{x}^T - \hat{B}^T eu^T + \frac{1}{\gamma_1}\tilde{A}^T\dot{\hat{A}} + \frac{1}{\gamma_2}\tilde{B}^T\dot{\hat{B}}$$

So in order to achieve

$$\dot{V}(\hat{\theta}) \leq 0$$

we choose

$$\dot{\hat{A}} = \gamma_1 e\hat{x}^T$$
$$\dot{\hat{B}} = \gamma_2 eu$$

3.4.1 Simulation

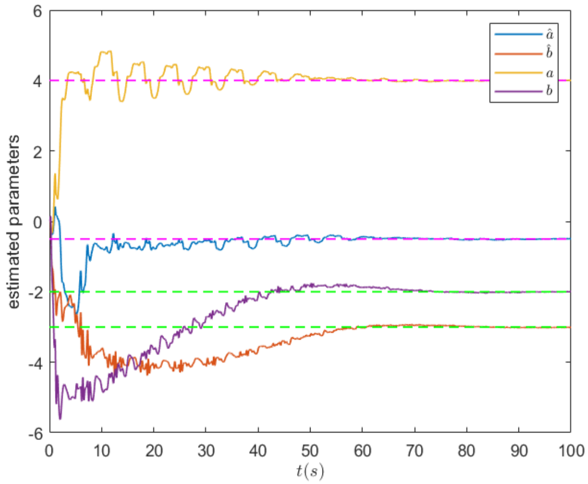

Figure 5.1: Estimated matrix $\hat{A}$ with noise (P)

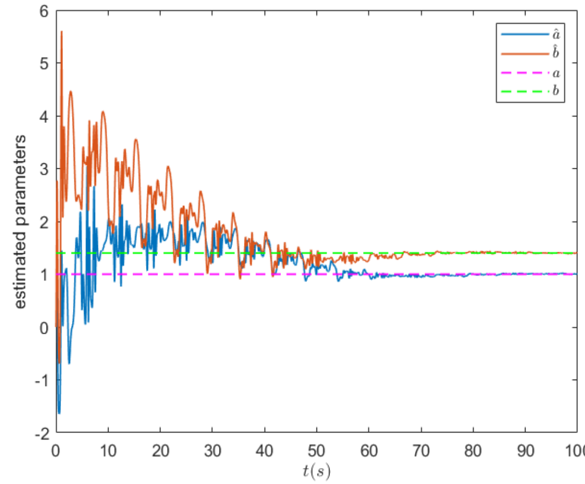

Figure 5.2: Estimated matrix $\hat{B}$ with noise (P)

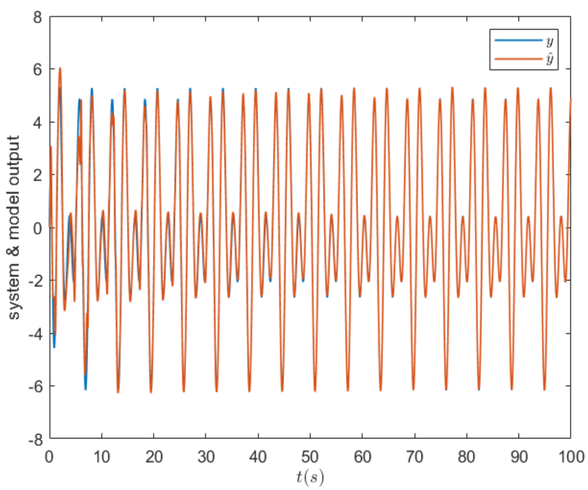

Figure 5.3: Output with noise (P)

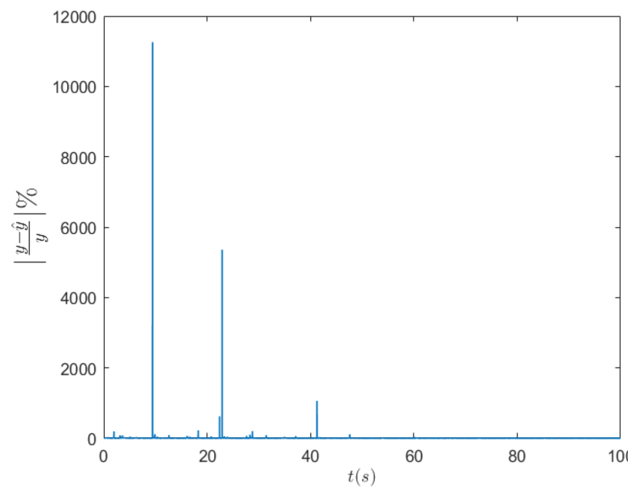

Figure 5.4: Percentage error with noise (P)

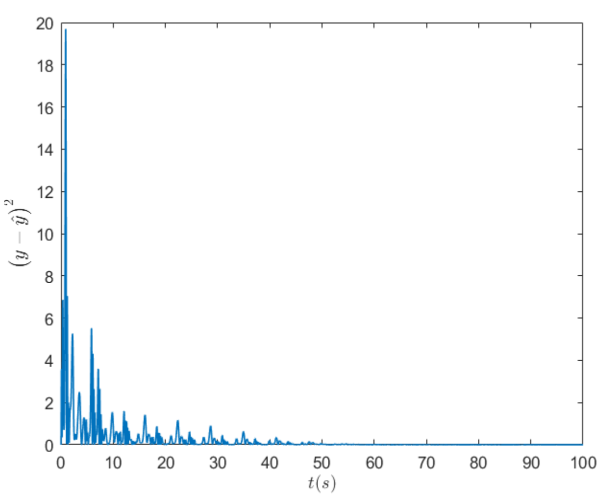

Figure 5.5: Mean square error with noise (P)