Simulation & Modelling of Dynamical Systems

Online Estimation of Dynamical System Parameters

Nikolaos Konstas

Department of Electrical & Computer Engineering
Aristotle University of Thessaloniki

April 2022

Contents

1 Problem 1 - Slope method (Steepest Descend 2

1.1 Theoretical Analysis 2

1.1.1 Linear Parameterization 2

1.1.2 Method of Steepest Descent 2

1.2 Implementation & Results 3

1.2.1 Implementation in MATLAB 3

1.2.2 Results& Remarks 3

2 Problem 2 - Lyapunov Method 13

2.1 Theoretical Analysis 13

2.1.1 Parallel Structure 13

2.1.2 Mixed Structure 14

2.2 Implementation & Results 15

2.2.1 Implementation in MATLAB 15

2.2.2 Results & Remarks 15

3 Problem 3 - Higher Order System 28

3.1 Theoretical Analysis 28

3.2 Implementation & Results 29

3.2.1 Implementation in MATLAB 29

3.2.2 Results & Remarks 29

1 Problem 1 - Slope method (Steepest Descend

)

1.1 Theoretical Analysis

In Problem 1 we are given the following system:

$$\dot{x} = -ax + bu; ,$$

with the request to calculate the parameters a & b by the Steepest Descend Method.

1.1.1 Linear Parameterization

First I need to bring the system into a linearly parameterized form. For this I use a trick, adding to both members the quantity $a_m x$, where a_m is an arbitrary, positive constant.

$$\dot{x} = -ax + bu \xrightarrow{+a_m x}$$
$$\dot{x} + a_m x = a_m x - ax + bu \Rightarrow$$
$$\dot{x} + a_m x = (a_m - a)x + bu \overset{\mathcal{L}}{\Longrightarrow}$$
$$sx + a_m x = (a_m - a)x + bu \Rightarrow$$
$$x(s + a_m) = (a_m - a)x + bu \Rightarrow$$
$$x = \frac{1}{s + a_m}\Big[(a_m - a)x + bu\Big] \Rightarrow$$
$$x = \begin{bmatrix}(a_m - a) & b\end{bmatrix} \cdot \begin{bmatrix}\frac{1}{s + a_m}x & \frac{1}{s + a_m}u\end{bmatrix}^T \tag{1}$$

So I have configured the system linearly, with:

$$\theta = \begin{bmatrix}(a_m - a) & b\end{bmatrix}^T \varphi = \begin{bmatrix}\frac{1}{s + a_m}x & \frac{1}{s + a_m}u\end{bmatrix}^T \tag{2}$$

1.1.2 Method of Steepest Descent

Now about the Steepest Descend Method, I first define the estimate $\hat{x}$ and the error e:

$$\hat{x} = \hat{\theta}^T f e = x - \hat{x},$$

and the error function:

$$K(\hat{\theta}) = \frac{e^2}{2} = \frac{x - \hat{x}}{2} = \frac{x - \hat{\theta}^T\varphi}{2}$$

According to the above for $\dot{\hat{\theta}}$ we will have:

$$\left.\begin{array}{r}\dot{\hat{\theta}} = -\gamma\nabla K \\ \nabla K = -(x - \hat{\theta}^T\varphi)\varphi = -e\varphi\end{array}\right\} \Rightarrow \dot{\hat{\theta}} = \gamma e\varphi \tag{3}$$

,where γ is an arbitrary, positive constant.

To find the values taken by $\hat{\theta_1}$ and $\hat{\theta_2}$:

$$
\begin{aligned}
(3) \Rightarrow \varphi_1 &= \frac{1}{s+a_m}x \xRightarrow{\mathcal{L}^{-1}}{}_1 = -\alpha_m\varphi_1 + x \qquad (\varphi_1(0) = 0) \\
(3) \Rightarrow \varphi_2 &= \frac{1}{s+a_m}u \xRightarrow{\mathcal{L}^{-1}}{}_2 = -\alpha_m\varphi_2 + u \qquad (\varphi_2(0) = 0)
\end{aligned}
$$

$$(4) \Rightarrow \dot{\hat{\theta}}_1 = \gamma e \varphi_1$$

$$(4) \Rightarrow \dot{\hat{\theta}}_2 = \gamma e \varphi_2\ .$$

So, to calculate the values of $\hat{\theta}_1$ and $\hat{\theta}_2$ I will solve the above system of 4 differential equations.

1.2 Implementation & Results

1.2.1 Implementation in MATLAB

As the MATLAB files are provided in the deliverables folder, the steps I followed for the implementation are briefly mentioned:

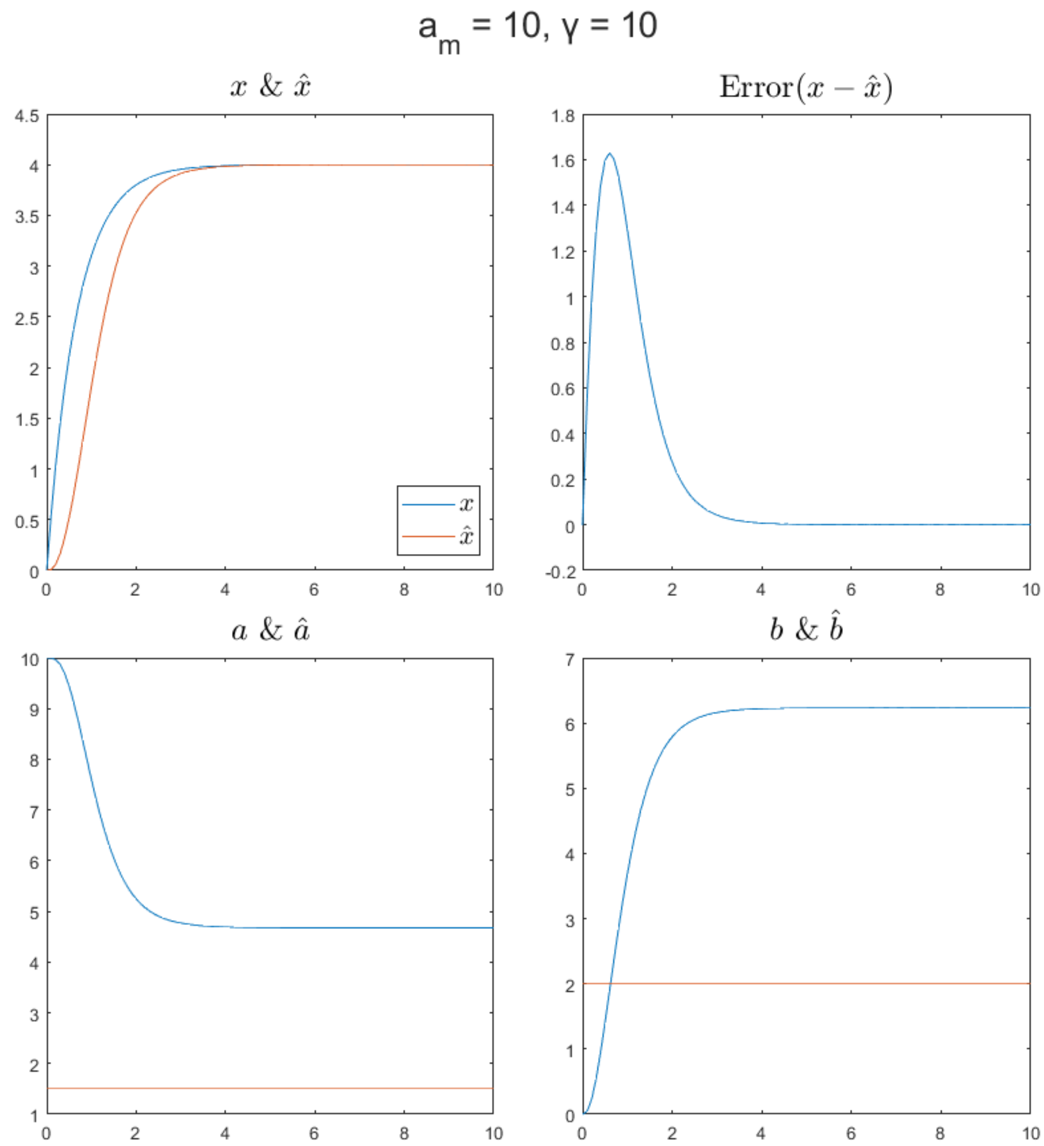
- Set the real values a, b and initialize c, a_m.
- Initialize the vector t (time), the input u.
- Calculation of x by solving the d.e. of the system.
- Finding $\hat{\theta}_1$ and $\hat{\theta}_2$ by the Steepest Descend Method , solving the system of equations presented above using ode45.
- Print the necessary graphs.
- Iterative loop to find the optimal values c, a_m.

1.2.2 Results& Remarks

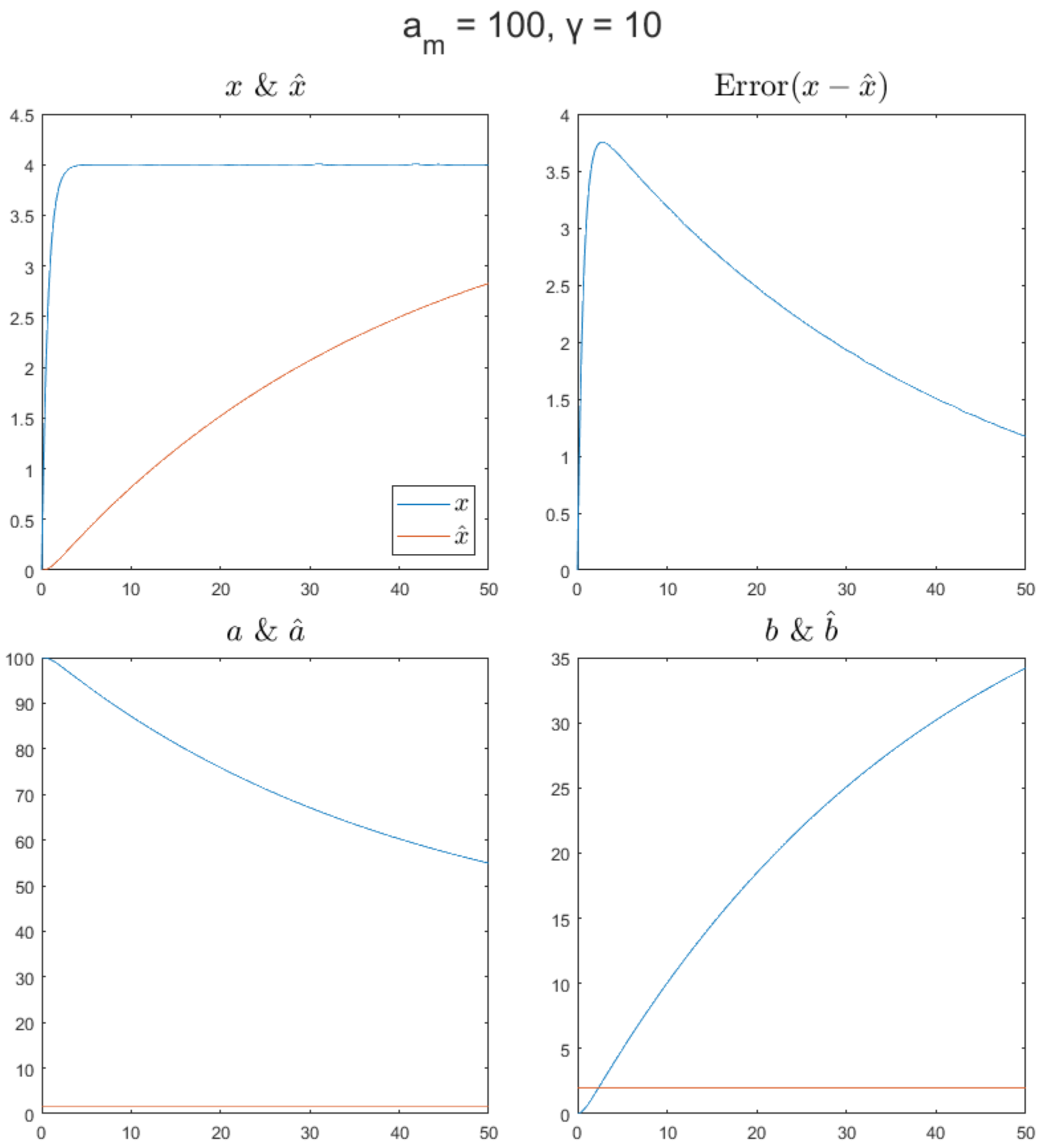
Example a)

In the a) task of Theorem 1, I need to estimate the values of the parameters by applying input $u = 3$.
To estimate the parameters of the system, I have 2 hyper-parameters (c, a_m) which I can vary so as to get better approximations.

So I start testing to get a baseline estimate, with $\gamma = 10$ and $a_m = 10$. The next page shows the graphs of the estimates and the error.
I observe that $\hat{x}$ follows x with a delay, with the error e reaching its Steepest value for about $t = 1s$, while in the steady state where the output x is stabilized at 4 the error is zero. The $\hat{a}$ and $\hat{b}$ although converging to some value are quite far from the real parameters of the system. This result may be reasonable, as the fixed input $u = 3$ probably does not excite the system appropriately.

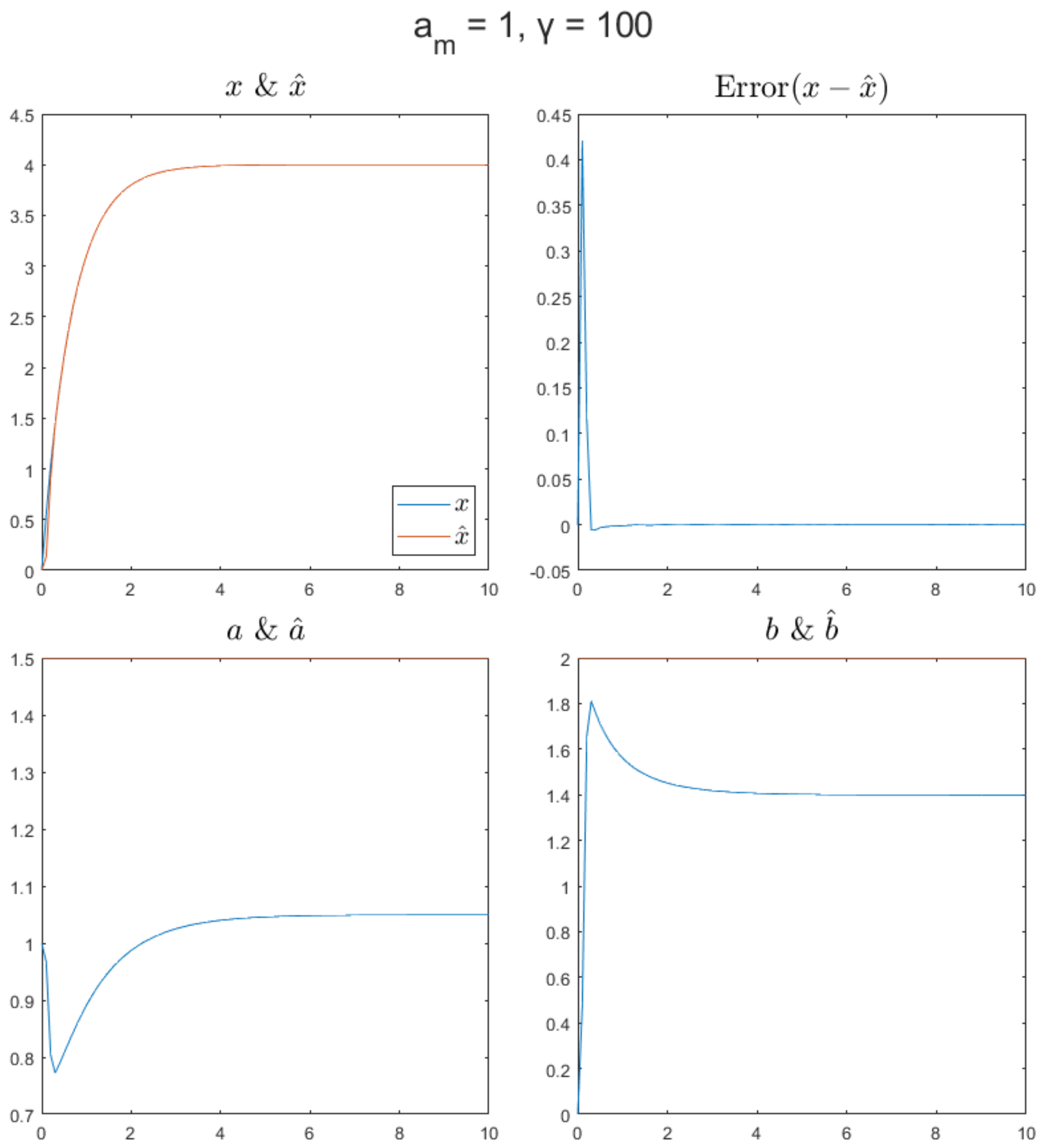


To draw further conclusions and find better estimates of the input and system parameters we try new values of c, a_m. For the moment I keep c fixed and test $a_m = 1$ & $a_m = 100$. For $a_m = 1$ the error is smaller with above and $\hat{x}$ matches x earlier. As for $\hat{a}$ and $\hat{b}$, they exhibit similar behavior as before, converging to values with large deviation from the true ones.
Of great interest are the results for $a_m = 100$, which are presented below. In particular, the error is greatly increased with $_{h}atx$ not satisfactorily approaching x and $_{h}ata$ and $_{h}atb$ being at even more distant values, not even converging this time. Considering that the limited time of 10 seconds of the experiment may affect the results I increase it to 50, but the results are similar. The graphs are for $tspan = [0, 50]$.

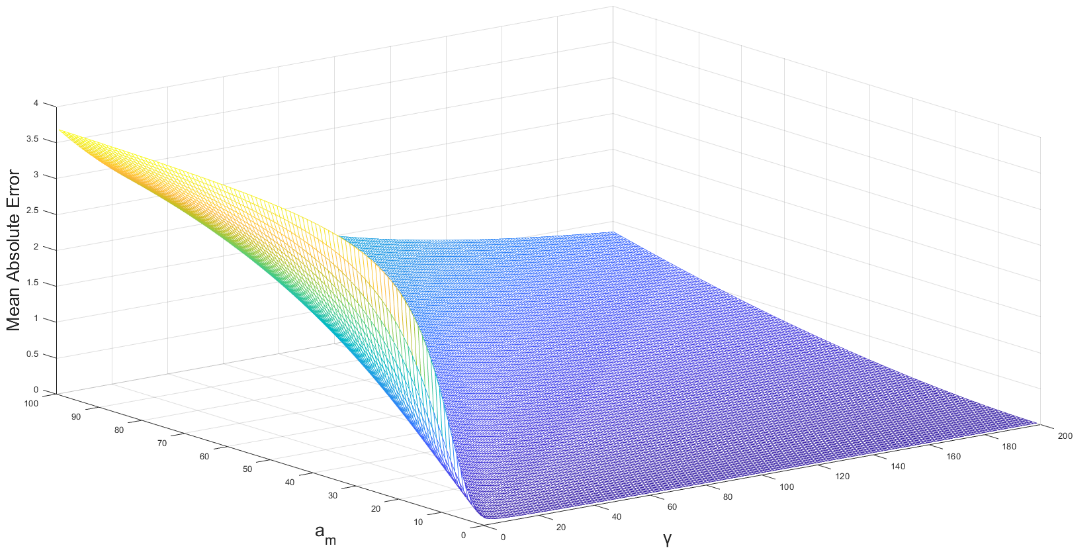


As we saw above the small value a_m led to better results, so I keep $a_m = 1$ and try to give c the values 1 & 100.

The results are similar to before, but for $\gamma = 100$ I have a noticeably better approximation. In particular, the error is at each time instant less than 0.5 and it quickly resets to zero. Also, the values $\hat{a}$ and $\hat{b}$ are closer to the true values, but still have a noticeable deviation, as we see below.

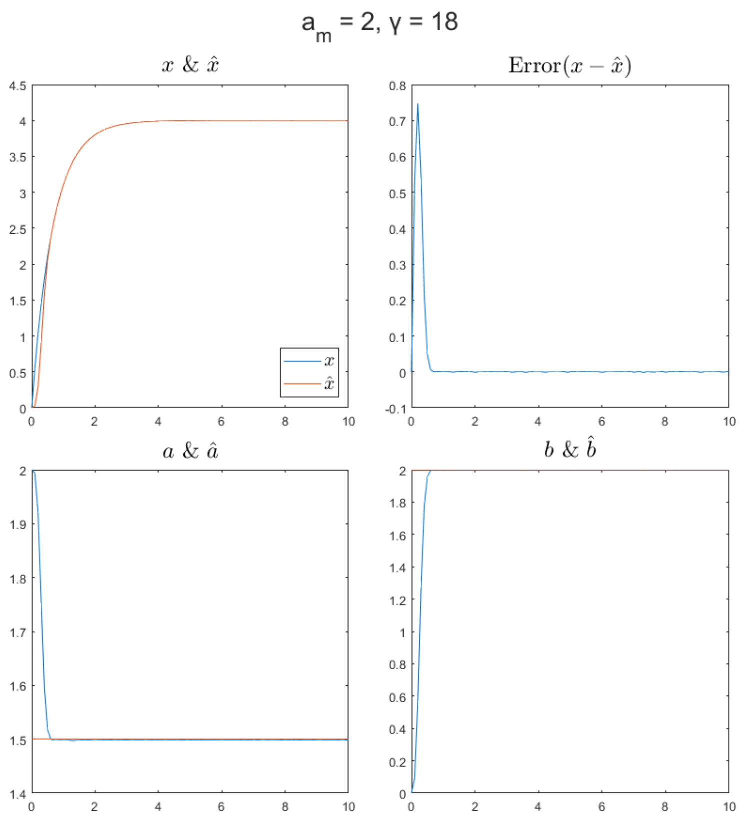


At this point I use the regression method I have implemented in order to verify my assumptions about the effect of c, a_m. At this point I consider values: $\gamma \in [1, 100]$ and $a_m \in [1, 200]$ with $step = 1$.
Indeed, as already observed, the best value for a_m is 1, while for c, there seems to be no critical point and the higher the value, the smaller the estimation error. This, mainly, is due to the speed with which $_hatx$ approximates the actual signal during the transient phenomenon, since in the steady state we have practically zero error regardless of γ. The mean absolute error of the approximation is shown below as a function of a_m and γ.



By doing an additional test for values of a_m up to 1000, I confirm my suspicion that the error continues to decrease for any higher value of a_m.

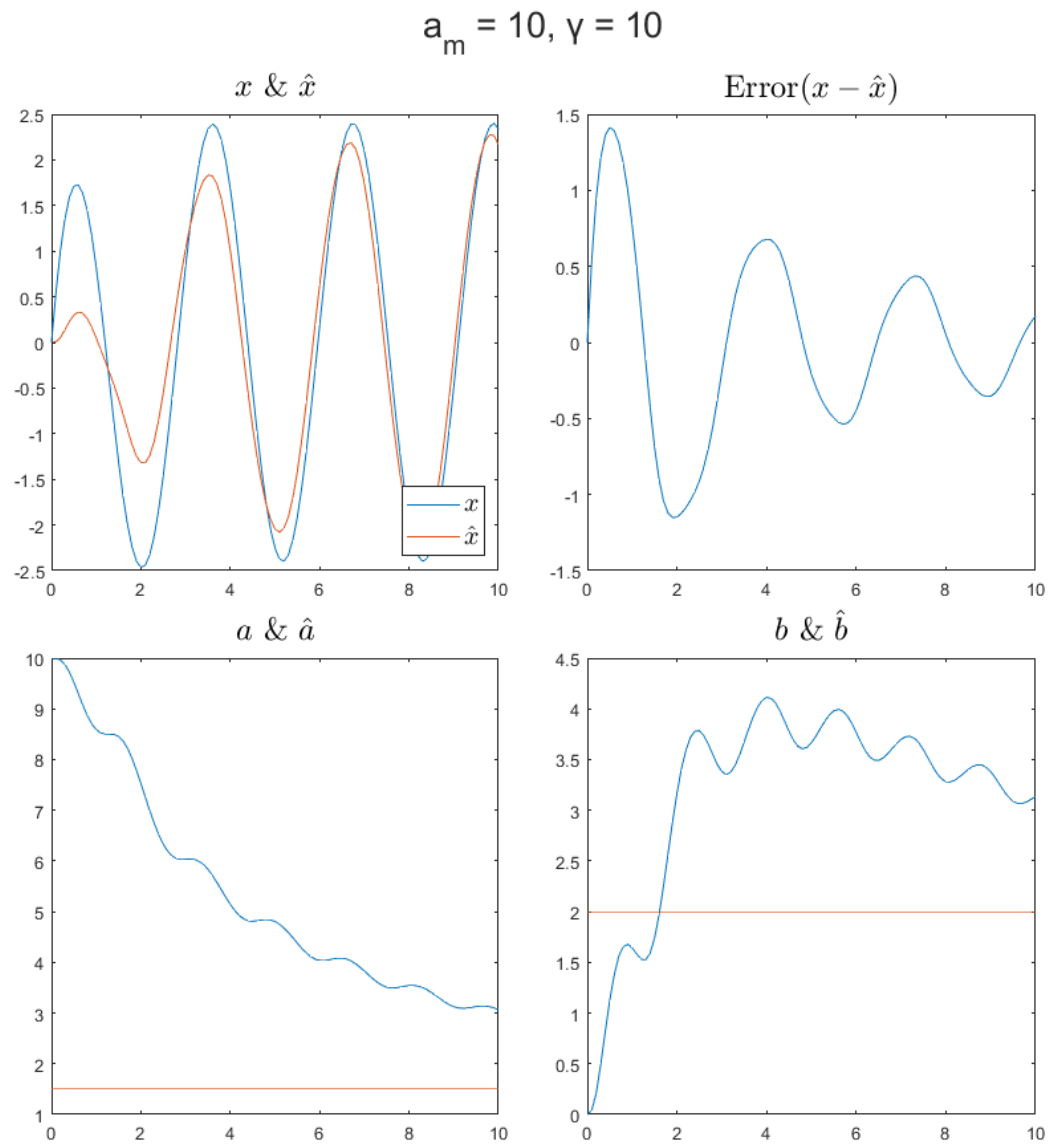
Now I do an unorthodox, and not at all realistic, test, looking for the best approximation of a and b, which in a real project would not be possible, as we would not know their values.



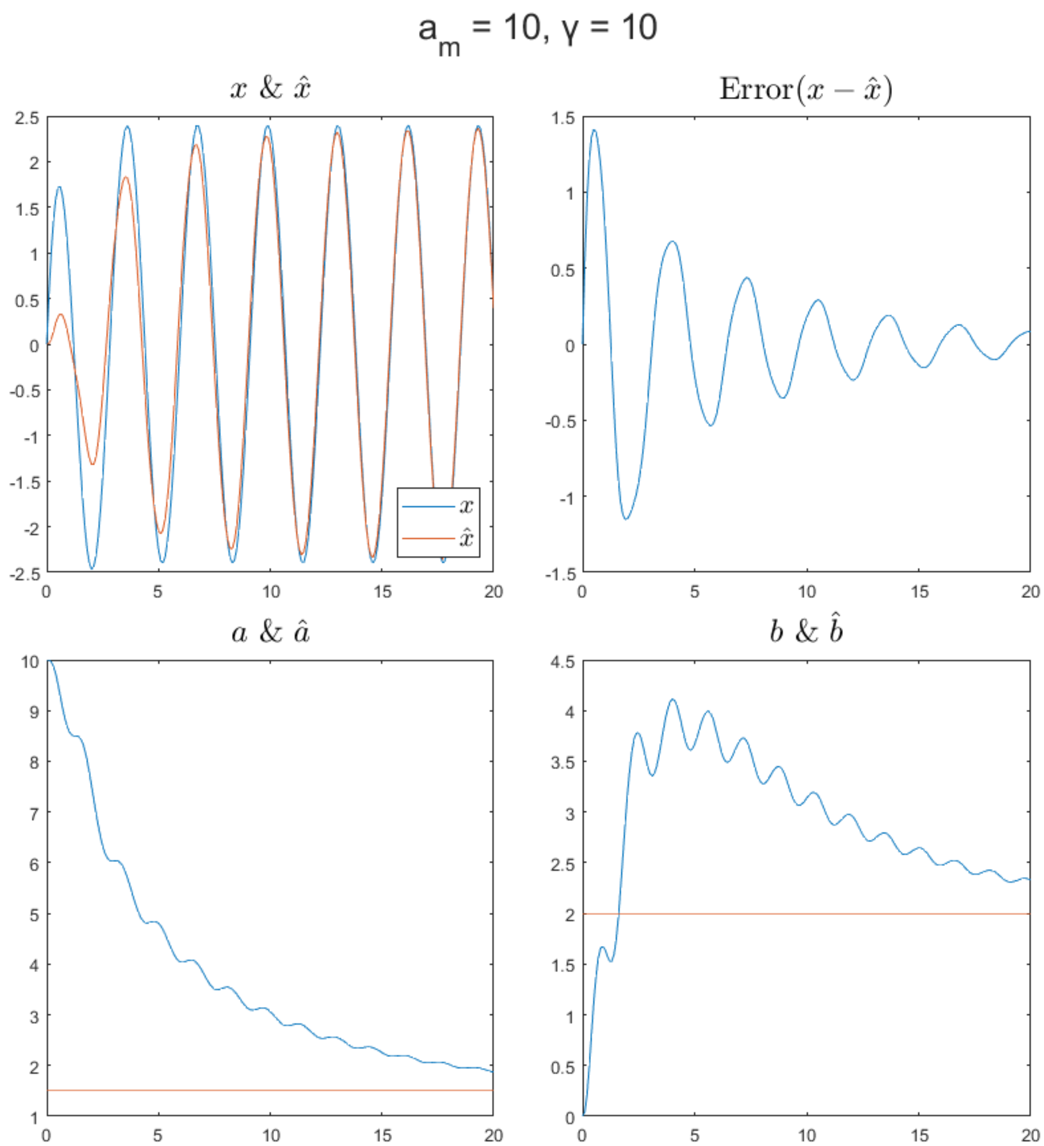
In this case I observe similar behavior from a_m with the best approximation occurring for $a_m = 2$, while the best value of c is 18, i.e. there is not the continuous improvement observed earlier. The approximations $_hata$ and $_hatb$ are very good with practically zero error after convergence. Moreover, the output error is very small, with a small divergence in the early moments which disappears in the steady state.
The last results lead us to the conclusion that smaller error does not imply better parameter estimation, and also reinforce my initial suspicion that the input is not suitable for parameter estimation.

Example b)

So I proceed to question b) where I have input $u = 3cos(2t)$. Using the same logic as earlier I start my tests with $c = 10$ and $a_m = 10$.

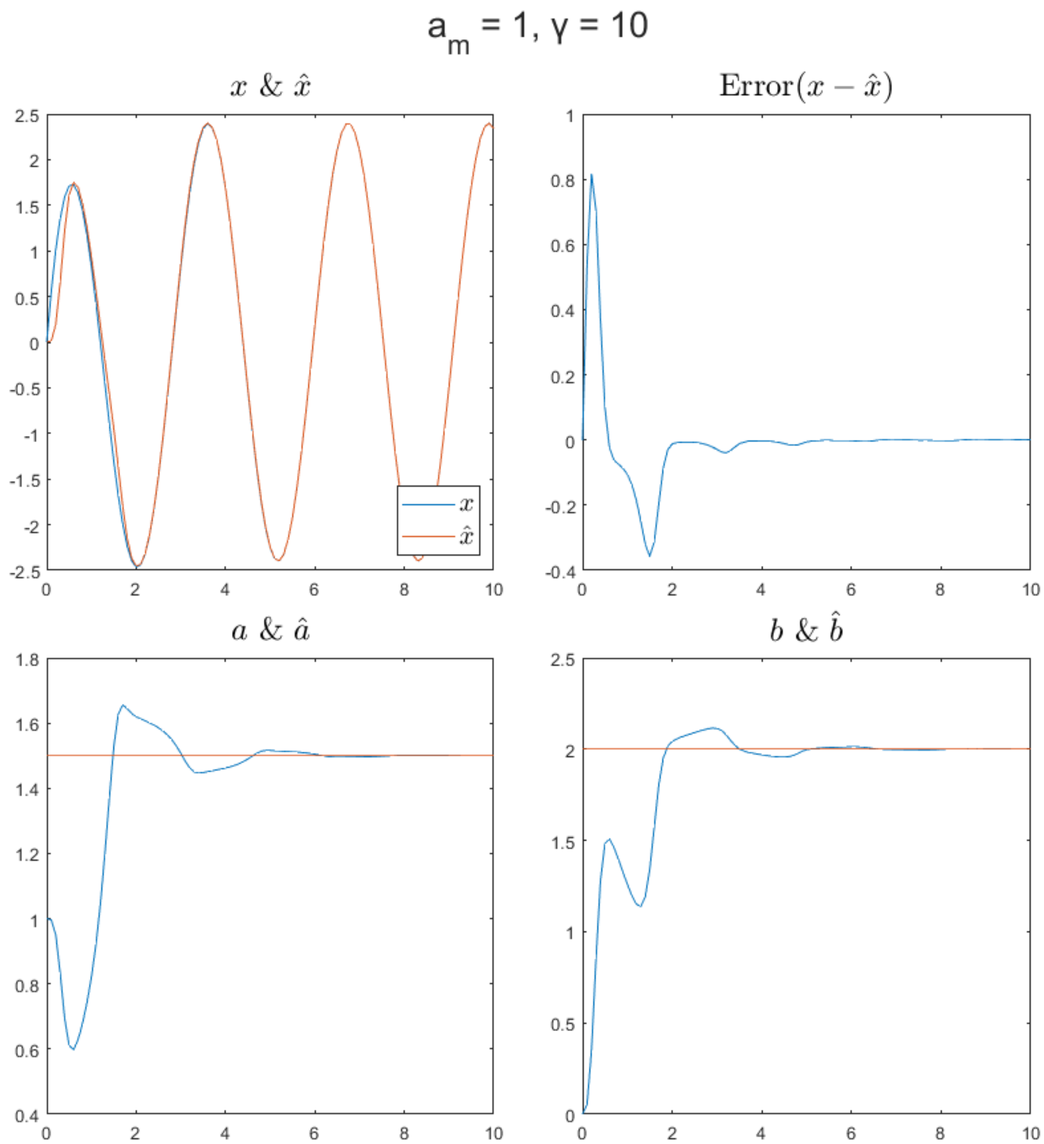


In this test I observe quite a large error with $\hat{x}$ not having accurately approximated the output signal until the end of the time interval, while $\hat{a}$ and $\hat{b}$ seem to be slowly converging towards some value, but the time duration is not sufficiently long for them either. So, I do the same experiment again with $tspan = [0, 20]$ this time.



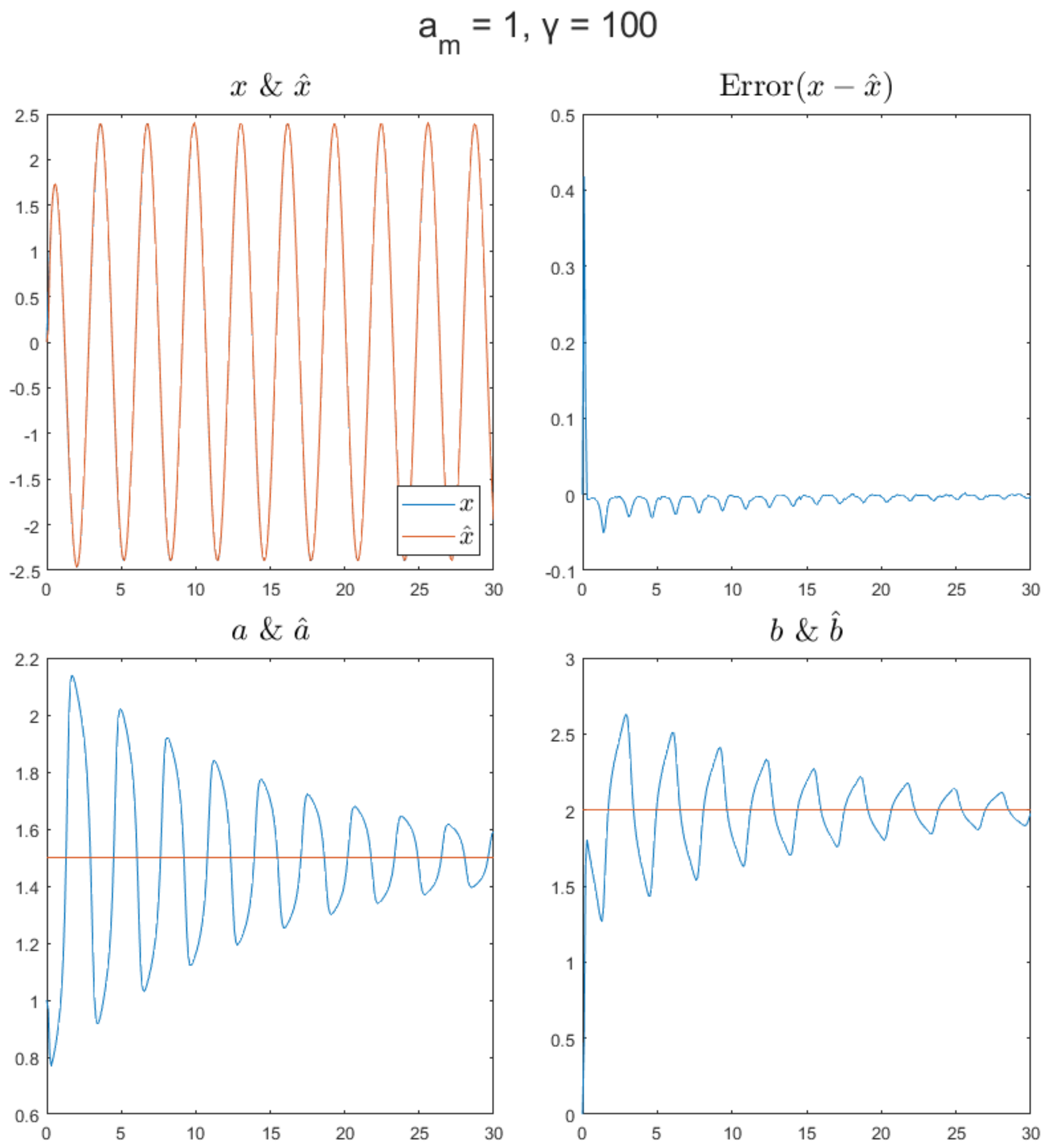
Indeed, by increasing the duration of the test, there is an improvement in the approximation over time, but the convergence is quite slow, so I try to change the hyperparameters by first changing a_m from 10 to 1& 100.
The results for $a_m = 100$ are tragic with the error being large throughout the test and the system parameters not converging, so the corresponding graphs are not shown.

In contrast, by giving a_m a value of 1, I observe a significant reduction in the error, which occurs for the first 2 seconds or so, and is zeroed out later. Also, $\hat{a}$ and $\hat{b}$ converge rapidly, and even to the true values of a and b.

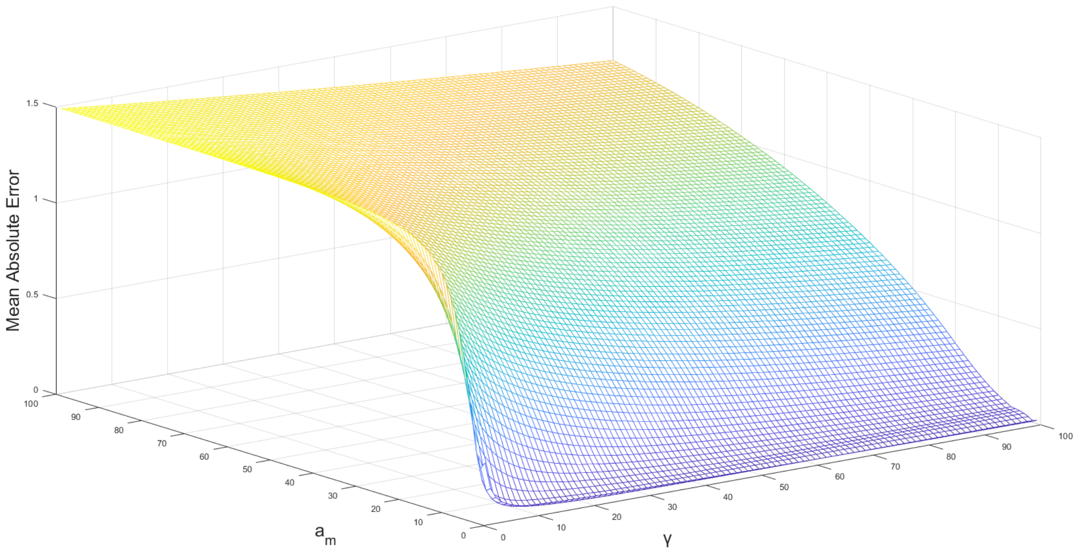


I have already gotten very good approximations for both the output and the system parameters, but I continue my tests, this time changing the value of c to 1 & 100, while for a_m I keep the value 1 which gave very good results.

For $\gamma = 1$, similar results are observed with a larger error, and thus are omitted. Of great interest are the results for $c = 100$. As shown below in the graphs, the error takes a Steepest value of about 0.4 and within a few tenths of a second it gets very close to 0. As for $\hat{a}$ and $\hat{b}$, I notice that they perform a damped oscillation around the real values 1.5 & 2 respectively. The results shown are for $tspan = [0, 30]$, as the behavior of $\hat{a}$ and $\hat{b}$ is best seen.



Having formed a picture of how a_m and c affect the estimation, I explore further, looking at all pairs of values in the intervals [1,100]& [1,100].
Again we observe similar behavior to the first question, but with the value of a_m playing a larger role this time and for high values we do not perform as well. As for c, by increasing its value I observe a small and steady decrease in the mean absolute error. The optimal model in this test was the one with $a_m = 2$ and $c = 100$, and by doing another test where c took values up to 1000, it was observed that the error continued to decrease.



In conclusion, it should be noted that while for large values of c our approximation is more accurate it significantly delays the convergence of the parameters $\hat{a}$ and $\hat{b}$. Thus, depending on the application, a small value of γ may be preferred for faster convergence and estimation of the system parameters, or, since it is important to minimize the error throughout the measurements, we may prefer some larger value for γ. The only certainty is that a_m, for this particular system, should take small values.

2 Problem 2 - Lyapunov Method

2.1 Theoretical Analysis

Let the system be the same as the 1st Theme:

$$\dot{x} = -\theta_1^* x + \theta_2^* u \ , \ x(0) = 0$$

, where $\theta_1^* = a$ and $\theta_2^* = b$.

2.1.1 Parallel Structure

First I define the parallel structure recognition topology:

$$\dot{\hat{x}} = -\hat{\theta}_1 \hat{x} + \hat{\theta}_2 u \ , \ \hat{x}(0) = 0 \tag{1}$$

Now I set the error:

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

and I produce in terms of time:

$$\begin{aligned}
\dot{e} &= \dot{x} - \dot{\hat{x}} \Rightarrow \\
\dot{e} &= -\theta_1^* x + \theta_2^* u + \hat{\theta}_1 \hat{x} - \hat{\theta}_2 u \xrightarrow{\pm \theta_1^* \hat{x}} \\
\dot{e} &= -\theta_1^* x + \theta_1^* \hat{x} - \theta_1^* \hat{x} + \hat{\theta}_1 \hat{x} + \theta_2^* u - \hat{\theta}_2 u \Rightarrow \\
\dot{e} &= -\theta_1^* (x - \hat{x}) + (\hat{\theta}_1 - \theta_1^*)\hat{x} + (\theta_2^* - \hat{\theta}_2) u \Rightarrow \\
\dot{e} &= -\theta_1^* e + \tilde{\theta}_1 \hat{x} - \tilde{\theta}_2 u
\end{aligned} \tag{2}$$

Where, $\tilde{\theta}_1 = \hat{\theta}_1 - \theta_1^*$ and $\tilde{\theta}_2 = \hat{\theta}_2 - \theta_2^*$ are the parametric errors of the estimates. Then I define the Lyapunov function:

$$V = \frac{1}{2}e^2 + \frac{1}{2}\tilde{\theta}_1^{\ 2} + \frac{1}{2}\tilde{\theta}_2^{\ 2} \tag{3}$$

Obviously $V > 0 \ \forall e, \tilde{\theta}_1, \tilde{\theta}_2 \in \mathbb{R}$ and $V = 0$ only when $\begin{bmatrix} e & \tilde{\theta}_1 & \tilde{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. The time derivative of V is:

$$\begin{aligned}
\dot{V} &= e\dot{e} + \tilde{\theta}_1 \dot{\tilde{\theta}}_1 + \tilde{\theta}_2 \dot{\tilde{\theta}}_2 \xrightarrow{(2)} \\
\dot{V} &= -\theta_1^* e^2 + e\tilde{\theta}_1 \hat{x} - e\tilde{\theta}_2 u + \tilde{\theta}_1 \dot{\hat{\theta}}_1 + \tilde{\theta}_2 \dot{\hat{\theta}}_2
\end{aligned}$$

, since $\dot{\tilde{\theta}}_1 = \dot{\hat{\theta}}_1$ and $\dot{\tilde{\theta}}_2 = \dot{\hat{\theta}}_2$.

Choose:

$$\begin{aligned}
\dot{\hat{\theta}}_1 &= -e\hat{x} \\
\dot{\hat{\theta}}_2 &= eu
\end{aligned}$$

Therefore:

$$\dot{V} = -\theta_1^* e^2 \leq 0 \tag{4}$$

So, $e, \tilde{\theta}_1, \tilde{\theta}_2 \in \mathbf{L}_\infty$. Also, $V \geq 0$ and V decreasing function of time, since $\dot{V} \leq 0$. So, $\lim_{t\to\infty} V = V_\infty$ exists. Completing both terms of (3) from 0 to ∞ I have:

$$\theta_1 \int_0^\infty e^2 \, d\tau = -\int_0^\infty \dot{V} \, d\tau = -[V_\infty - V(0)] \tag{5}$$

Therefore, $e \in \mathbf{L_2}$. Since $e \in \mathbf{L}_\infty$, $x \in \mathbf{L}_\infty$(since $u \in \mathbf{L}_\infty$) and $e = x - \hat{x}$, then $\hat{x} \in \mathbf{L}_\infty$. Also from (3) I understand that $\dot{e} \in \mathbf{L}_\infty$, as a sum of uniformly blocked signals. So by Barbalat's Lemma, since $e \in \mathbf{L_2} \cap \mathbf{L}_\infty$ and $\dot{e} \in \mathbf{L}_\infty$, $\lim_{t\to\infty} e(t) = 0$. Also, it is $\lim_{t\to\infty} \dot{\hat{\theta}}_1 = \lim_{t\to\infty} \dot{\hat{\theta}}_2 = 0$ (zero *cdot* blocked).

Therefore, I understand that with the above options $\dot{\hat{\theta}}_1$ and $\dot{\hat{\theta}}_2$, since the error e is set to zero that the estimate $\hat{x}$ will converge to the output x of the system after a sufficient time interval, while $\lim_{t\to\infty} \dot{\hat{\theta}}_1 = \lim_{t\to\infty} \dot{\hat{\theta}}_2 = 0$ cannot assure us that $\tilde{\theta}_1$ and $\tilde{\theta}_2$ will converge to a fixed value.

2.1.2 Mixed Structure

The second topology we deal with in this paper is that of the mixed structure, where:

$$\dot{\hat{x}} = -\hat{\theta}_1 x + \hat{\theta}_2 u + \theta_m (x - \hat{x}) \ , \ \hat{x}(0) = 0 \tag{6}$$

Similar to before I generate the e error and have:

$$\dot{e} = -\theta_m e + \tilde{\theta}_1 x - \tilde{\theta}_2 u \tag{7}$$

I again define the Lyapunov function:

$$V = \frac{1}{2}e^2 + \frac{1}{2}\tilde{\theta}_1^{\,2} + \frac{1}{2}\tilde{\theta}_2^{\,2} \tag{8}$$

, and I generate it in terms of time:

$$\dot{V} = -\theta_m e^2 + e\tilde{\theta}_1 x - e\tilde{\theta}_2 u + \tilde{\theta}_1 \dot{\hat{\theta}}_1 + \tilde{\theta}_2 \dot{\hat{\theta}}_2 \tag{9}$$

, so this time I choose:

$$\dot{\hat{\theta}}_1 = -ex$$
$$\dot{\hat{\theta}}_2 = eu$$

Therefore:

$$\dot{V} = -\theta_m e^2 \leq 0 \tag{10}$$

As before, it turns out that $\lim_{t\to\infty} e(t) = 0$, $\lim_{t\to\infty} \dot{\hat{\theta}}_1 = \lim_{t\to\infty} \dot{\hat{\theta}}_2 = 0$.

At this point, before proceeding with the implementation, it should be noted that for the sake of simplicity, and as the goal is to observe the behavior of the estimates for different degrees of noise, the scaling parameters γ_1, γ_2 have not been used.

2.2 Implementation & Results

2.2.1 Implementation in MATLAB

As the MATLAB files are provided in the deliverables folder, the steps I followed for the implementation are listed schematically:

- Definition of the actual values a, b and initialization of n_0, f, θ_m.
- Initialize vector t (time), input u.
- Calculation of x by solving the d.e. of the system.
- Finding $\hat{\theta_1}$ and $\hat{\theta_2}$ with the Lyapunov Method, for a parallel & mixed estimation structure.
- Print the necessary graphs.

To estimate the output and parameters of the system, the following systems of differential equations were used, which were solved using *ode45*:

Parallel Structure:

$$\begin{cases} \dot{x} = -\theta_1^* x + \theta_2^* u \\ \dot{\hat{x}} = -\hat{\theta_1}\hat{x} + \hat{\theta_2} u \\ \dot{\hat{\theta_1}} = -e\hat{x} \\ \dot{\hat{\theta_2}} = eu \end{cases} \tag{11}$$

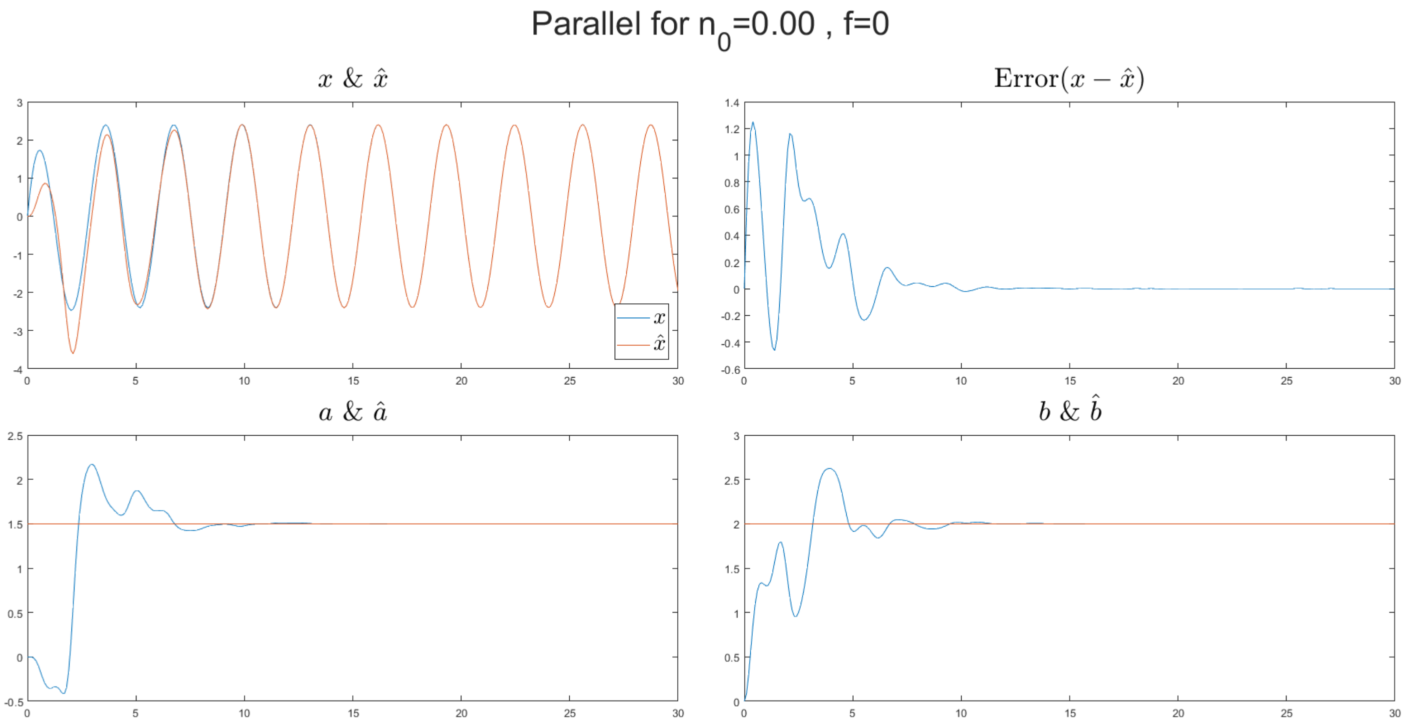
Mixed structure:

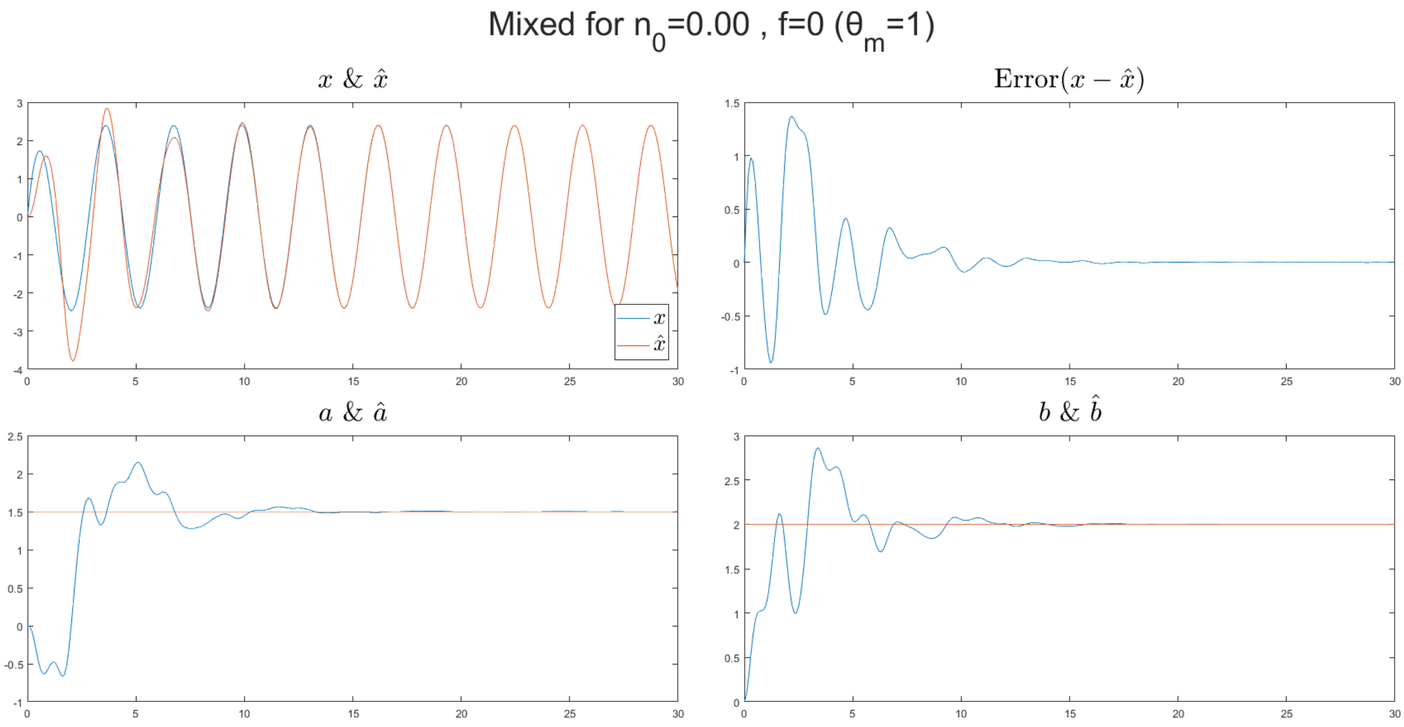
$$\begin{cases} \dot{x} = -\theta_1^* x + \theta_2^* u \\ \dot{\hat{x}} = -\hat{\theta_1} x + \hat{\theta_2} u + \theta_m (x - \hat{x}) \\ \dot{\hat{\theta_1}} = -ex \\ \dot{\hat{\theta_2}} = eu \end{cases} \tag{12}$$

Both topologies used initial conditions $\begin{bmatrix} x(0) & \hat{x}(0) & \dot{\hat{\theta_1}}(0) & \dot{\hat{\theta_2}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$

2.2.2 Results & Remarks

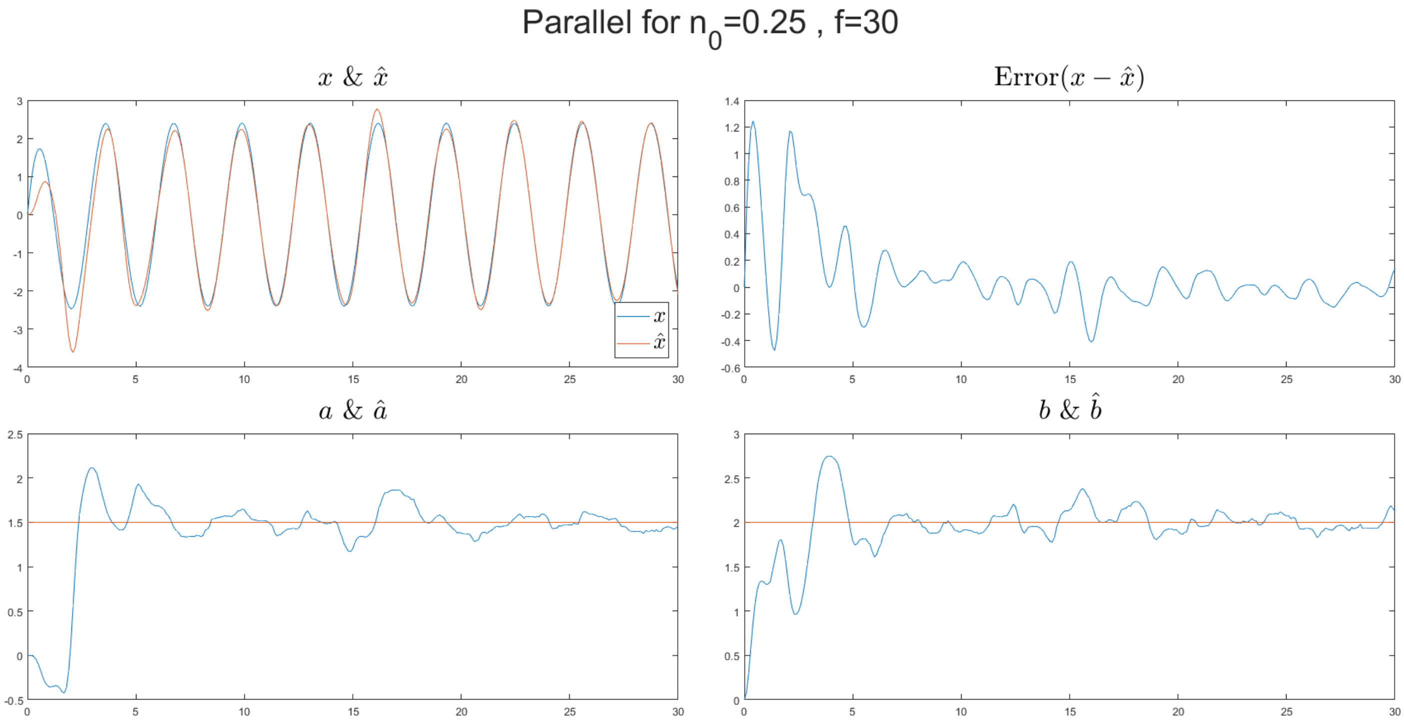
First I do a test without noise.

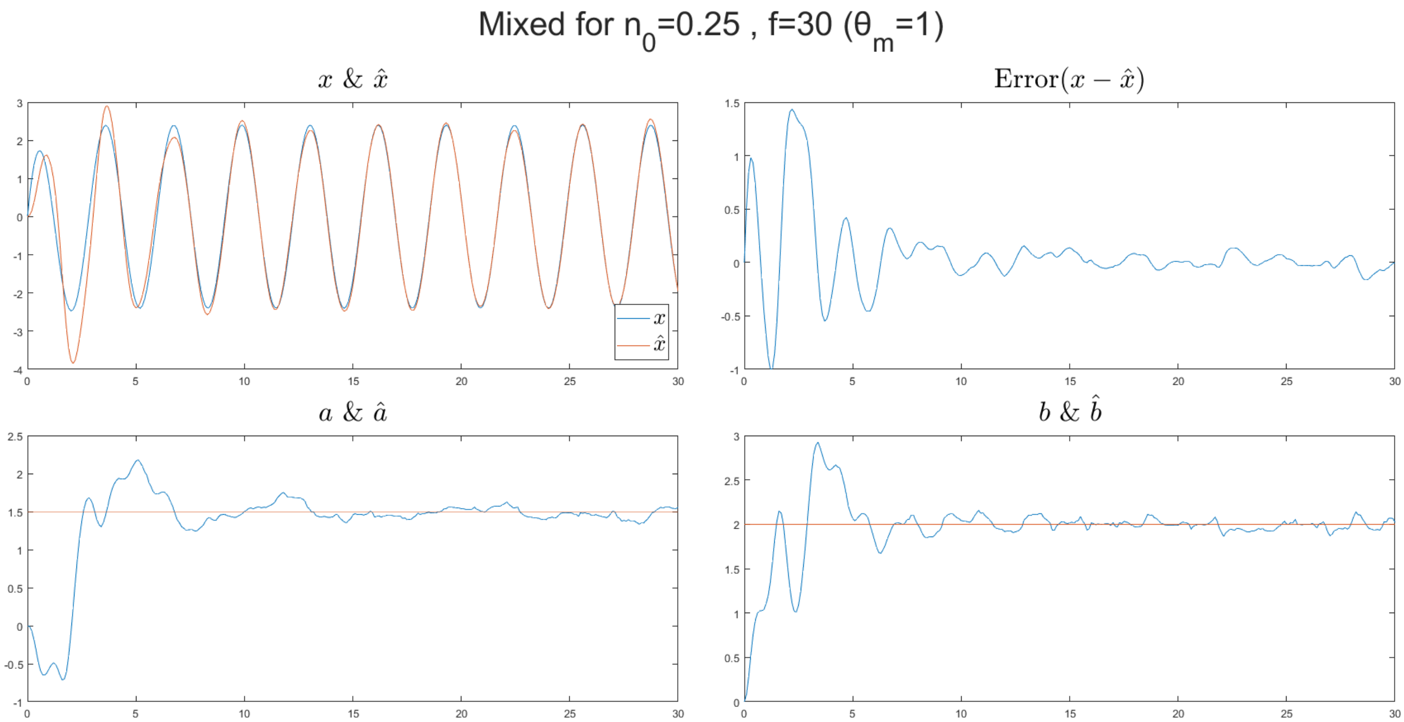




Both topologies show similar behavior with the error taking its maximum value, in the first seconds of the test, and the parameter estimates converging, to the true values, around 10-15 seconds.

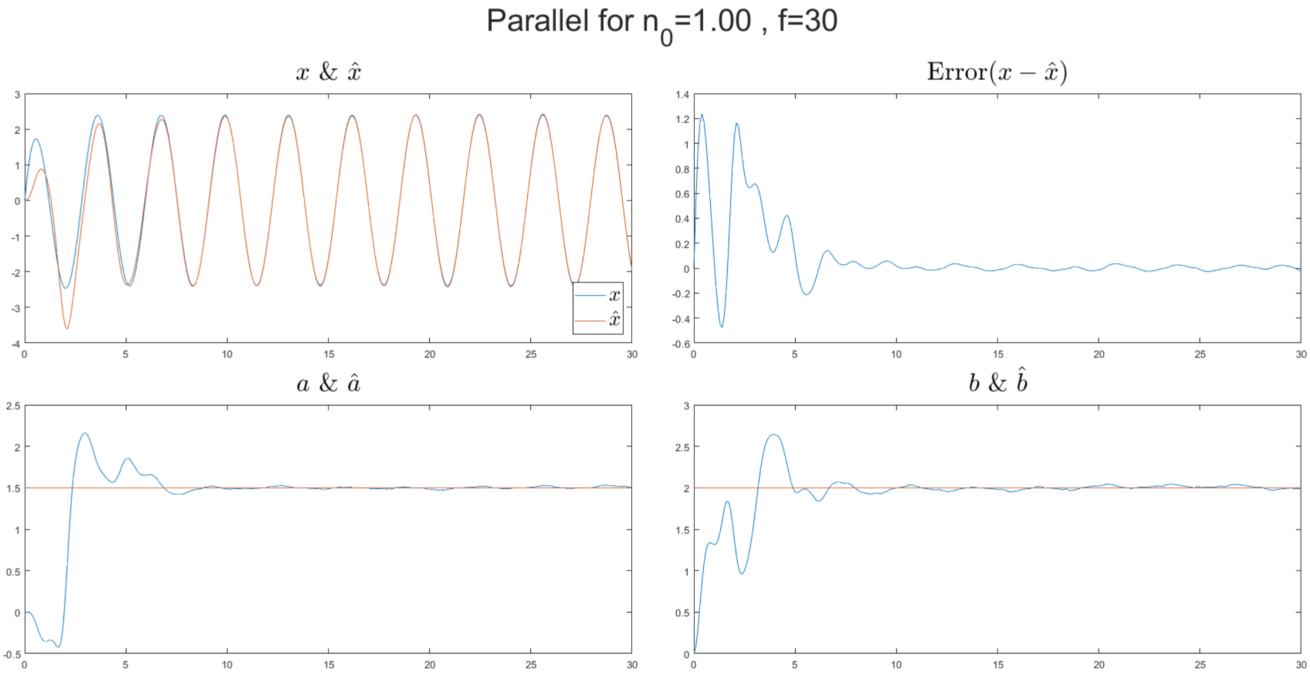
Now I set the noise signal values to the requested values, $n_0 = 0.25$, f =30, and I run the program again.

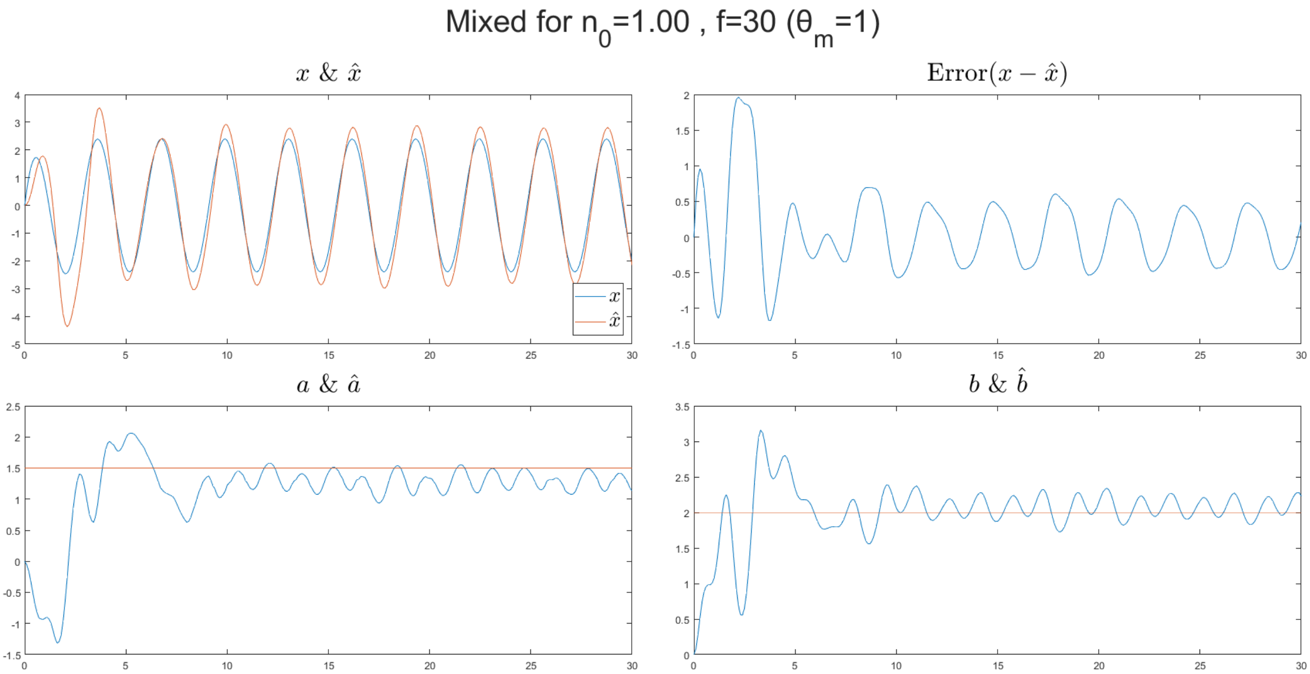




This time, due to the added noise, I notice that the error oscillates throughout the tests but taking small values, approximately equal to n_0. $\hat{a}$, $\hat{b}$ approach the true values of a, b and perform oscillation around them. Here again the two topologies show identical behavior.

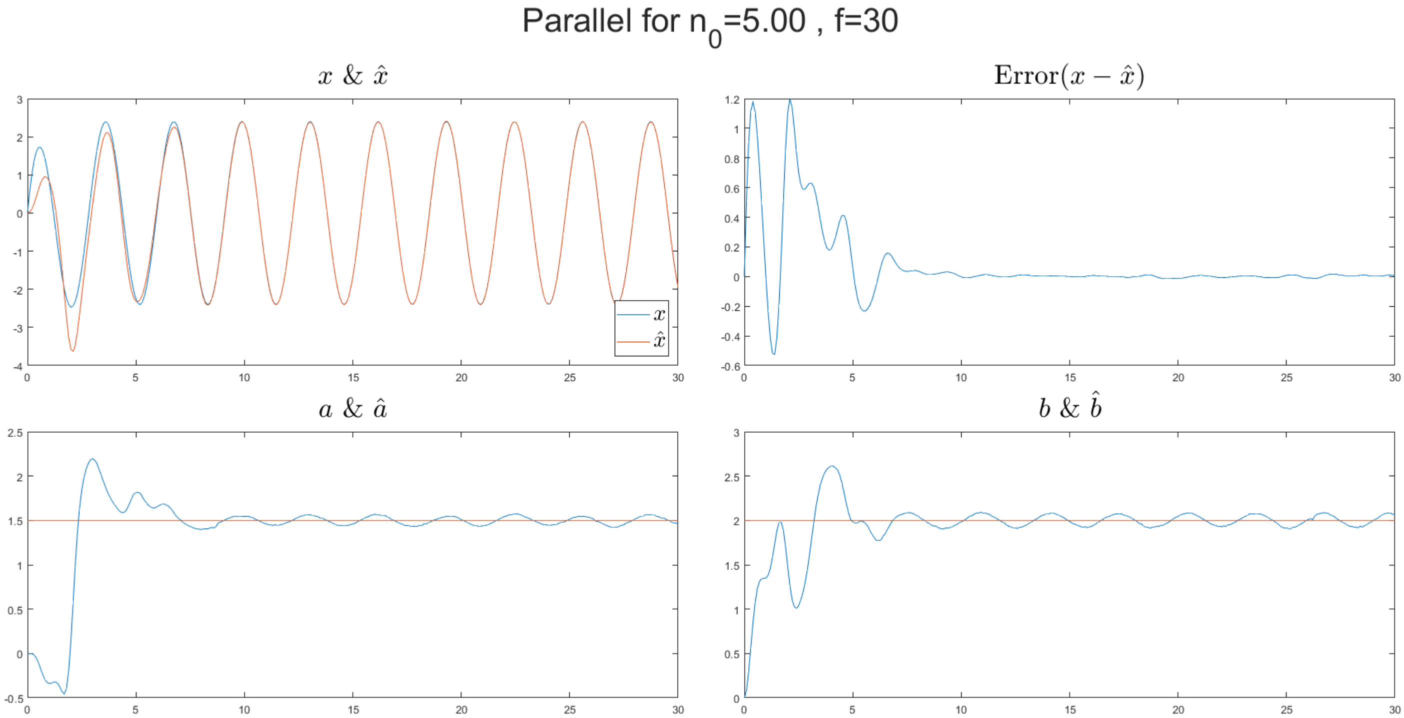
Now I increase the amplitude of the noise signal to $n_0 = 1$.





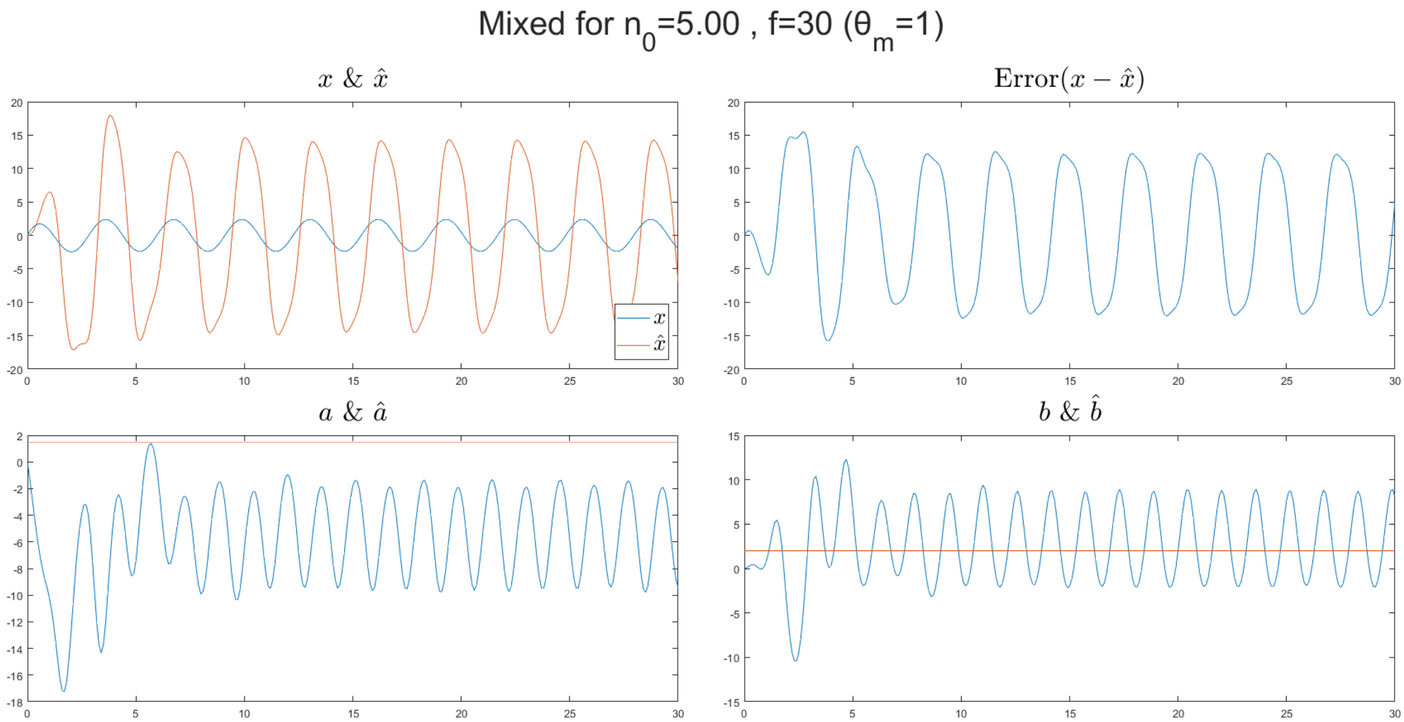
In this particular case, I notice that the parallel topology makes a better approximation, with the error being almost zero in the steady state and the parameter values converging to the real ones with some small fluctuations. Instead the mixed topology is forced to approximate the true output and $\hat{a}$, $\hat{b}$ to oscillate close to the values of a, b.

Now I give the noise width a fairly large value $n_0 = 5$, which is about 2.5 times the width of the output signal!



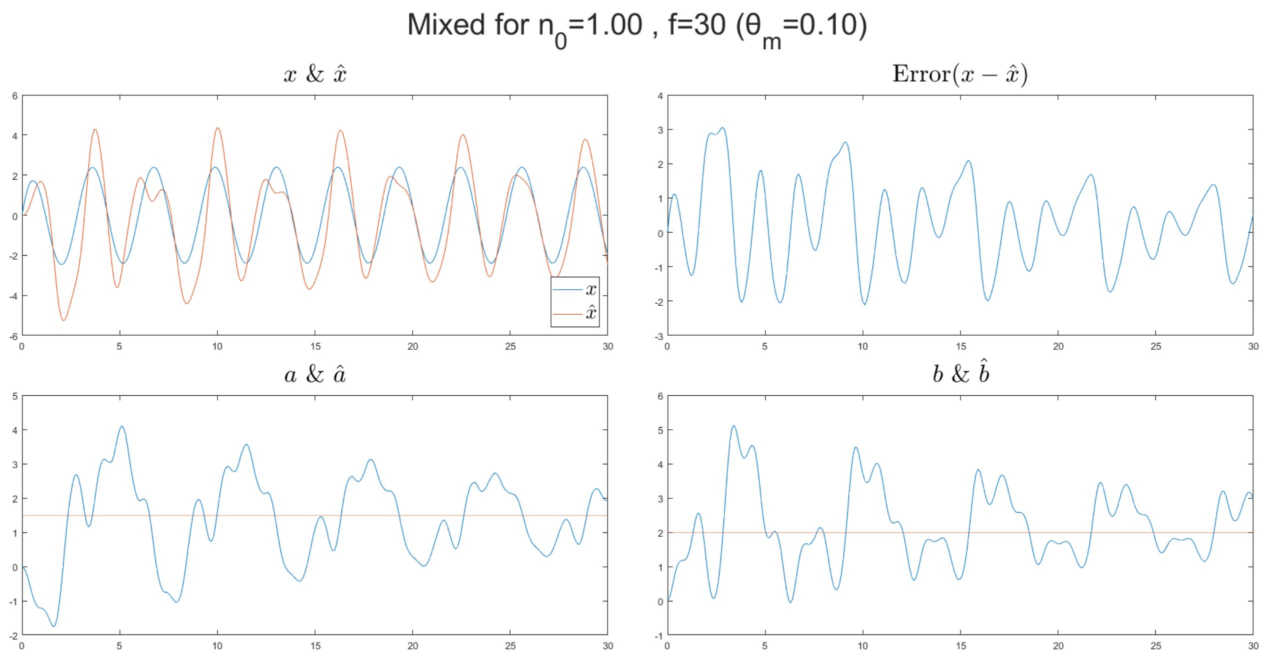
Despite the large noise values added, the parallel structure manages to approximate the output signal very well. The error takes a maximum value in the first seconds of the test, about 1.2, while $\hat{a}$, $\hat{b}$ converge again to the correct values performing a small-amplitude oscillation.

On the contrary, as shown in the figures below, the mixed structure performs very poorly. $\hat{x}$ never approaches the true output of the system with the error often taking values > 10, while the parameter estimates also take wrong values, failing to converge.

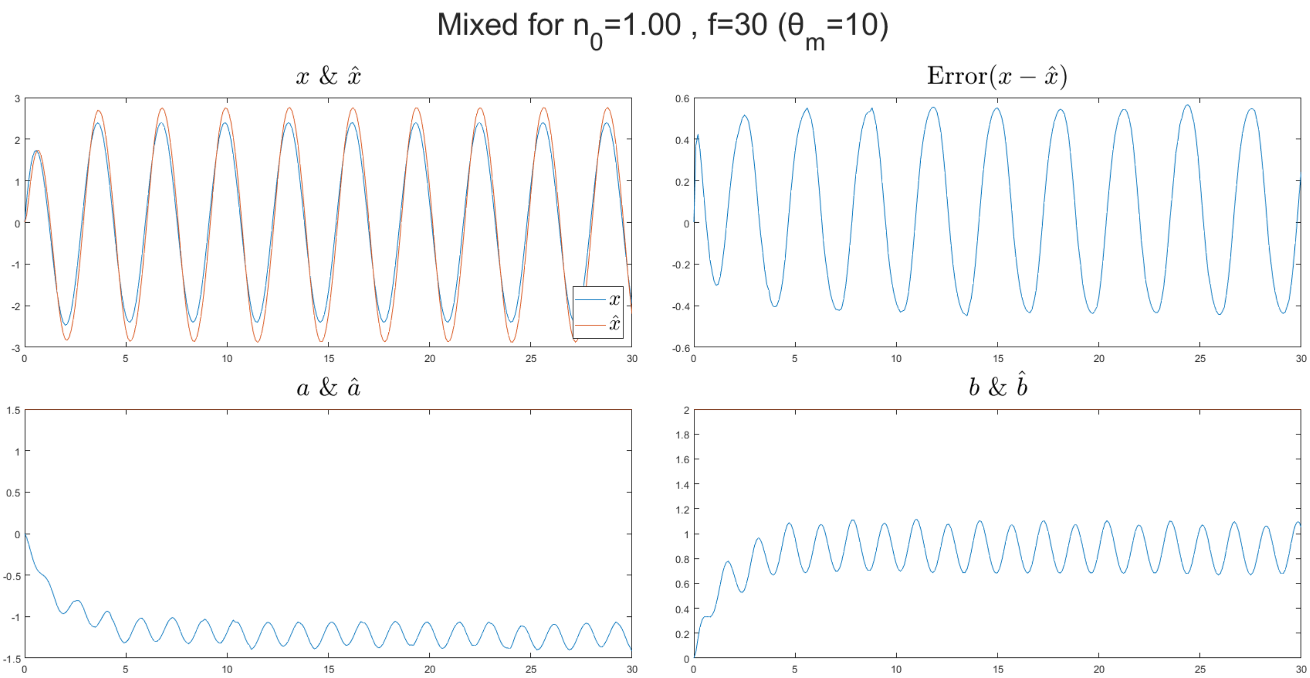


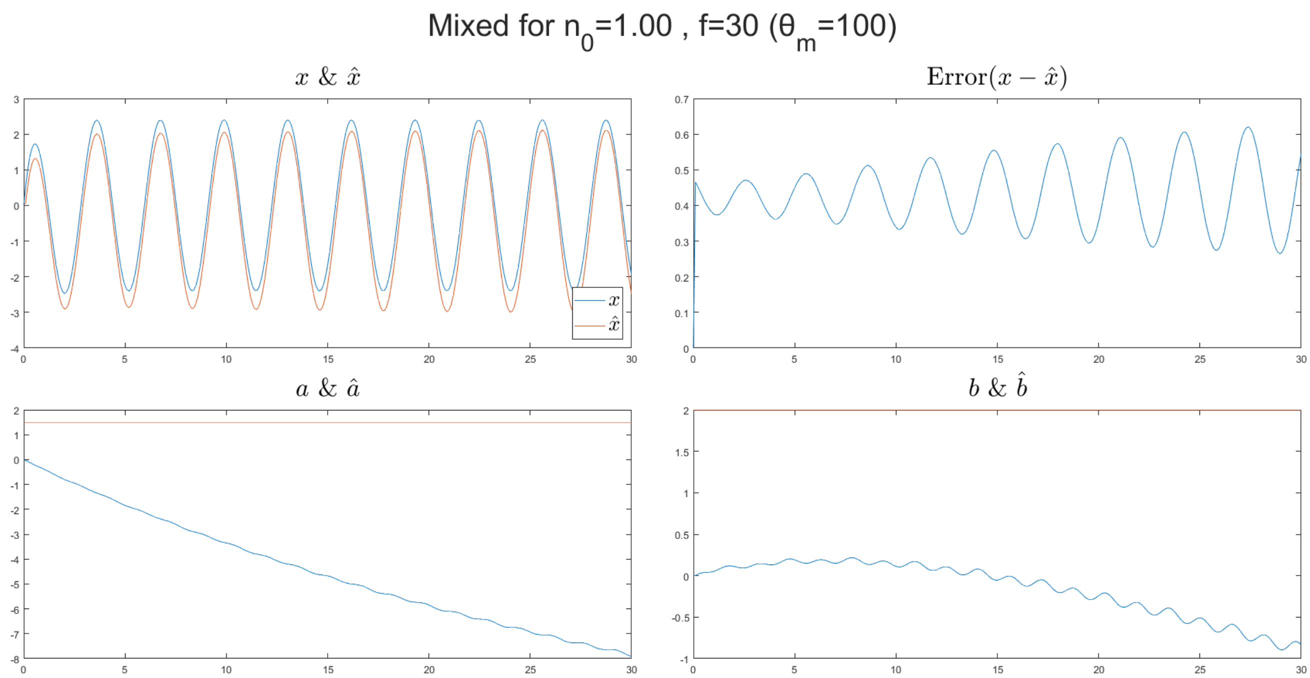
Notice that in all the above tests on the mixed structure $\theta_m = 1$. At this point I experiment with the θ_m parameter by trying different values. As an example, I use $n_0 = 1$.

First I reduce the value of θ_m to 0.1. This does not help by increasing both the error of the estimate of the attack and the deviation of the parameter estimates:



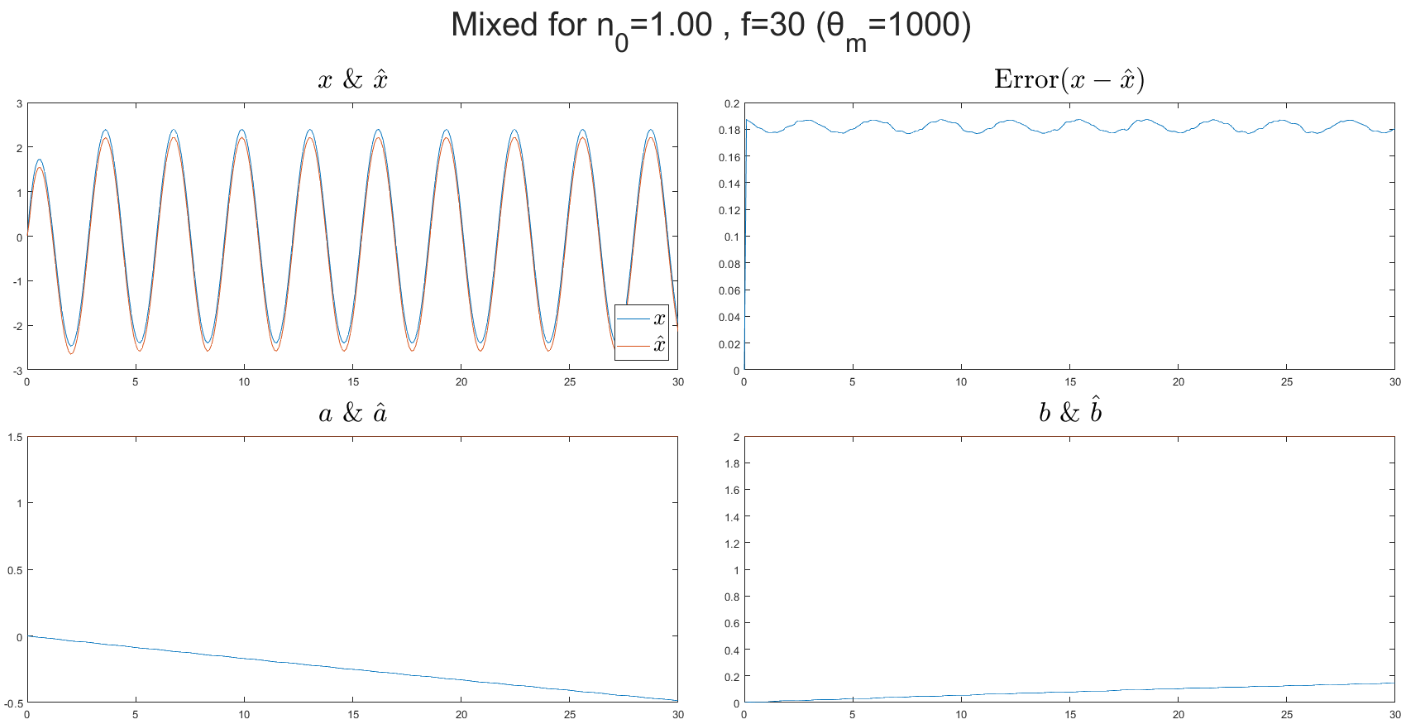
Then I try increasing the value of θ_m to 10 & 100.

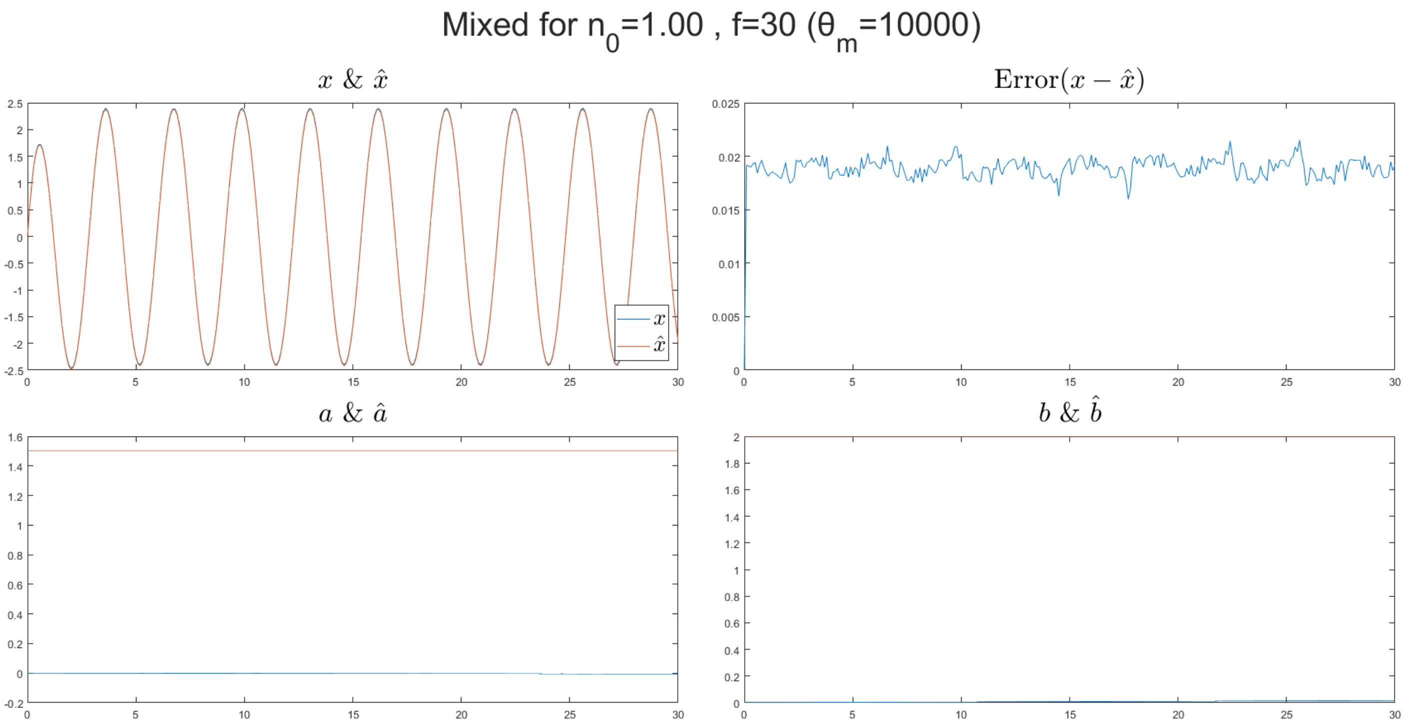




In the case where $\theta_m = 10$ I notice a reduced error with the estimated parameters, however, taking marginal values from the real ones. For $\theta_m = 100$ the estimate shows strange behavior, with the error increasing continuously and the system parameters taking even worse values.

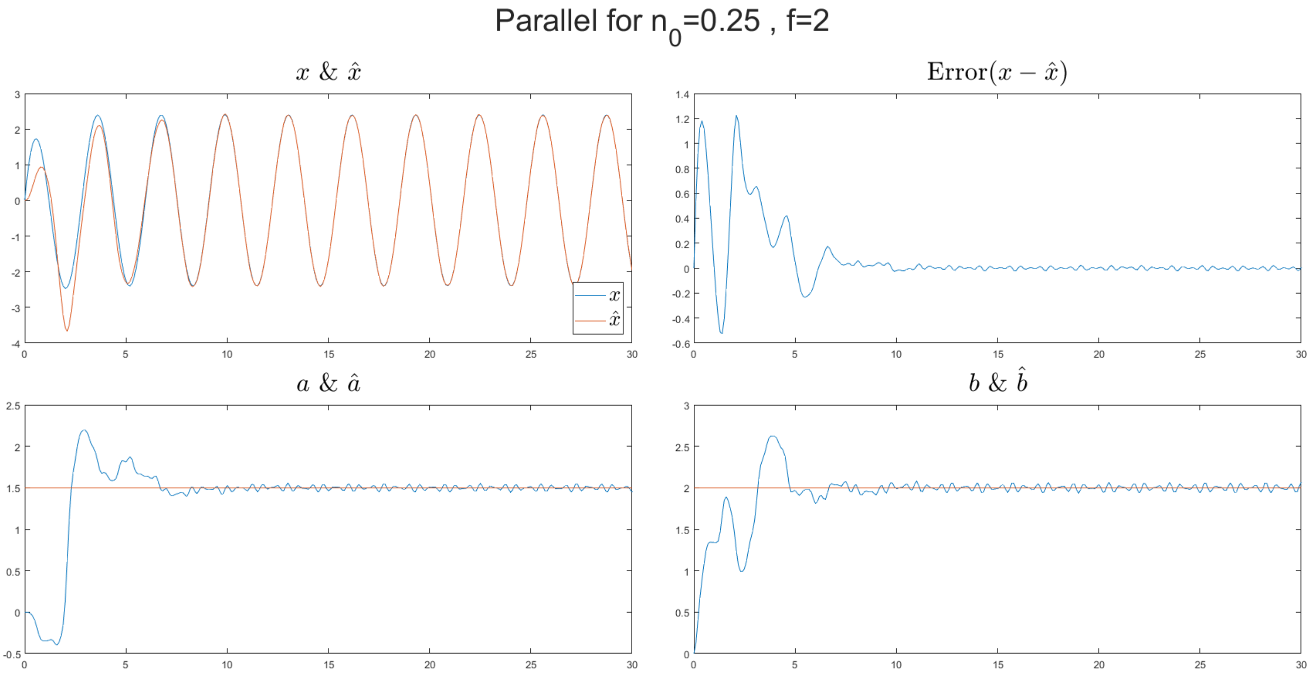
Further increasing the value of θ_m to 1000 & 10000 I notice that the error in the output practically becomes zero, but the estimates of a, b become useless, as they converge to 0.

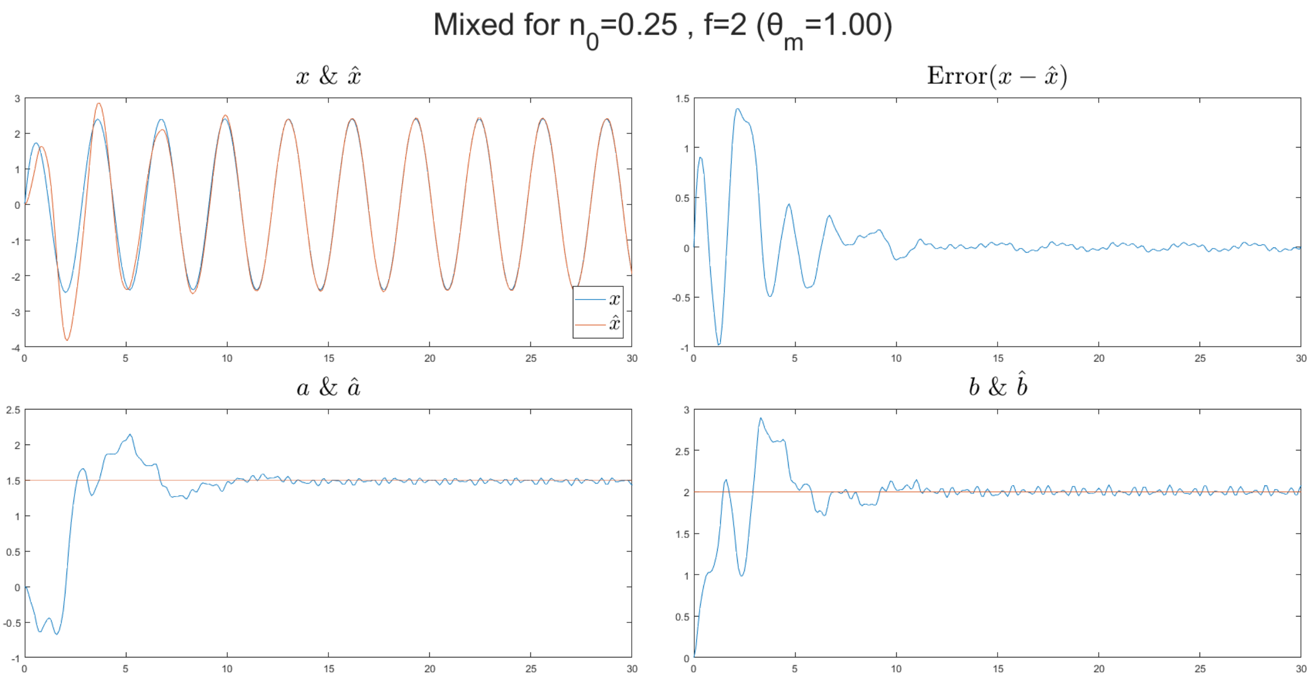




The rest of Problem 2 is devoted to how the variation of the noise coherency f affects the Lyapunov estimations. For this purpose we return to the initial values $n_0 = 0.25$, $\theta_m = 1$ and change the frequency f.

First I try a fairly small value of $f = 2$:

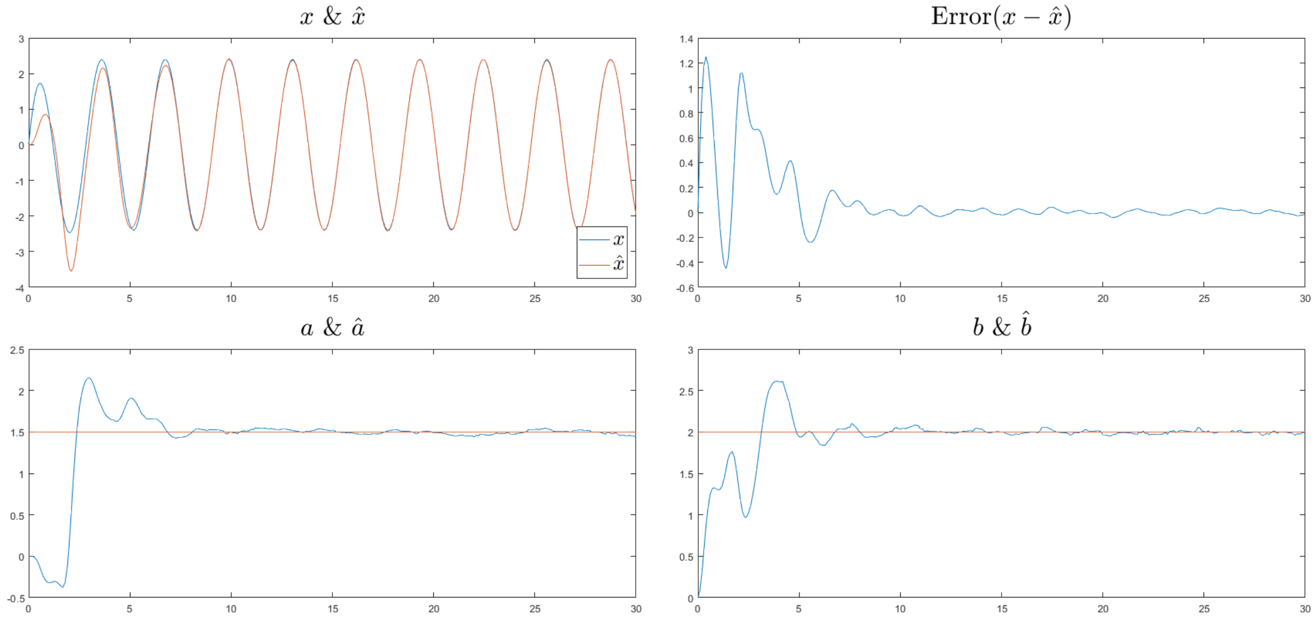




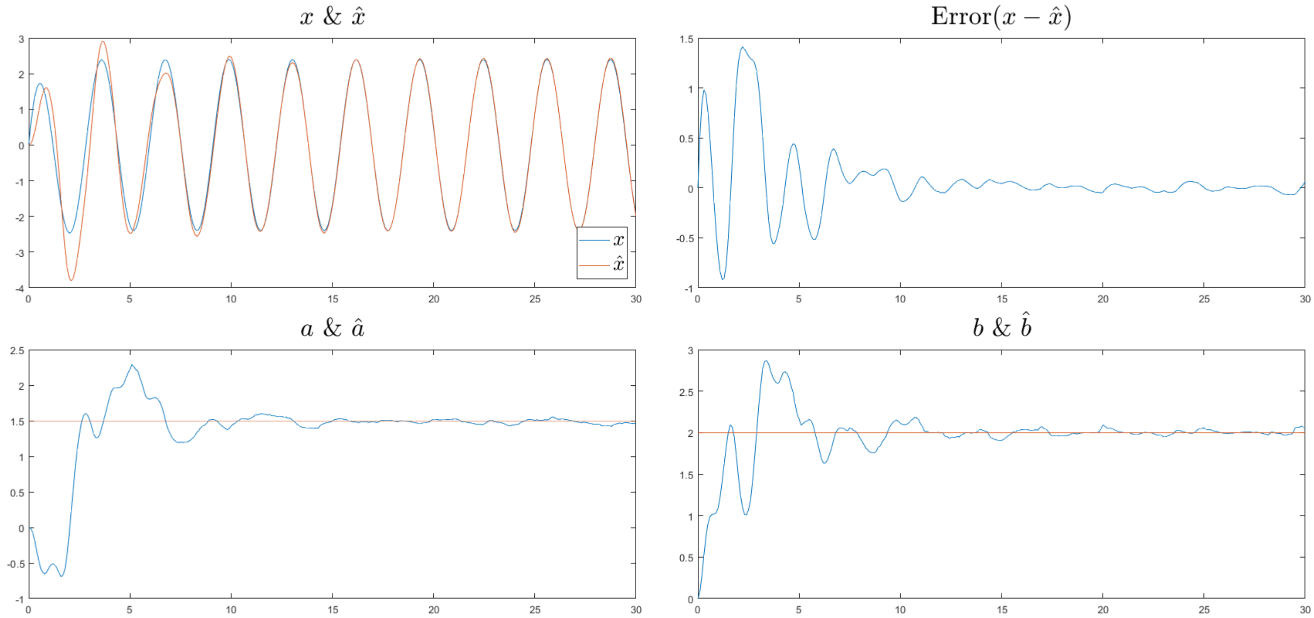
For both structures I observe identical behavior similar to that for $f = 30$, i.e. the error is maximized at the beginning of the test, while the estimated parameters approach the true values and perform a small-amplitude oscillation around them.

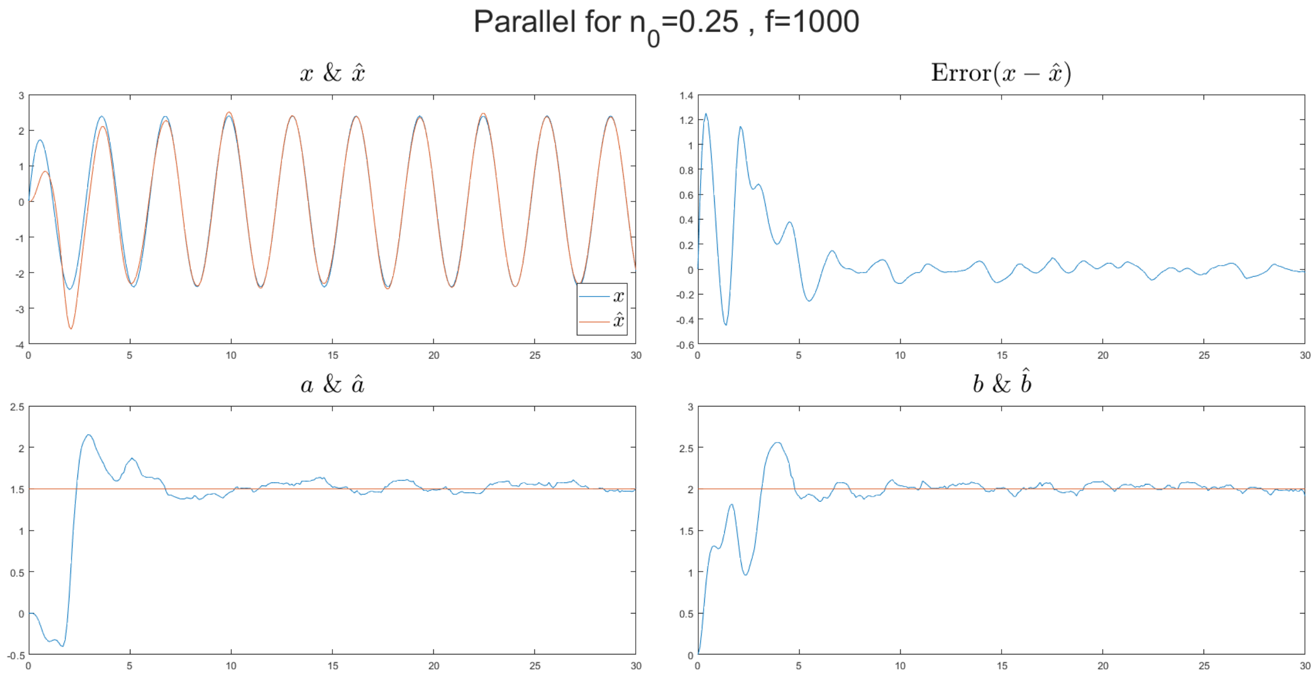
Now I'm trying some frequencies >30.

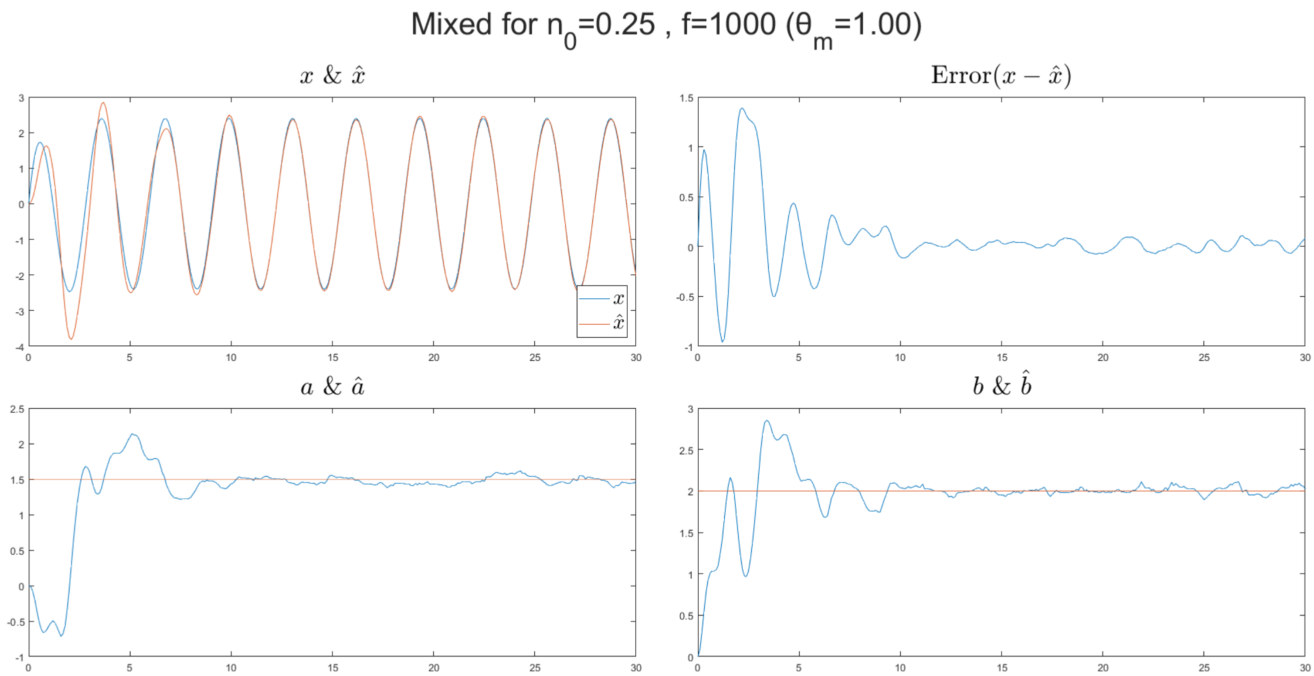
Parallel for n_0=0.25 , f=100



Mixed for n_0=0.25 , f=100 (θ_m=1.00)

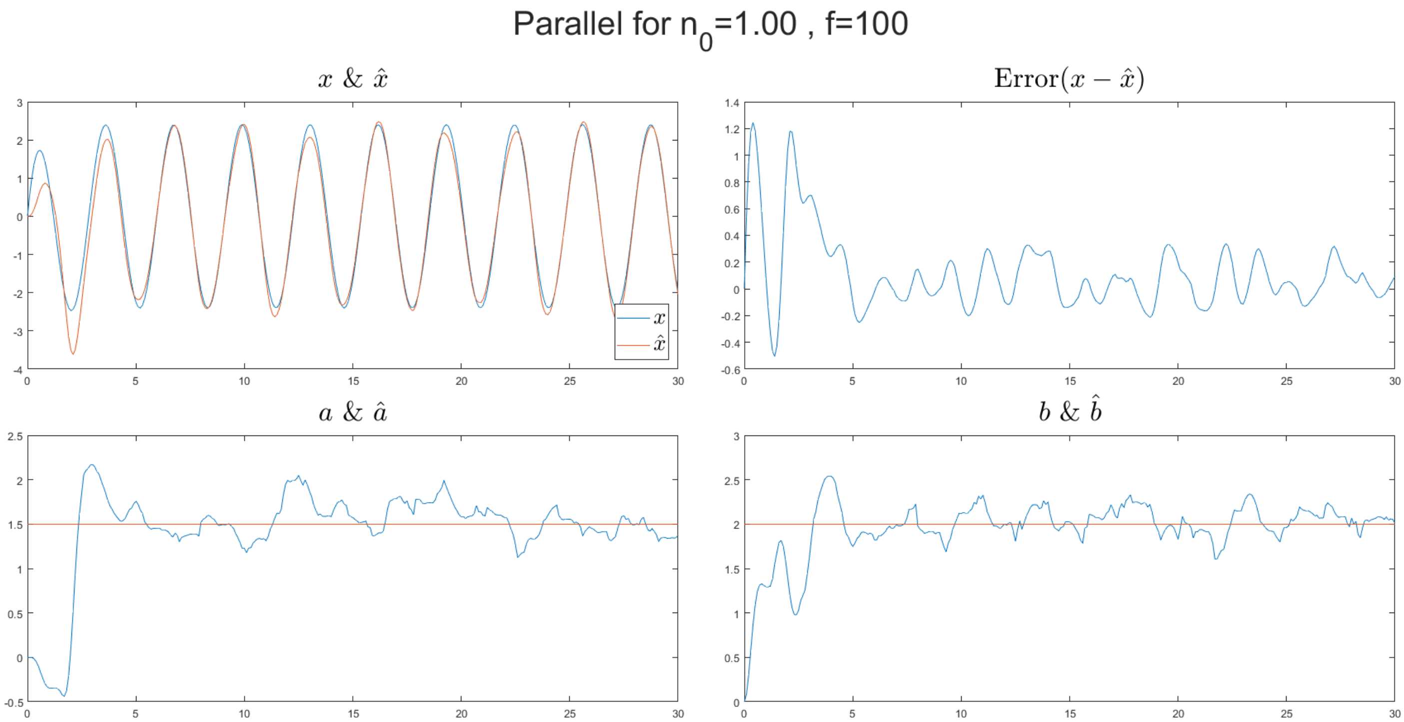


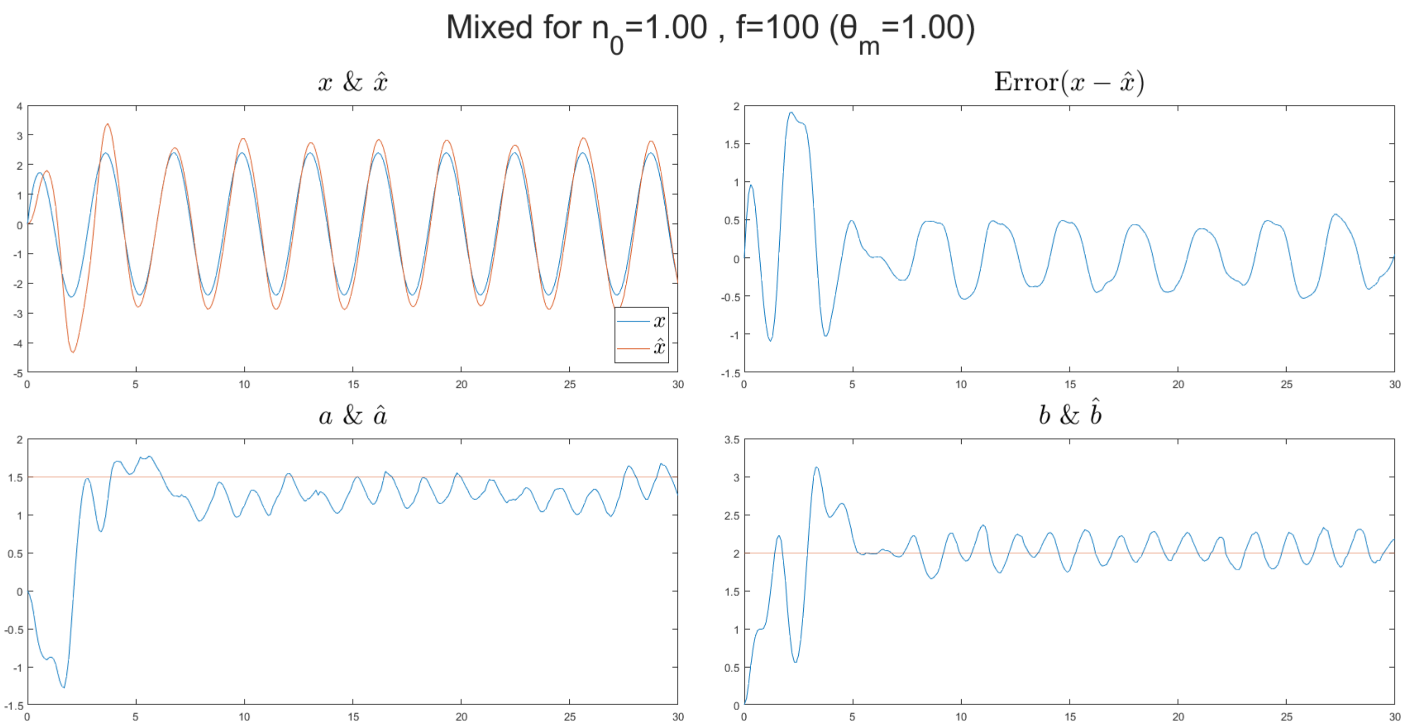




Again I observe similar behavior to before, with both estimators producing similar results, without difficulty approximating the true values of the system.

For this reason I try a combination of increased noise amplitude and frequency, $n_0 = 1$, $f = 100$:





In this case, we observe that the parallel structure produces a slightly better estimate, while increasing the value of θ_m in the mixed structure, as above, reduces the error, but the estimates $\hat{a}$, $\hat{b}$.

In conclusion, applying the Lyapunov method I observed that both the parallel and mixed structures are capable of producing reliable estimates, in the absence of noise. By introducing noise into the output measurement I again observe similar behavior from both topologies, but by increasing the noise amplitude (frequency did not seem to affect much), I see that the mixed structure produces estimates with large error, which makes sense as the noise is introduced into the estimation equation of $\hat{x}$, while it is introduced quadratically into the d.e. renewal of $\hat{\theta_1} = \hat{a}$. In contrast, the parallel structure remains almost unaffected by the increase in noise and continues to produce accurate and reliable results. Finally, although with the parameter θ_m the mixed structure gives us more freedom in the design of the estimator, in my tests I did not notice a significant improvement of the estimates by varying the parameter. I therefore come to the conclusion that if we cannot guarantee the measurement of signals without noise, the parallel structure should be preferred.

3 Problem 3 - Higher Order System

3.1 Theoretical Analysis

In this problem we will deal with a second-order system:

$$\dot{x} = Ax + Bu \ , x_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{1}$$

, where x are the states of the system and $u = 7.5cos(3t) + 10cos(2t)$ the input , while $A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$, $B = \begin{bmatrix} b_1 & b_2 \end{bmatrix}$ fixed but unknown matrices.

The Lyapunov method will be used to estimate the parameters, specifically the *Parallel Structure* recognition system. Corresponding to the 2nd Problem, I first define the estimation of the states x of the system:

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u \tag{2}$$

I set the identification error $e = x - \hat{x}$ and derive with respect to time:

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

$$\dot{e} = Ax + Bu - \hat{A}\hat{x} - \hat{B}u \xRightarrow{\pm A\hat{x}}$$

$$\dot{e} = Ax - A\hat{x} + A\hat{x} - \hat{A}\hat{x} + Bu - \hat{B}u \Rightarrow \quad \dot{e} = A(x - \hat{x}) - (\hat{A} - A)\hat{x} - (\hat{B} - B)u \Rightarrow$$

$$\dot{e} = Ae - \tilde{A}\hat{x} - \tilde{B}u \tag{3}$$

, where $\tilde{A} = \hat{A} - A$, $\tilde{B} = \hat{B} - B$, are the parametric errors.

Let the Lyapunov function be:

$$V = \frac{1}{2}e^T e + \frac{1}{2\gamma_1}\,\mathrm{tr}\{\tilde{A}^T\tilde{A}\} + \frac{1}{2\gamma_2}\,\mathrm{tr}\{\tilde{B}^T\tilde{B}\} \tag{4}$$

, where tr is the trace of a matrix and $\gamma 1, \gamma 2$ are arbitrary positive constants.

I derive the Lyapunov function with respect to time:

$$\dot{V} = e^T\dot{e} + \frac{1}{\gamma_1}\,\mathrm{tr}\{\tilde{A}^T\dot{\hat{A}}\} + \frac{1}{\gamma_2}\,\mathrm{tr}\{\tilde{B}^T\dot{\hat{B}}\} \Rightarrow$$

$$\dot{V} = e^T Ae - e^T\tilde{A}\hat{x} - e^T\tilde{B}u + \mathrm{tr}\{\frac{1}{\gamma_1}\tilde{A}\dot{\hat{A}}^T\} + \mathrm{tr}\{\frac{1}{\gamma_2}\tilde{B}\dot{\hat{B}}^T\} \Rightarrow$$

$$\dot{V} = e^T Ae - \mathrm{tr}\{\tilde{A}\hat{x}e^T\} - \mathrm{tr}\{\tilde{B}ue^T\} + \mathrm{tr}\{\frac{1}{\gamma_1}\tilde{A}\dot{\hat{A}}^T\} + \mathrm{tr}\{\frac{1}{\gamma_2}\tilde{B}\dot{\hat{B}}^T\} \Rightarrow$$

$$\dot{V} = e^T Ae + \mathrm{tr}\{\frac{1}{\gamma_1}\tilde{A}\dot{\hat{A}}^T + \frac{1}{\gamma_2}\tilde{B}\dot{\hat{B}}^T - \tilde{A}\hat{x}e^T - \tilde{B}ue^T\} \tag{5}$$

Choose:

$$\dot{\hat{A}} = \gamma_1(\hat{x}e^T)^T$$

$$\dot{\hat{B}} = \gamma_2(ue^T)^T$$

, so:

$$\dot{V} = e^T Ae \leq 0 \tag{6}$$

Similar to the 2nd Problem, it proves that $\lim_{t\to\infty} e(t) = lim_{t\to\infty}\dot{\hat{A}}(t) = lim_{t\to\infty}\dot{\hat{B}}(t)$ =0

3.2 Implementation & Results

3.2.1 Implementation in MATLAB

As the MATLAB files are provided in the deliverables folder, the steps I followed for the implementation are listed schematically:

- Setting the actual values of A, B arrays and initializing γ_1, γ_2.
- Initialize vector t (time), input u.
- Computation of the states x by solving the system d.e. of the system.
- Finding $\hat{A}$ and $\hat{B}$ with the Lyapunov Method using a parallel structure, as discussed above.
- Print the necessary graphs.

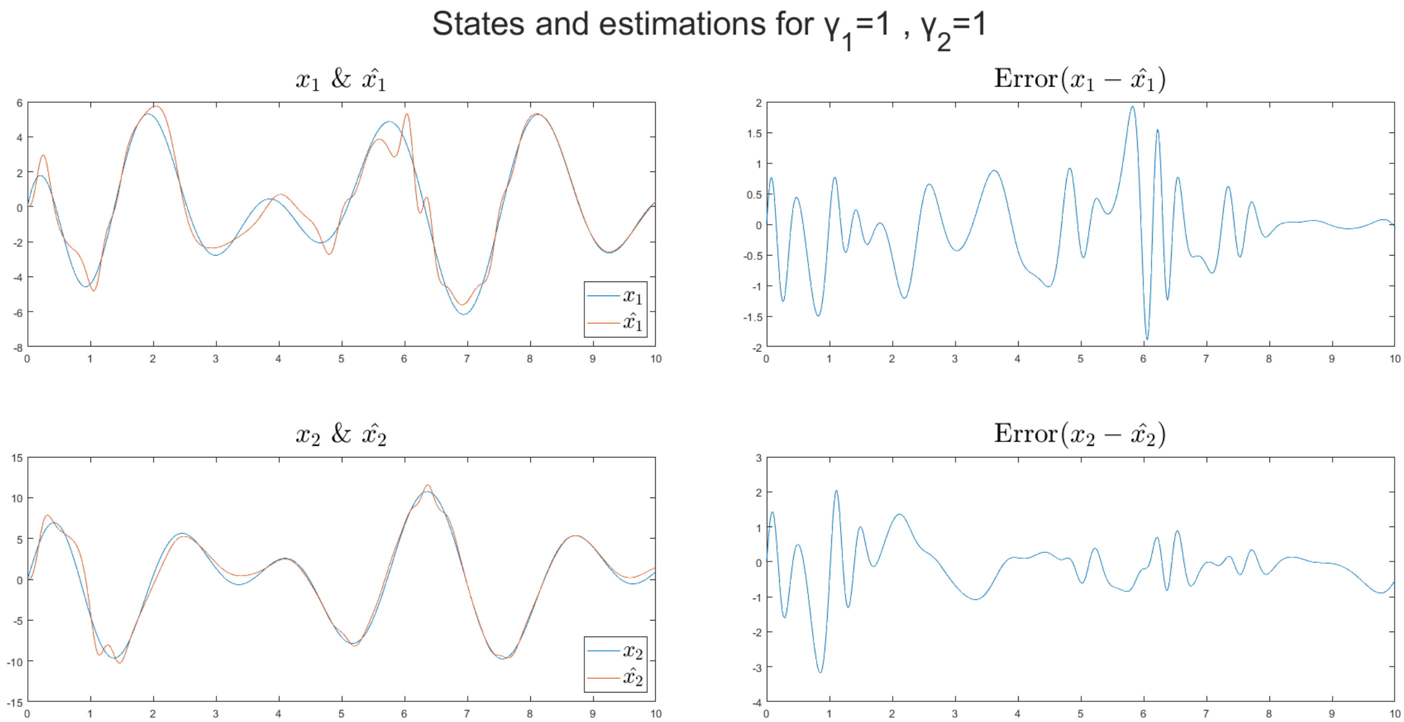
To estimate the output and parameters of the system, the following system of differential equations was used, which was solved using *ode45*:

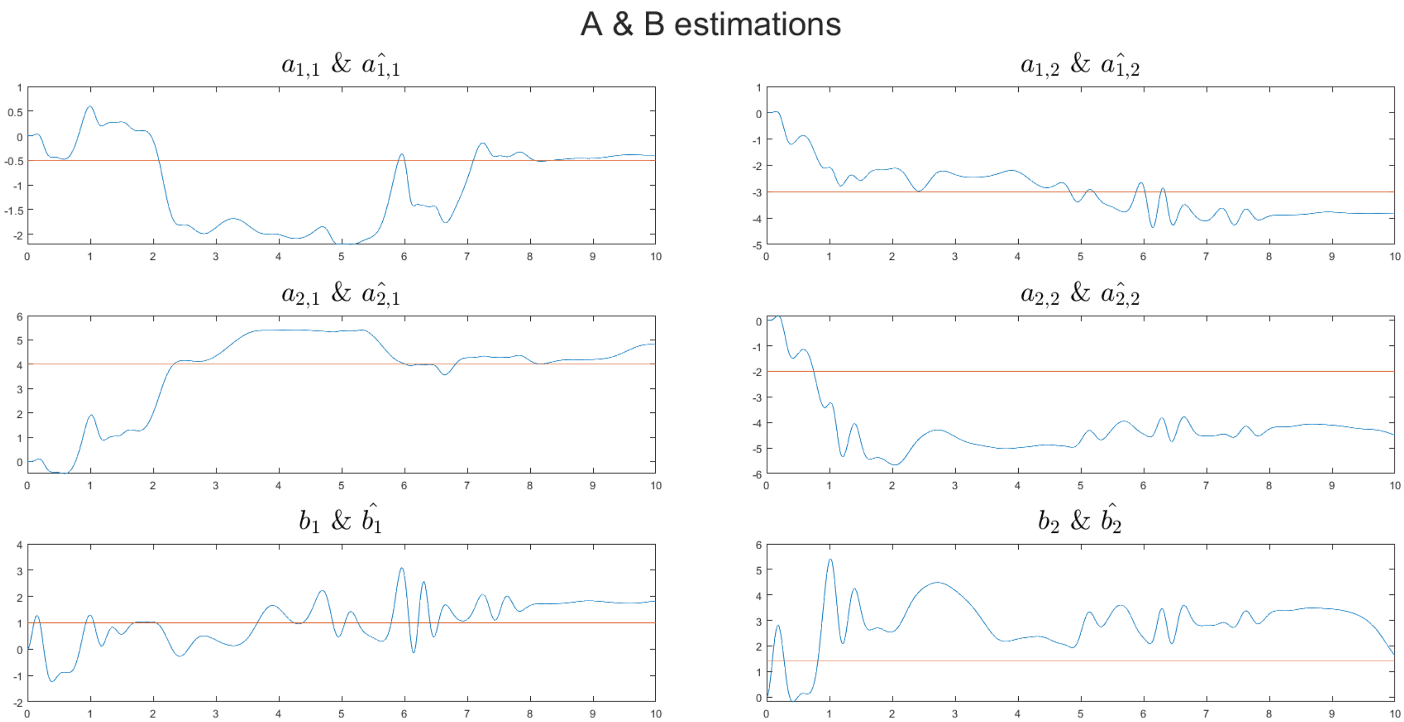
$$\begin{cases} \dot{x_1} = a_{1,1}x_1 + a_{1,2}b_1u \\ \dot{x_2} = a_{2,1}x_1 + a_{2,2}b_1u \\ \dot{\hat{x_1}} = \hat{a_{1,1}}\hat{x_1} + \hat{a_{1,2}}\hat{b_1}u \\ \dot{\hat{x_2}} = \hat{a_{2,1}}\hat{x_1} + \hat{a_{2,2}}\hat{b_2}u \\ \dot{\hat{a_{1,1}}} = \gamma_1\hat{x_1}e_1 \\ \dot{\hat{a_{1,2}}} = \gamma_1\hat{x_2}e_1 \\ \dot{\hat{a_{2,1}}} = \gamma_1\hat{x_1}e_2 \\ \dot{\hat{a_{2,2}}} = \gamma_1\hat{x_2}e_2 \\ \dot{\hat{b_1}} = \gamma_2ue_1 \\ \dot{\hat{b_2}} = \gamma_2ue_2 \end{cases} \tag{7}$$

3.2.2 Results & Remarks

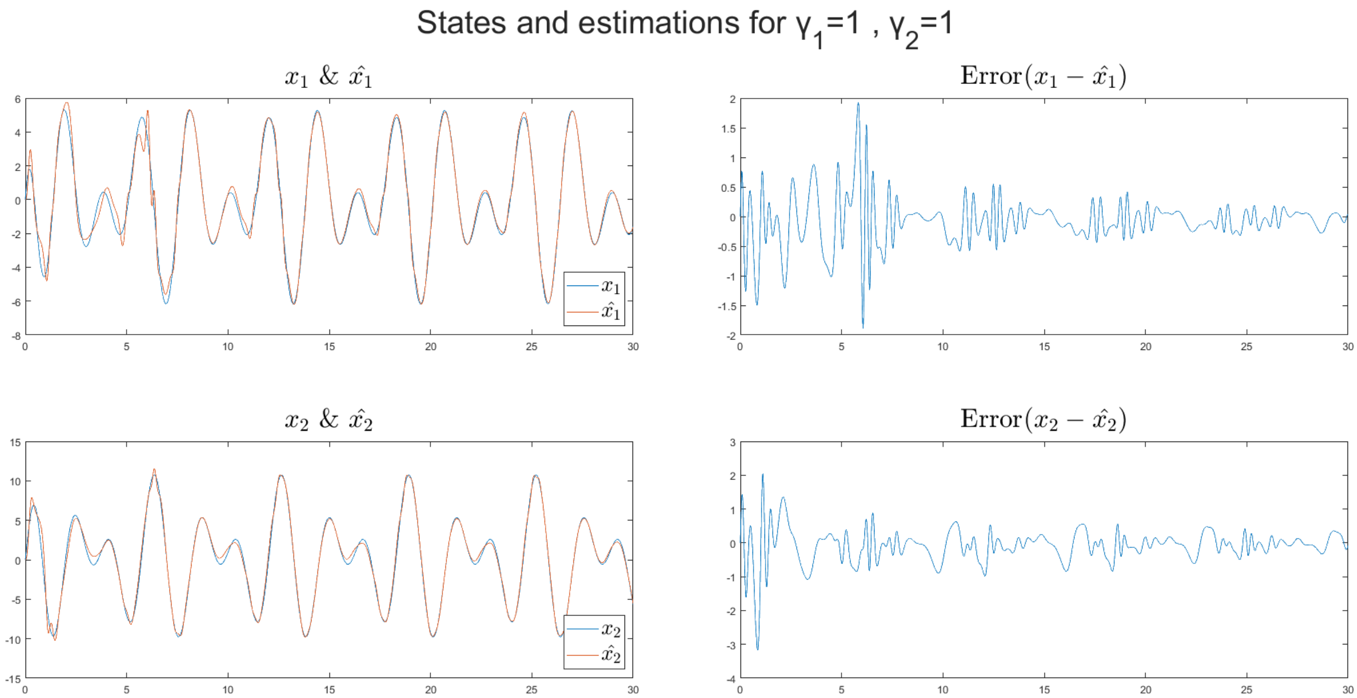
I start my tests with the simplest case, where $\gamma_1 = \gamma_2 = 1$.

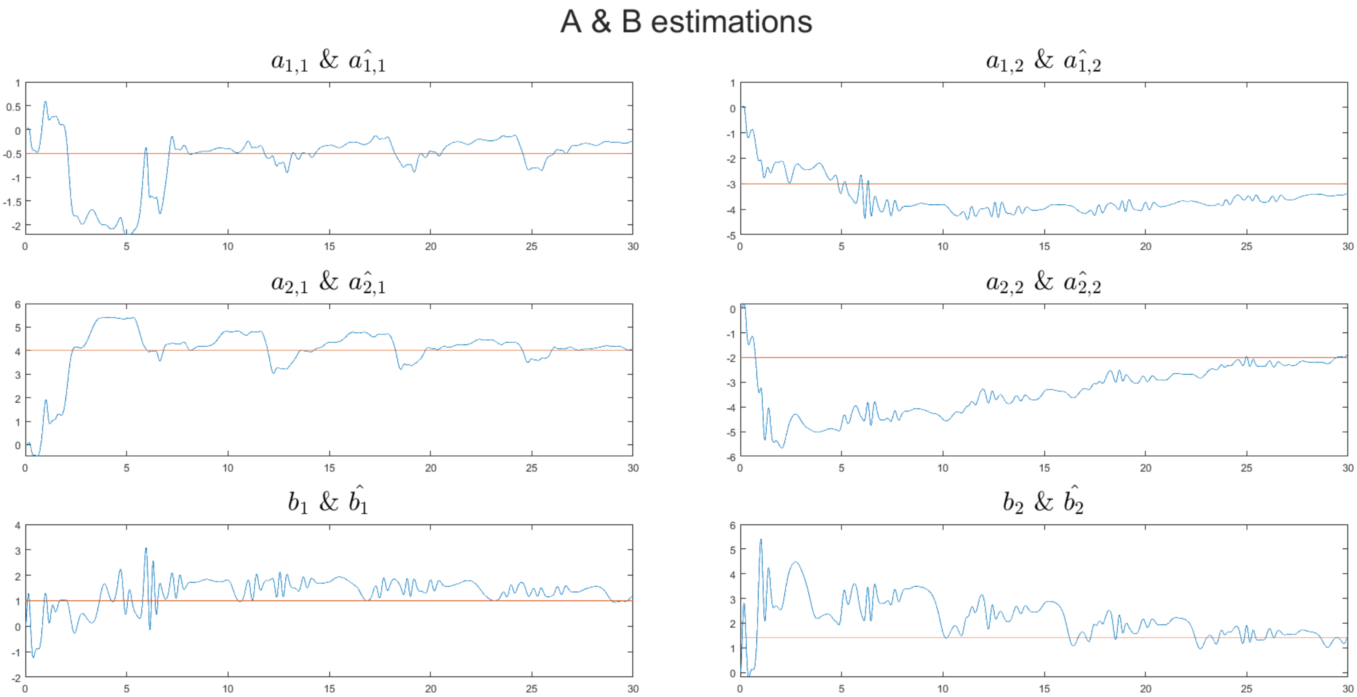
As can be seen in the figures below, the approximation of the states is quite good, while the approximations of the system parameters are close to their true values, but they still seem to vary to some extent, so before proceeding to other tests, I increase the time of test from 10 seconds to 30 seconds.





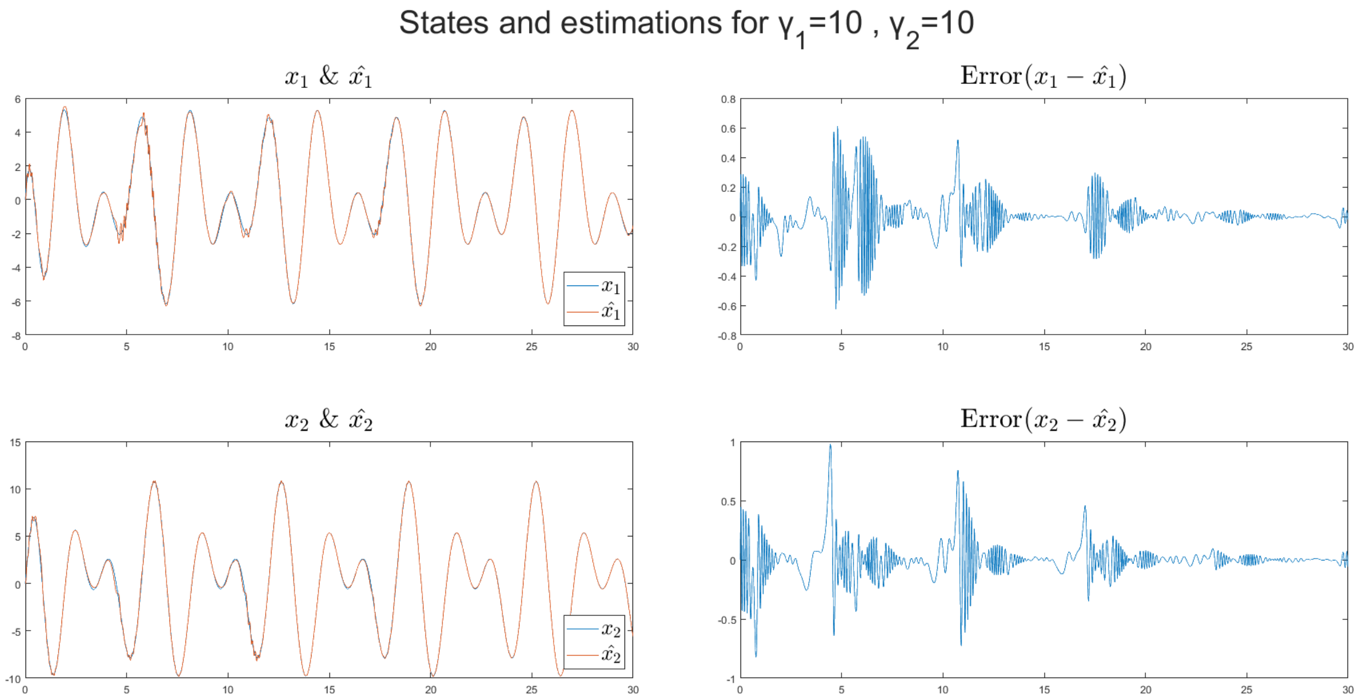
Here are the results for the 30 second test:

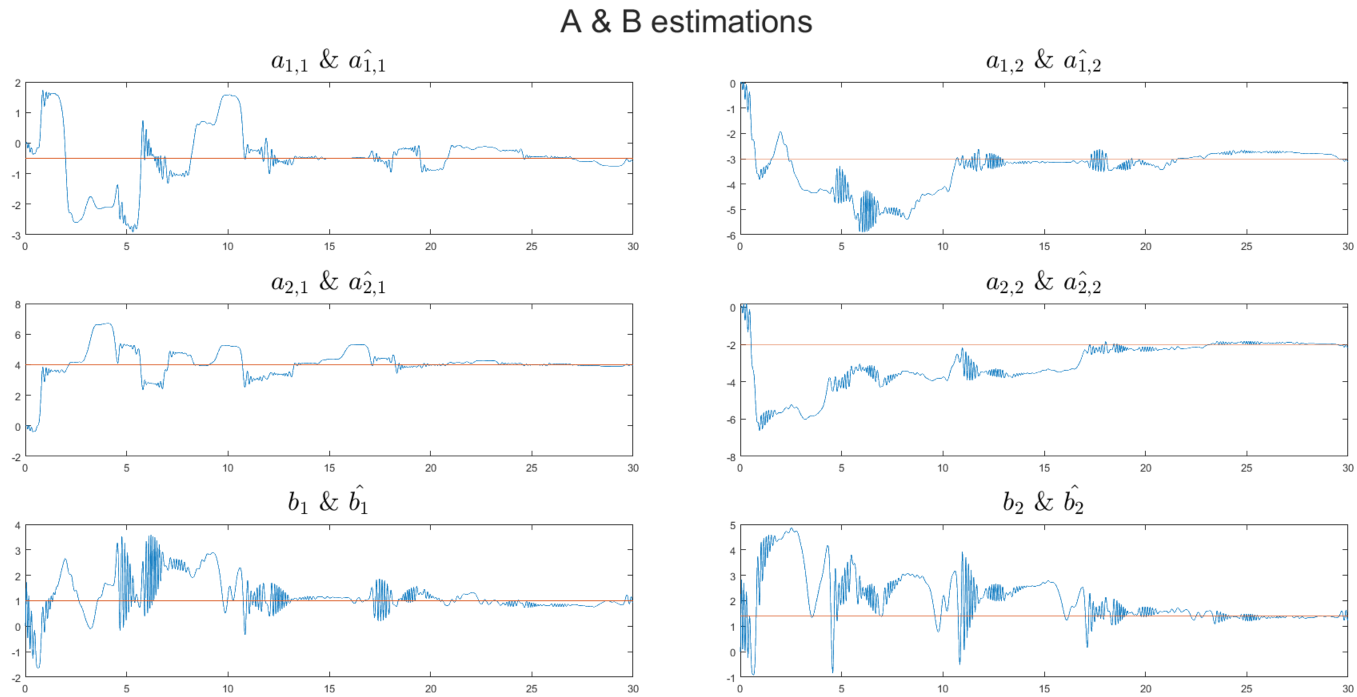




I notice that the extra test time helps to reduce the error and better estimate the parameters. Perhaps by giving even more time we can achieve perfect convergence and zero error, but then by changing γ_1, γ_2 we will try to speed up this convergence.

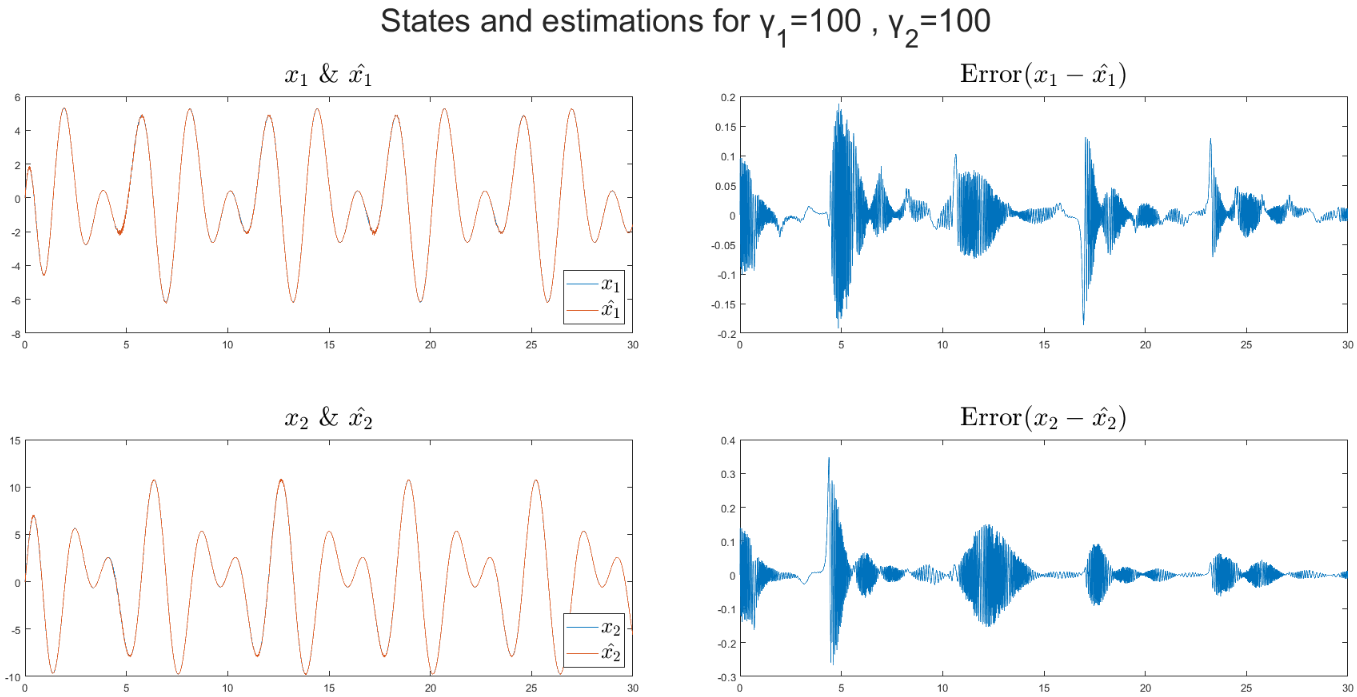
So the next test is for $\gamma_1 = \gamma_2 = 10$ while keeping the increased time interval of 30 seconds.

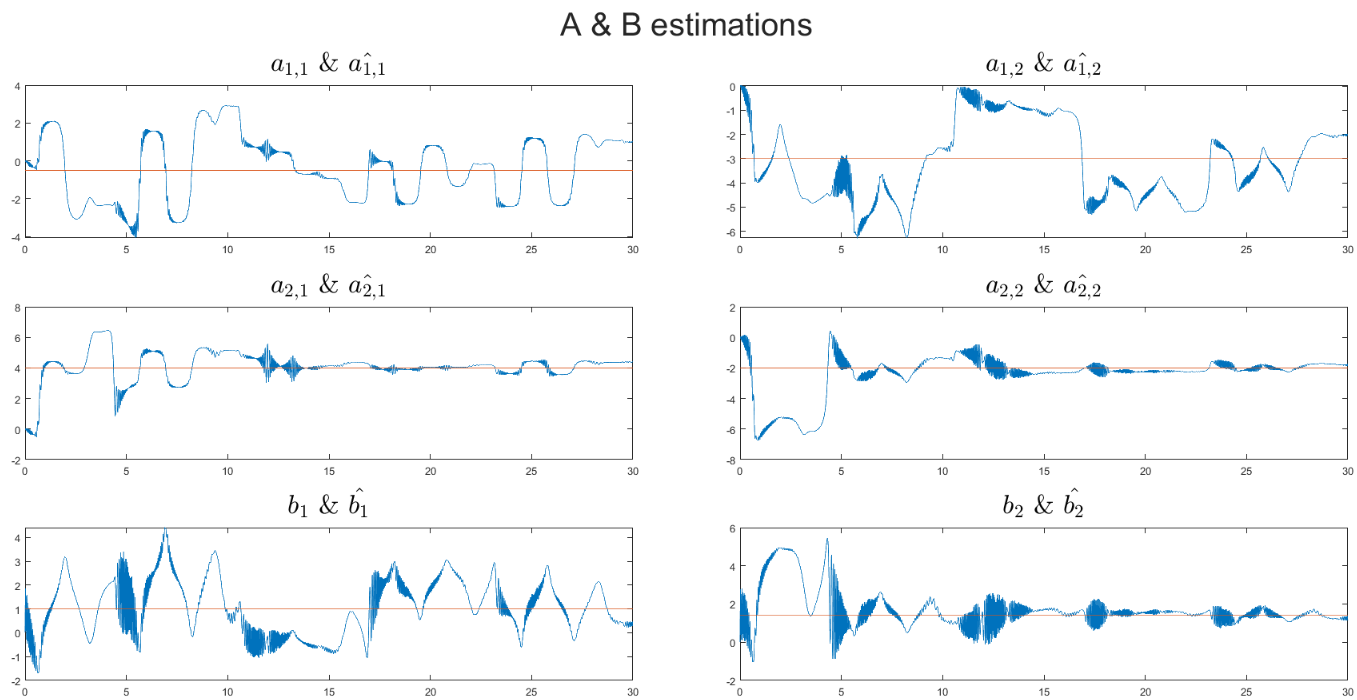




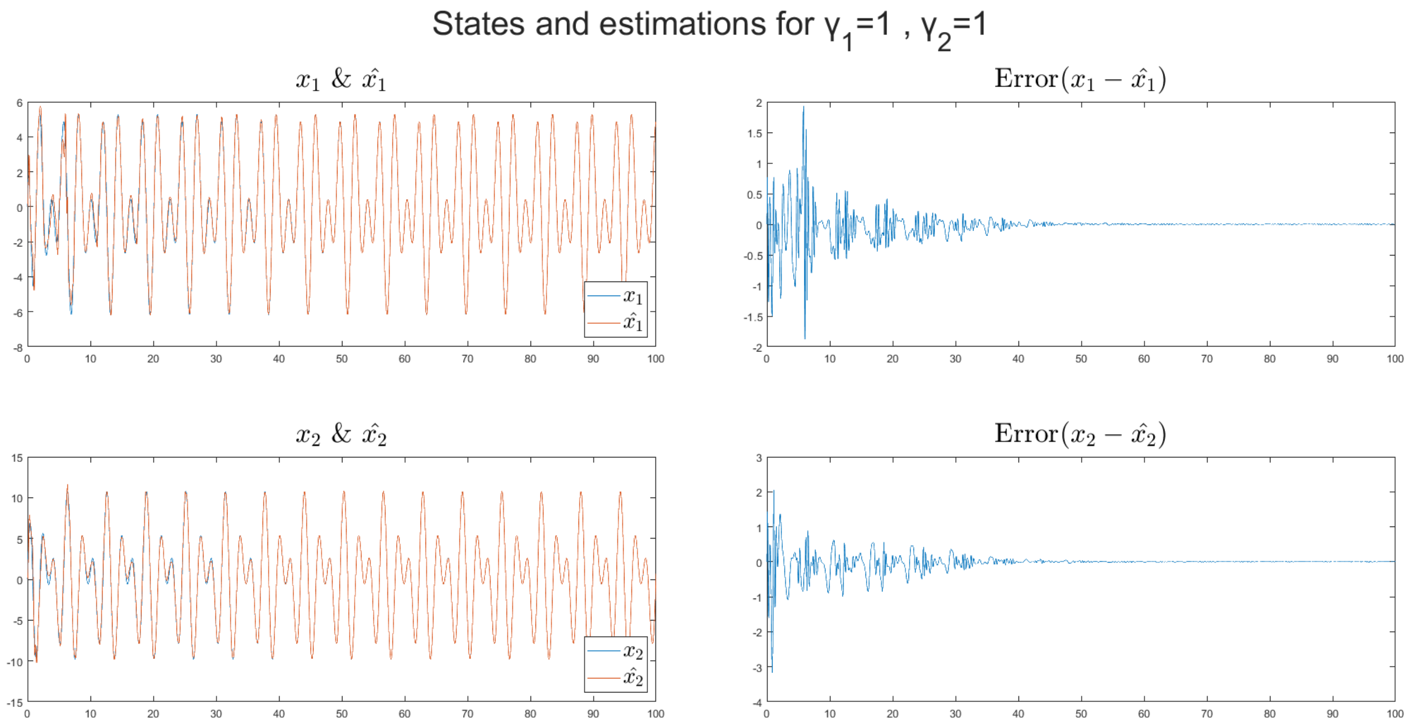
With this change I notice in all estimates a special behavior with high frequency oscillations, but finally the error takes a smaller value than before, while the estimated parameters have almost stabilized at the real values.

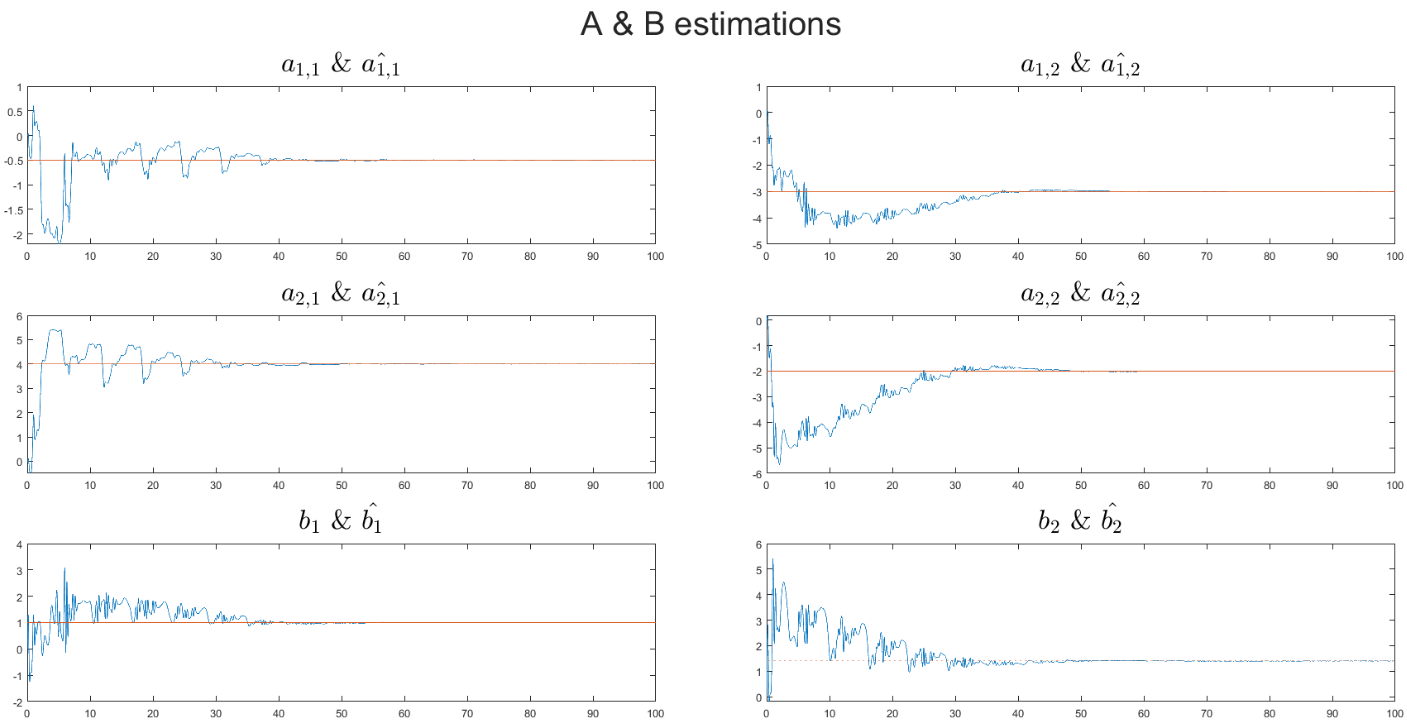
Further increasing the values of γ_1, γ_2 giving both the value 100, it is clear that the estimation of the states presents a very small error sacrificing, however, the accuracy in the estimations of the system parameters.





So I return to the initial values $\gamma_1 = \gamma_2 = 1$ and try to get better estimates of the A, B matrices by increasing the test time to 100 seconds.





I notice that at the beginning of the test the error is relatively large, but with the end of time, as the estimated parameters converge to the real values, the error is practically zero. The estimates obviously have practically zero error, but we can be sure by taking their values at the end of the test:

$$A = \begin{bmatrix} -0.5 & -3 \\ 4 & -2 \end{bmatrix} \; B = \begin{bmatrix} 1 & 1.4 \end{bmatrix}$$
$$\hat{A} = \begin{bmatrix} -0.5011 & -3.0019 \\ 4.008 & -1.9974 \end{bmatrix} \; \hat{B} = \begin{bmatrix} 0.992 & 1.4029 \end{bmatrix}$$

The largest error is of the order of 10^{-2}!

Thus, similar to the 1st Problem, I come to the conclusion that, in addition to the system, the application also plays a big role in the selection of the appropriate parameters γ_1, γ_2. That is, if we want the optimal estimation of the system parameters we will choose small enough γ and do longer tests. In another case, where it is important to accurately estimate the states, throughout the system operation we will prefer larger values of γ, which will lead to a faster minimization of the error e. Finally, although the 2 are equally significant, we will seek a balance between the two, of course making some compromises.