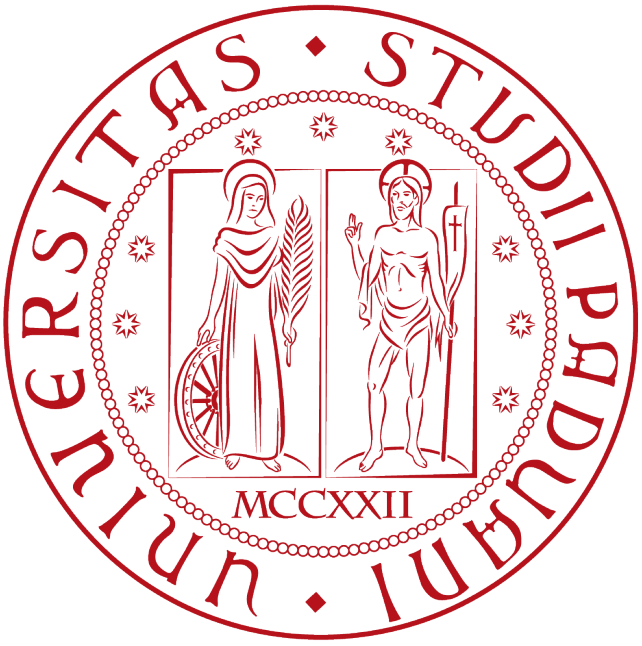

UNIVERSITY OF PADUA

Department of Management and Engineering

Second-cycle degree in Mechatronic Engineering

Laboratory activity 2: Identification of parameters

Professor
Dr. Luca Schenato

Student
Luca Zanrosso
1205388

Academic Year 2018 - 2019

Contents

Introduction

The activity consists in estimating unknown parameters from data using the least squares estimate. In particular:

- The first part concerns the estimation of a scalar parameter using some measures affected by noise
- In the second part, starting from the estimation of the model parameters, we have to choose, according to the level of complexity, the model that best reflects the real one.
- The third and final part require the identification of damping parameters for a DC motor using the input/output data relationship. Finally, a PD position controller is designed.

All parts are based on synthetic data of abstract systems, so only matlab and simunlink are required. The available data are not real measurements, but simple simulations, so no particular hardware and software setup is necessary.

1 Estimation of a scalar parameter

The first part of the laboratory consists in estimating an unknown parameter, first assuming to know the variance of the noise, and then estimating the variance itself. Finally, assuming that the parameter is not constant, we have to make an estimate using a forgetting factor.

Consider the following measurement model, also represented in Figure 1:

$$y_i = \theta + v_i, \qquad i = 1, ..., t \tag{1.1}$$

where $\theta = 1$ is the unknown parameter and v_i is i.i.d white Gaussian noise with unit variance, i.e. $v_i \sim \mathcal{N}(0, \sigma^2)$ where $\sigma = 1$.

The least-square method is particularly useful for estimating model parameters that can be written in the form

$$y(i) = \phi_1(i)\theta_1 + ... + \phi_n(i)\theta_n = \phi^T(i)\theta \tag{1.2}$$

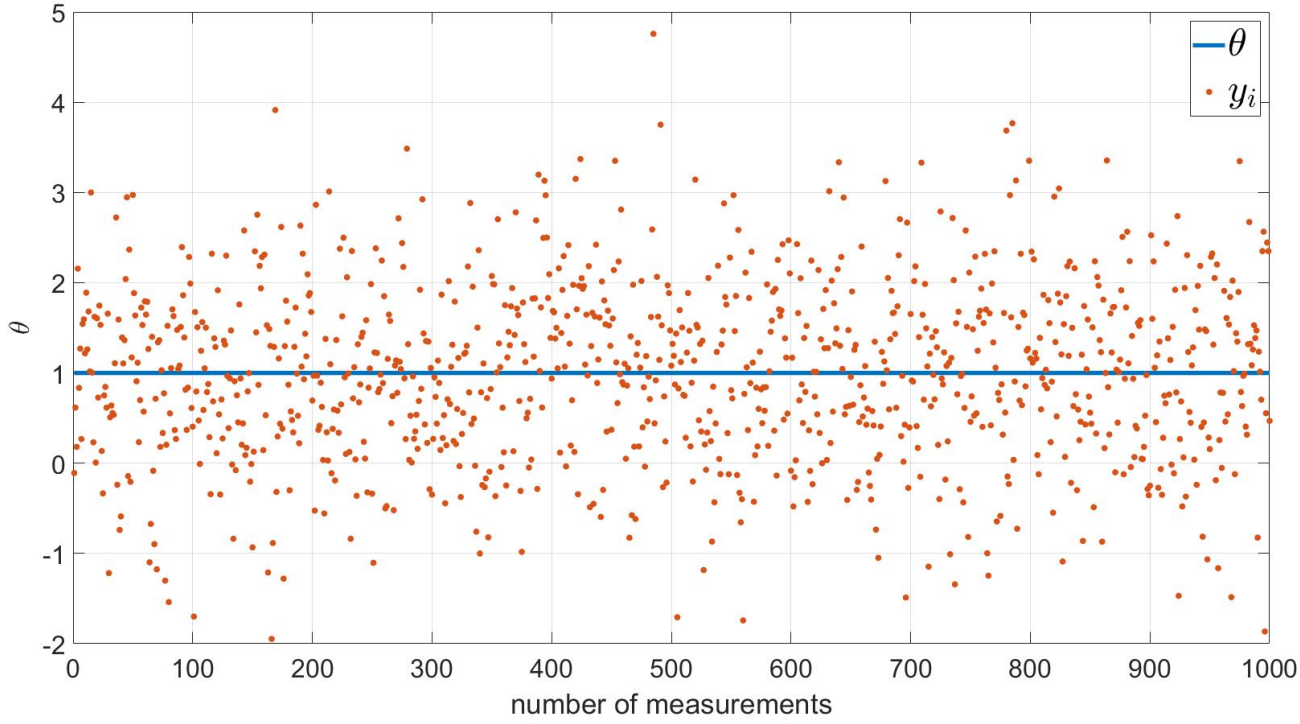

Figure 1: Real parameter θ and measurements y_i.

where y is the observed variable, $\theta_1, ..., \theta_n$ are parameters of the model to be determined, and $\phi_1, ..., \phi_n$ are known functions. The variable i often denotes time. We introduce the vectors

$$\phi^T(i) = (\phi_1(i) \ ... \ \phi_n(i))$$

$$\theta = (\theta_1 \ ... \ \theta_n)^T$$

The problem is to determine the parameters in such a way that the outputs computed from the model in Eq. (1.2) agree as closely as possible with the measured variables $y(i)$ in the sense of least squares. Therefore, the parameter θ should be chosen to minimize the least-squares loss function

$$V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t}(y(i) - \phi^T(i)\theta)^2$$

We introduce the vectors

$$Y(t) = (y(1) \ ... \ y(t))^T$$

$$\Phi = \begin{pmatrix} \phi^T(1) \\ \vdots \\ \phi^T(t) \end{pmatrix}$$

The solution to the least-squares problem is given by the following theorem: if the matrix $\Phi^T\Phi$ is nonsingular, the minimum is unique and given by

$$\hat{\theta}_t = (\Phi^T\Phi)^{-1}\Phi^T Y$$

Now that the method of minimum squares has been introduced, it remains to be seen how to apply it recursively. We introduce two new vectors

$$Z_t = \Phi^T \Phi = \sum_{i=1}^{t} \phi_i \phi_i^T$$

$$z_t = \Phi^T Y = \sum_{i=1}^{t} \phi_i y_i$$

Using the theorem one can say that

$$\hat{\theta}_t = (Z_t)^{-1} z_t \tag{1.3}$$

but also

$$\hat{\theta}_{t+1} = (Z_{t+1})^{-1} z_{t+1} \tag{1.4}$$

In particular

$$Z_{t+1} = \sum_{i=1}^{t+1} \phi_i \phi_i^T = \sum_{i=1}^{t} \phi_i \phi_i^T + \phi_{t+i} \phi_{t+i}^T = Z_t + \phi_{t+i} \phi_{t+i}^T$$

$$z_{t+1} = \sum_{i=1}^{t+1} \phi_i y_i = \sum_{i=1}^{t} \phi_i y_i + \phi_{t+i} y_{t+i} = z_t + \phi_{t+i} y_{t+i} \tag{1.5}$$

Now consider equation (1.3) and multiply both members by Z_t. The result is

$$Z_t \hat{\theta}_t = z_t \tag{1.6}$$

From equation (1.4), (1.5) and (1.6) it is possible to write

$$\begin{aligned} \hat{\theta}_{t+1} &= (Z_{t+1})^{-1}(z_t + \phi_{t+i} y_{t+i}) = (Z_{t+1})^{-1}(Z_t \hat{\theta}_t + \phi_{t+i} y_{t+i}) \\ &= (Z_{t+1})^{-1} Z_t \hat{\theta}_t + (Z_{t+1})^{-1} \phi_{t+i} y_{t+i} \end{aligned} \tag{1.7}$$

Equation (1.7) has a significant importance: Since all Z terms depend only on known functions, it is clear that the new estimate $\hat{\theta}_{t+1}$ depends only on the previous estimate $\hat{\theta}_t$ and the new measure y_{t+1}. In this way, while maintaining the correct result, a recursive least squares estimate has been made, because all the terms of the previous calculations do not appear.

1.1 Recursive least squares (known noise variance)

These results will be more useful in the second part of the laboratory. Since in the first part you only have to estimate one parameter and each measurement has the same weight, remembering the eq. (1.2) we can write:

$$\phi^T(i) = (\phi_1(i)) = 1$$

$$\theta = \theta_1$$

$$y(i) = \phi_1(i)\theta_1 = \theta_1 = \theta$$

Equation (1.1) is found, except for the term related to noise, which has been added specifically to simulate a series of real measurements. We can calculate Z_t and Z_{t+1} for the case considered, which are scalar and equal to

$$Z_t = \sum_{i=1}^{t} \phi_i \phi_i^T = \sum_{i=1}^{t} 1 = t$$

$$Z_{t+1} = \sum_{i=1}^{t+1} \phi_i \phi_i^T = \sum_{i=1}^{t+1} 1 = t+1$$

The estimate for the next step is taken from Eq. 1.7

$$\hat{\theta}_{t+1} = \frac{t}{t+1}\hat{\theta}_t + \frac{1}{t+1} y_{t+i}$$

So for the current estimate it is sufficient to calculate

$$\hat{\theta}_t = \frac{t-1}{t}\hat{\theta}_{t-1} + \frac{1}{t} y_t$$

The variance σ_t^θ of the estimate remains to be calculated. Using the properties of independent and identically distributed random variables, it can be demonstrated that

$$\sigma_t^\theta = \frac{\sigma}{\sqrt{t}}$$

where σ is the variance of noise.

Remembering that $\theta = 1$, we report an example of the trend of $\hat{\theta}$ and the $3-\sigma$ (99%) confidence bounds (Figure 2).

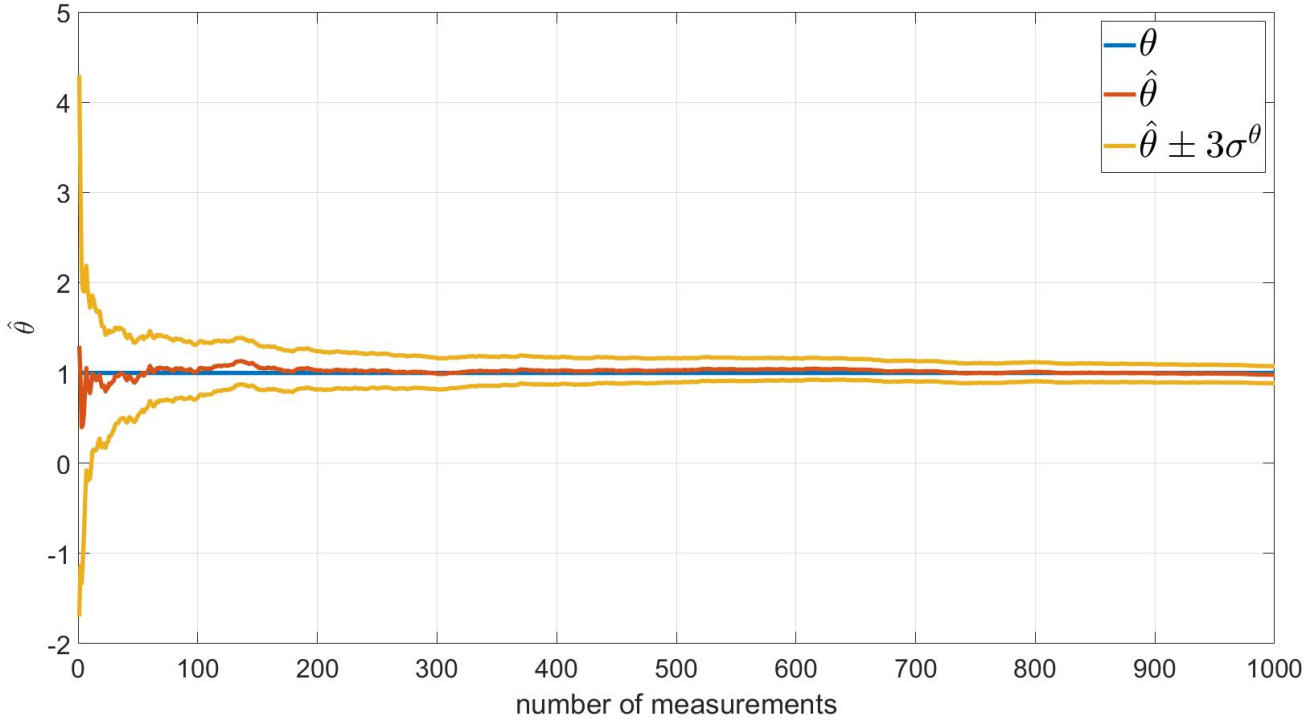

Figure 2: Recursive least squares (with known noise variance).

As the number of measurements increases, $\hat{\theta}$ gets closer and closer to the real value θ. In fact $\hat{\theta} = 0.991$.

It is also clear that θ always falls within the confidence bounds.

1.2 Recursive least squares (unknown noise variance)

If the noise variance is not given, it is possible to estimate it through the definition of the noise variance itself.

$$\hat{\sigma}_t^2 = \frac{1}{t}\sum_{i=1}^{t}(y_i - \hat{y}_i^t)^2 = \frac{1}{t}\sum_{i=1}^{t}(y_i - \hat{\theta}_t)^2$$

From which we obtain

$$\hat{\sigma}_t^\theta = \frac{\hat{\sigma}_t}{\sqrt{t}}$$

We can plot the graph using the estimate of the variance just obtained (Figure 3).

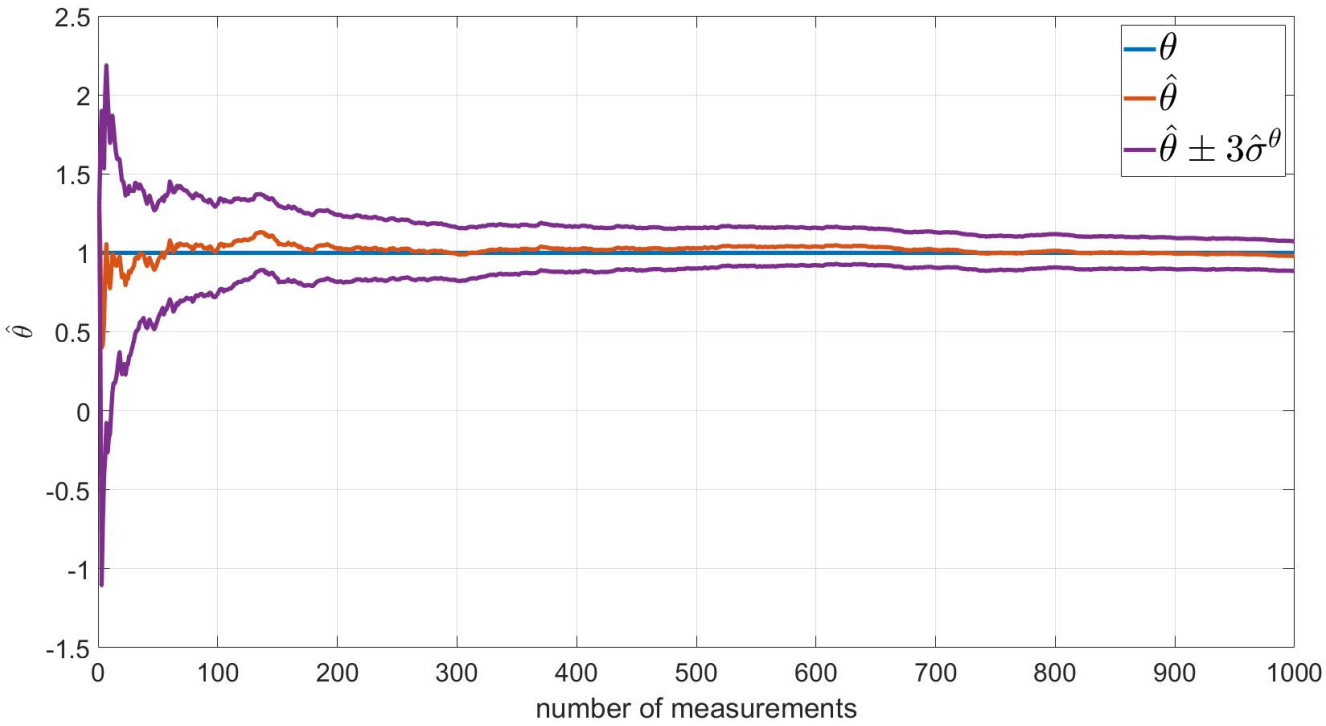

Figure 3: Recursive least squares (with unknown noise variance).

We compare the results obtained in Figure 4.

As for $\hat{\theta}$, also the estimate of the variance $\hat{\sigma}_t^\theta$ tends asymptotically to the value of the real variance σ_t^θ, and in any case θ falls within the real and estimated confidence bounds.

Finally we consider the relative estimation error (Fiugre 5), comparing the cases in which sigma is known

$$r_n = \frac{\sigma^\theta}{\hat{\theta}}$$

or unknown

$$\hat{r}_n = \frac{\hat{\sigma}^\theta}{\hat{\theta}}$$

.

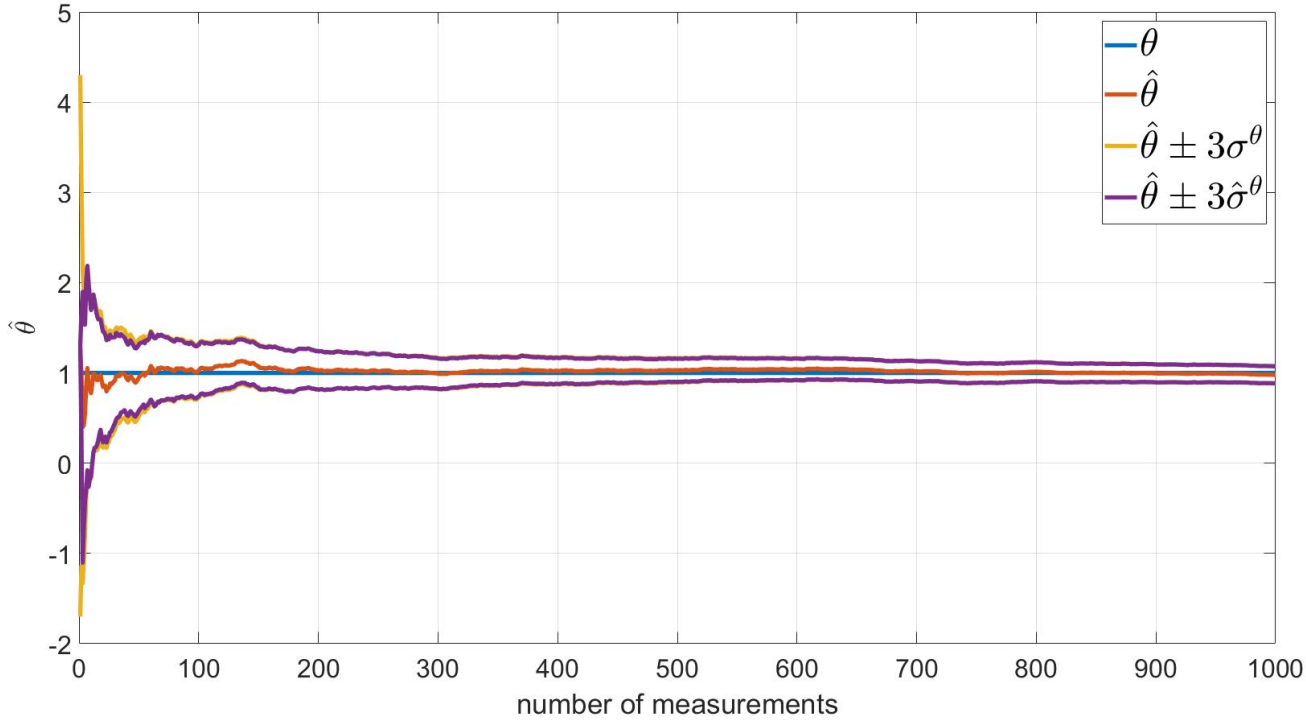

Figure 4: Recursive least squares (with known and unknown noise variance).

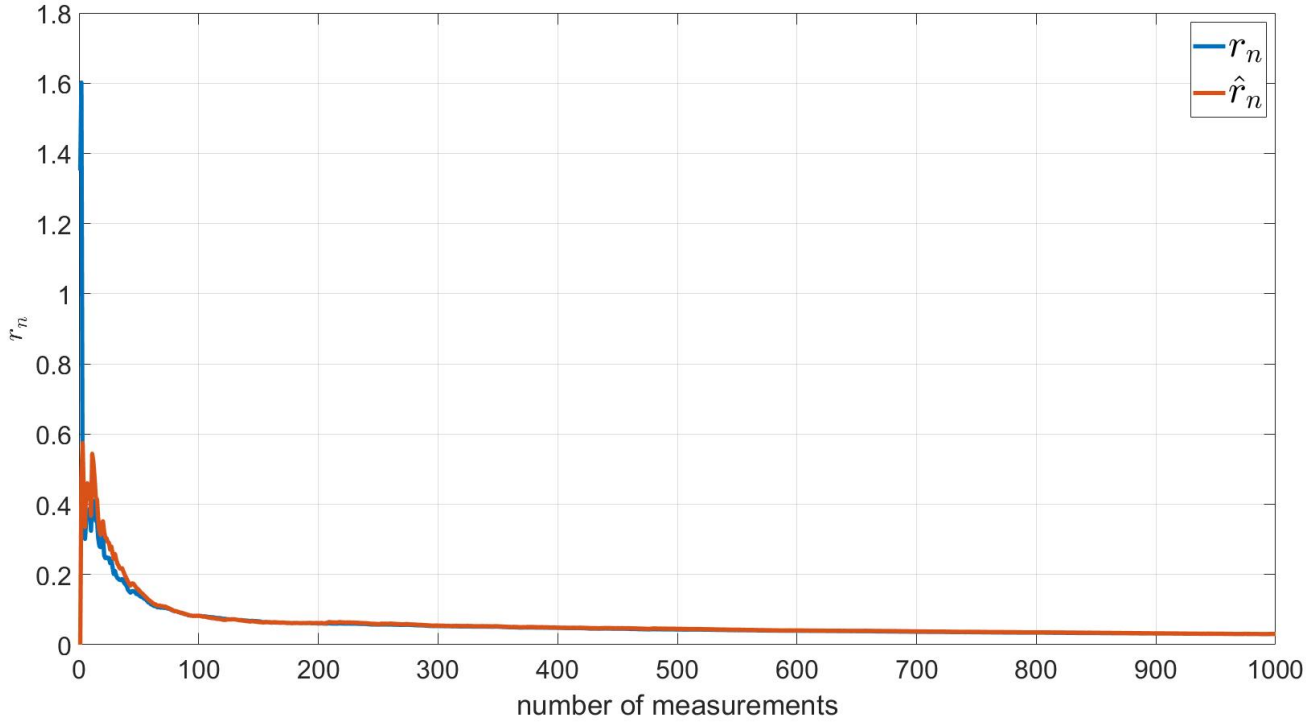

Figure 5: Relative estimation error (with known and unknown noise variance).

In both cases, after 100 measurements the relative error is is about 10% while after 1000 measurements the relative error is is about 3%.

1.3 Recursive least squares with forgetting factor

Now consider the theta parameter no longer constant. In this particular case θ is equal to

$$\theta_i = \begin{cases} 1 & \text{for } i = 1, ..., t \\ 2 & \text{for } i = t+1, ..., 2t \end{cases}$$

The objective is to realize a recursive least squares with forgetting factor $\lambda = \{1, 0.99, 0.95, 0.9, 0.8\}$ ($\lambda \in [0, 1]$) and compare the estimate with the true value θ. The new function to be minimized is

$$V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t}(y_i - \phi_i^T\theta)^2\lambda^{t-i}$$

We introduce the diagonal matrix Λ

$$\Lambda = \begin{bmatrix} \lambda^{t-1} & & & & \\ & \lambda^{t-2} & & & \\ & & \ddots & & \\ & & & \lambda & \\ & & & & 1 \end{bmatrix}$$

It can be demonstrated that the solution to the loss function is

$$\hat{\theta}_t = (\Phi^T\Lambda\Phi)^{-1}\Phi^T\Lambda Y$$

So we can write

$$Z_t = \Phi^T\Lambda\Phi = \sum_{i=1}^{t}\lambda^{t-i}\phi_i\phi_i^T$$

$$= \phi_t\phi_t^T + \sum_{i=1}^{t-1}\lambda^{t-i}\phi_i\phi_i^T = \phi_t\phi_t^T + \lambda\sum_{i=1}^{t-1}\lambda^{t-i-1}\phi_i\phi_i^T$$

$$= \phi_t\phi_t^T + \lambda Z_{t-1}$$

$$z_t = \Phi^T\Lambda Y = \sum_{i=1}^{t}\lambda^{t-i}\phi_i y_i$$

$$= \phi_t y_t + \lambda z_{t-1}$$

Starting from $Z_0 = 0$ and $z_0 = 0$ and remembering that in this case $\phi_i = 1$. we can simply write

$$Z_t = 1 + \lambda Z_{t-1}$$

$$z_t = 1 + \lambda z_{t-1}$$

$$\hat{\theta}_t = z_t/Z_t$$

The following is the trend of theta with the variation of lambda.

The smaller the lambda, the faster the system forgets. So for lambda close to 1 the system tends to remember the previous estimate then it is slower to converge to the new value. On the other hand, for

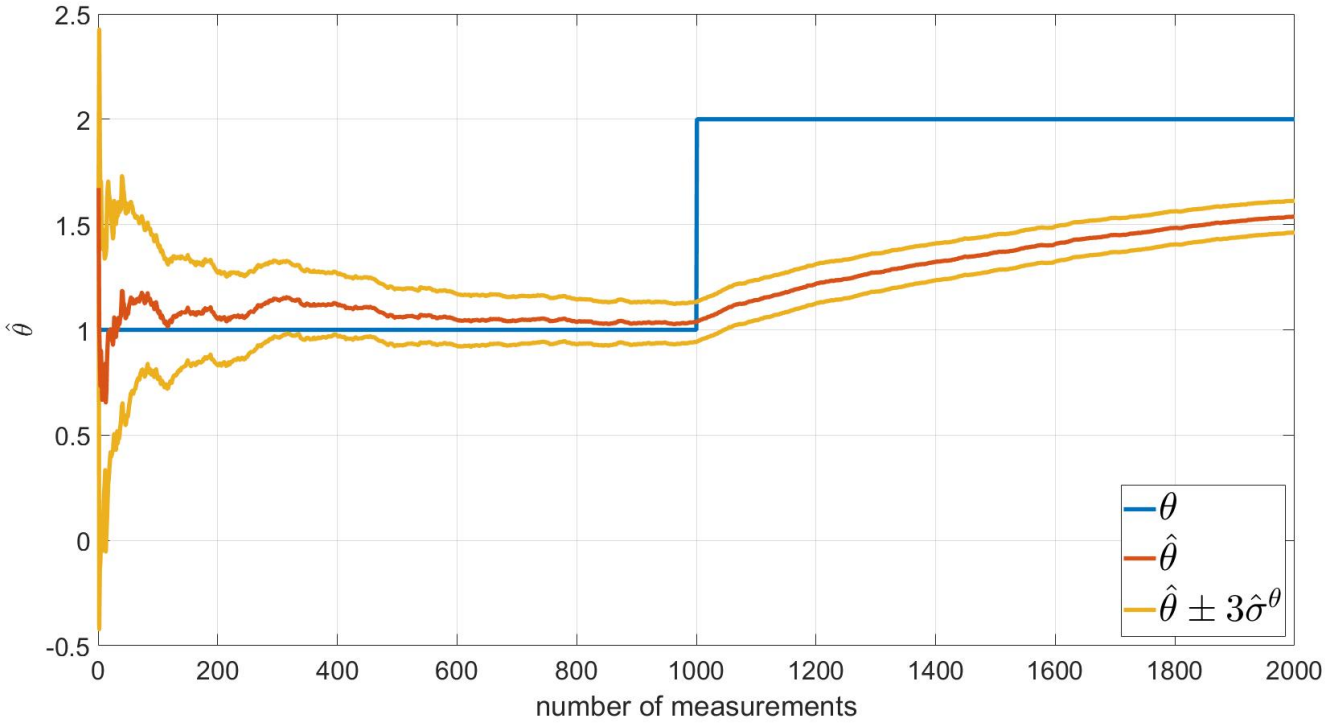

Figure 6: Recursive least squares with forgetting factor $\lambda = 1$.

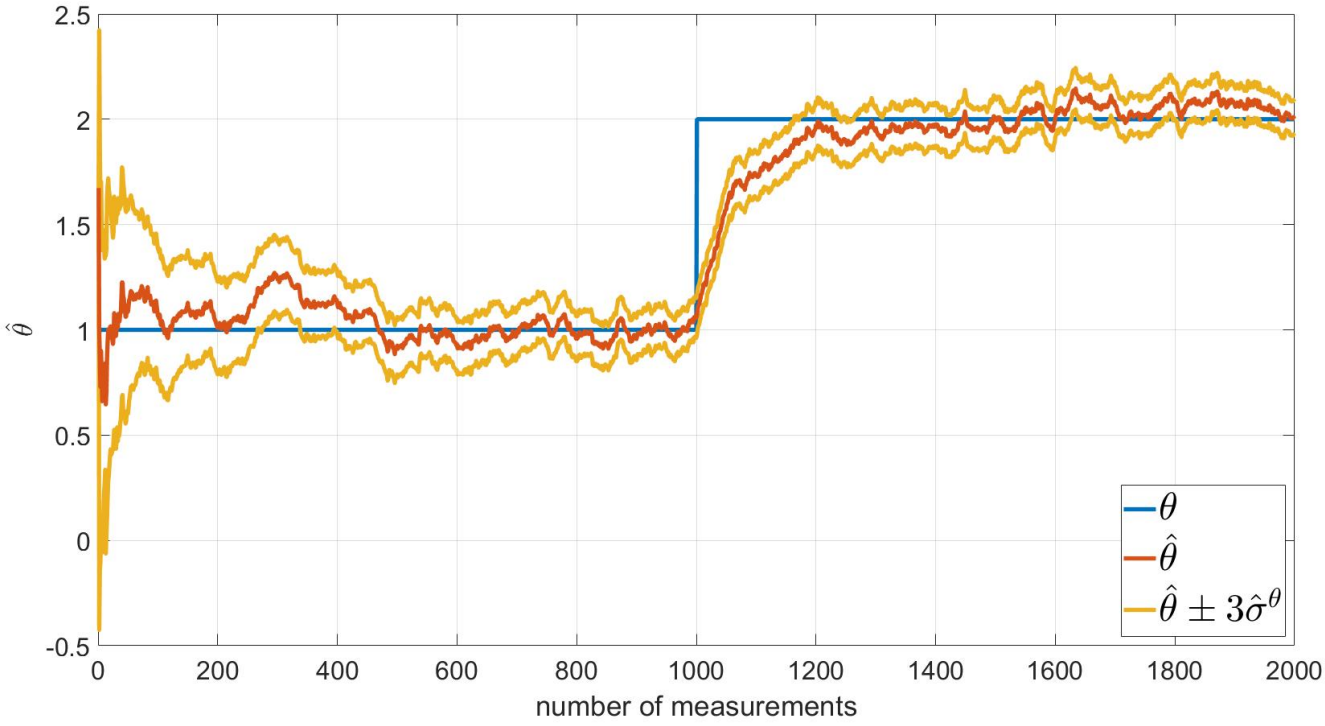

Figure 7: Recursive least squares with forgetting factor $\lambda = 0.99$.

small lambda, the estimate follows more faithfully the new measurements and quickly forgets the previous estimates, but is strongly subject to noise.

Therefore it is not possible to say beforehand which is the right λ for the best solution but it depends on the needs of the project. In any case, the solution with $\lambda = 1$ can be excluded, as it is too slow. We

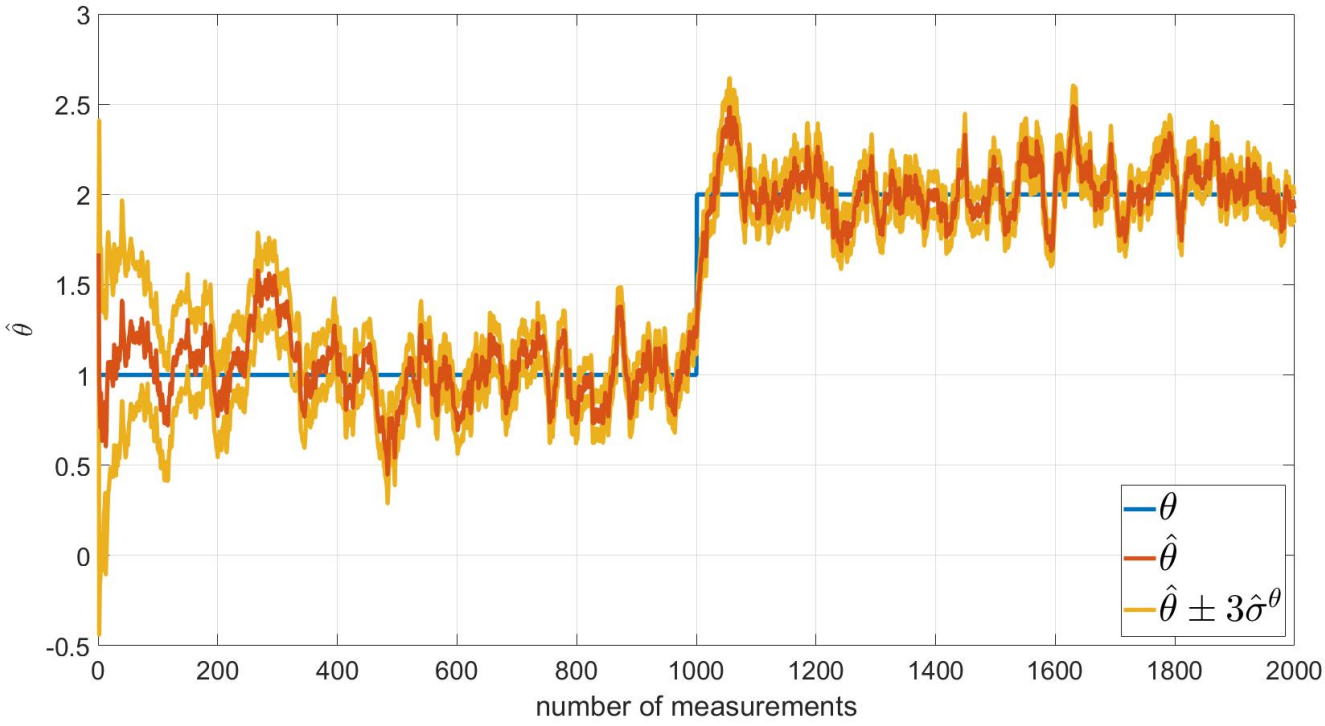

Figure 8: Recursive least squares with forgetting factor $\lambda = 0.95$.

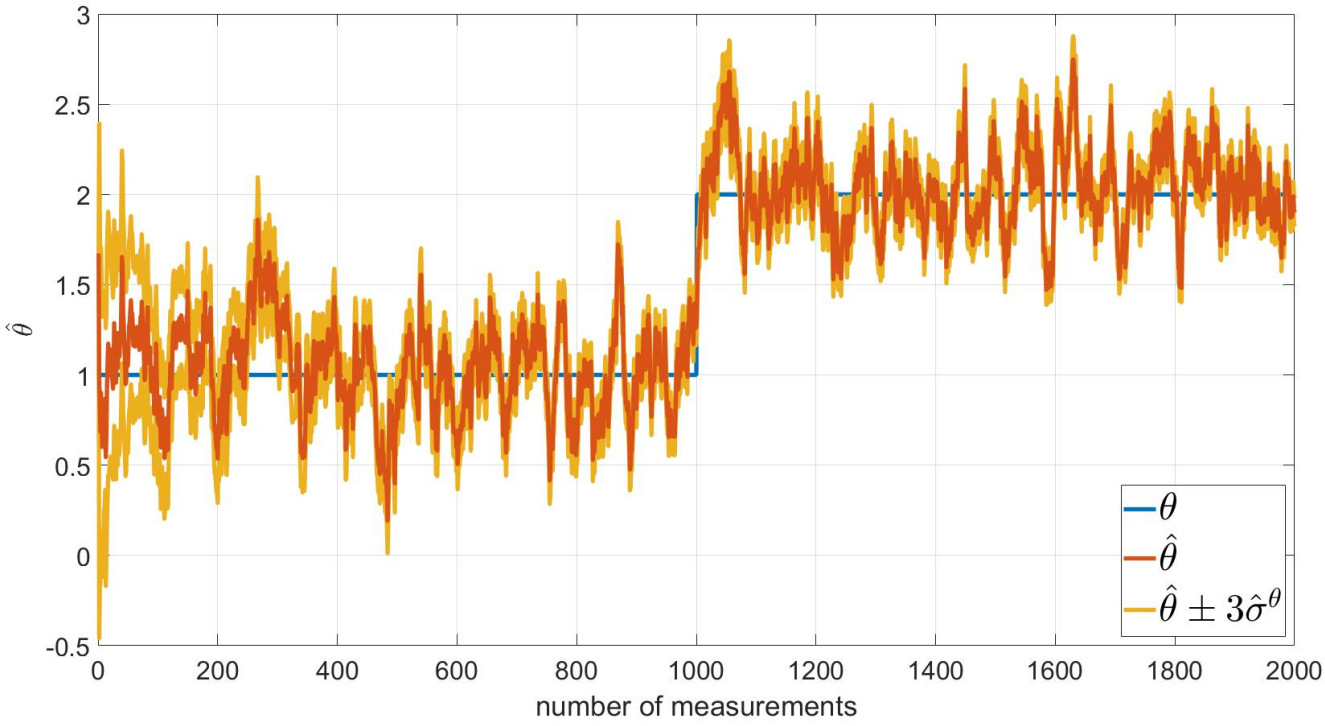

Figure 9: Recursive least squares with forgetting factor $\lambda = 0.9$.

can also exclude solutions with $\lambda = 0.9$ and $\lambda = 0.8$ because they have too much noise.

Hence the final choice is a compromise. For example, we can choose $\lambda = 0.98$. As we see in Figure 11, θ does not always fit into the confidence bounds, but it can be a good compromise with the speed of estimation $\hat{\theta}$.

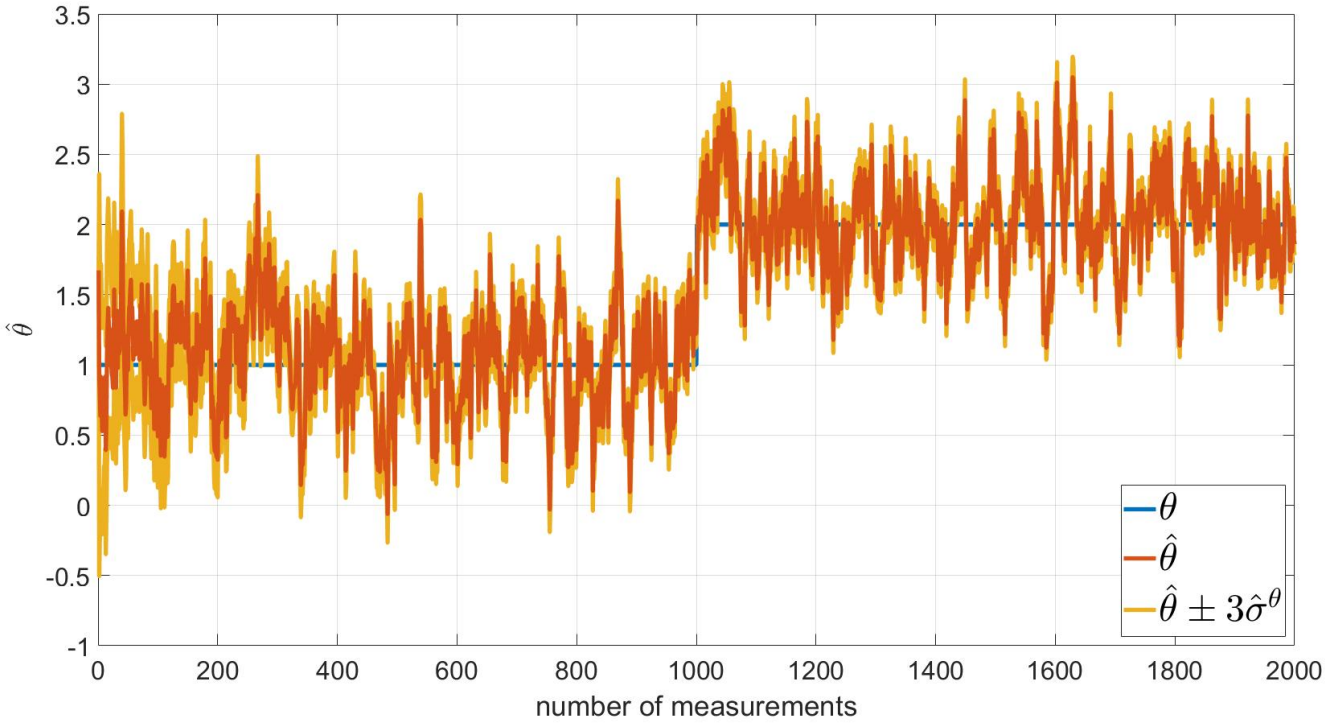

Figure 10: Recursive least squares with forgetting factor $\lambda = 0.8$.

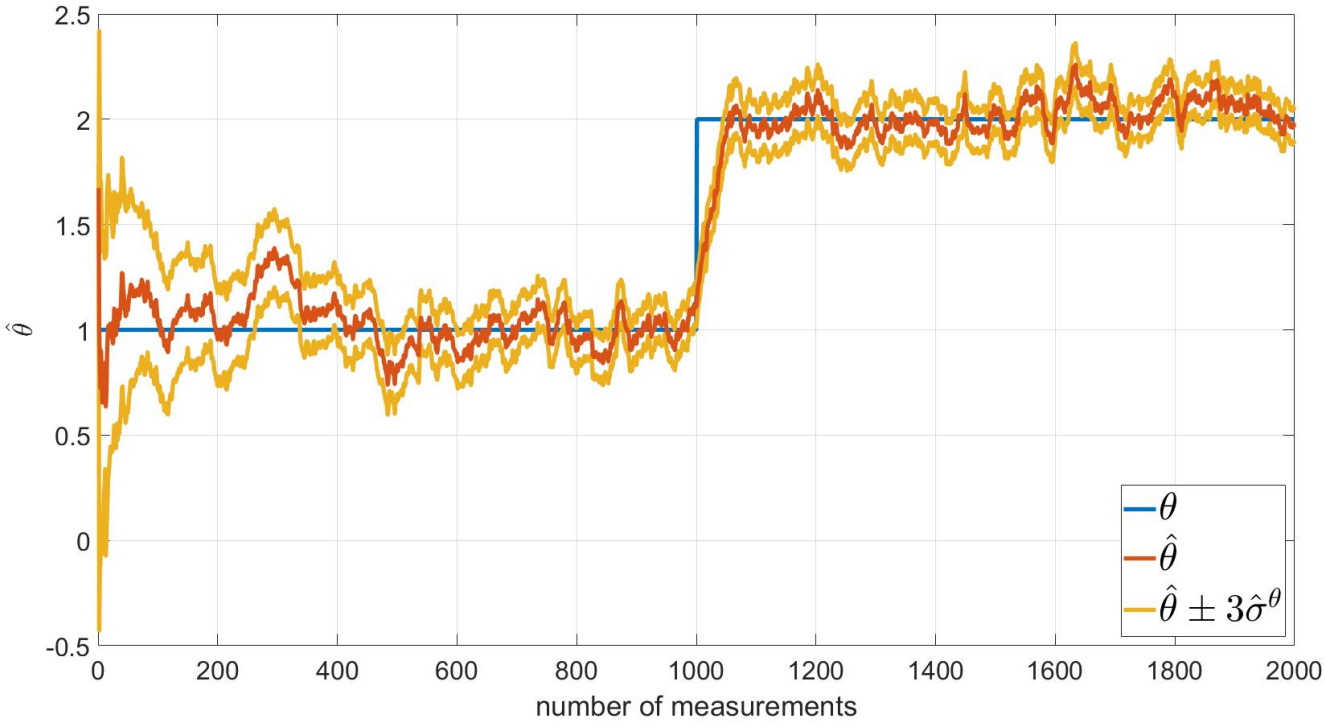

Figure 11: Recursive least squares with forgetting factor $\lambda = 0.98$.

2 Model selection with cross validation

The objective is to find a model that best approximates the real one, in particular by acting on the level of complexity of the model, using measurements subject to noise. We will use some data for training,

while the remaining data will be used for validation, in order to avoid overfitting. Again, everything was simulated on matlab.

Let's consider the real unknown model

$$y = \sin 4x + \frac{1}{4x+1}$$

The available data are:

$$y_i = \sin 4x_i + \frac{1}{4x_i+1} v_i, \quad i = 1, ..., t$$

where $t = 100$, $x_i = \frac{i-1}{99}$ and $v_i \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 0.2$ is i.i.d. Gaussian Noise whose variance is unknown.

We will consider the following polynomial estimator

$$\hat{y} = \theta_0 + \theta_1 x ... + \theta_m x^m = \phi^T(x)\theta$$

where $\theta = (\theta_0 + \theta_1 ... + \theta_m)$ is the vector of parameters to estimate, $\phi(x) = (1, x, ..., x^m)$ is the vector of known functions and m represents the model complexity, which will be the objective of this part.

Of all available data, 80% will be used for training and the remaining 20% will be used for validation. Why? We could easily think that a model with many parameters is certainly better than a model with few parameters. However, a model with too many parameters leads to an opposite effect to the desired one, i.e. the model stops learning and adapts to the available data. This is known as overfitting. For this reason, validation data is necessary: this data will only be used as a test and not for the training phase. Here is an example of the real model and the training and validation data (Figure 12)

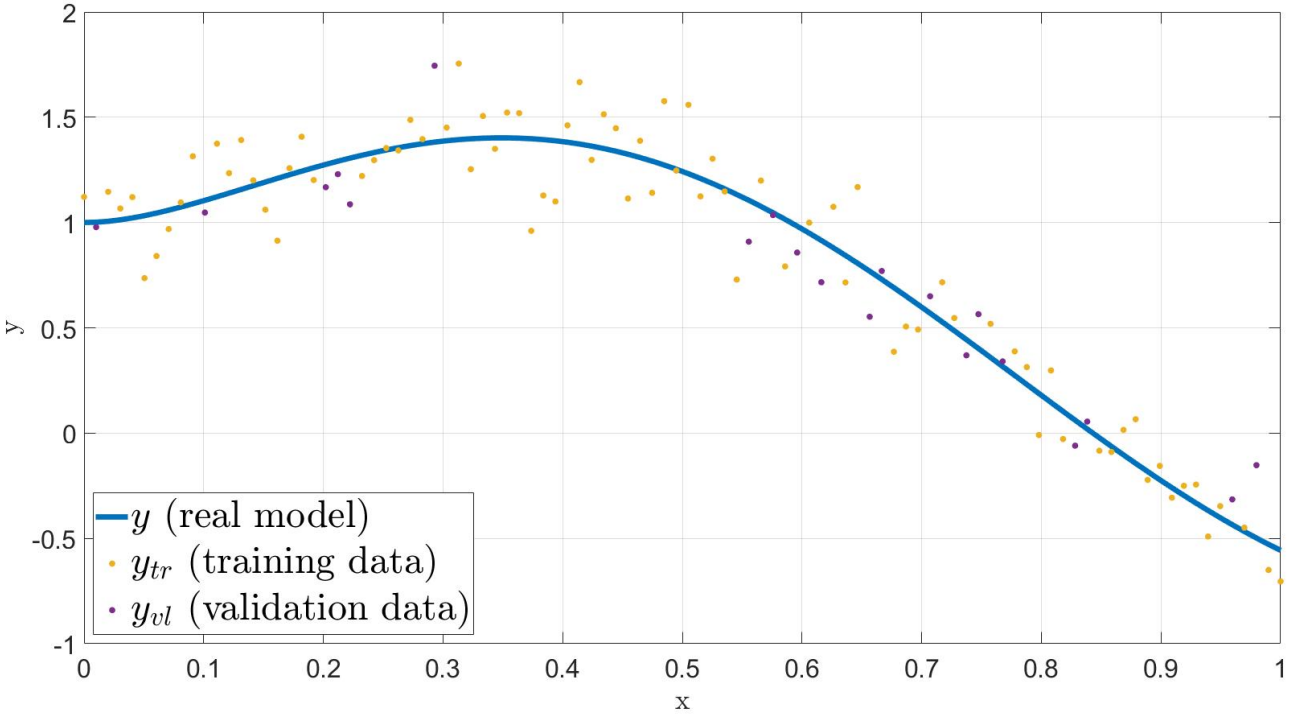

Figure 12: Real model, training and validation data set.

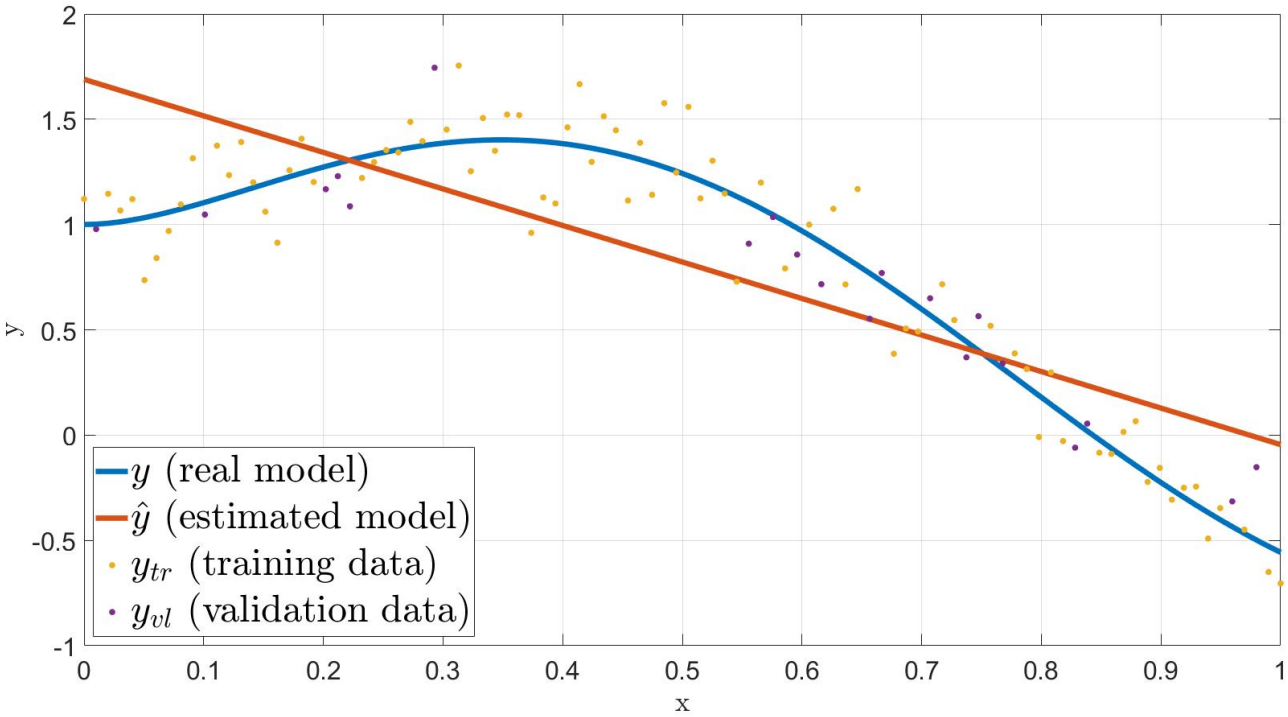

Figure 13: Estimated model for $m = 1$.

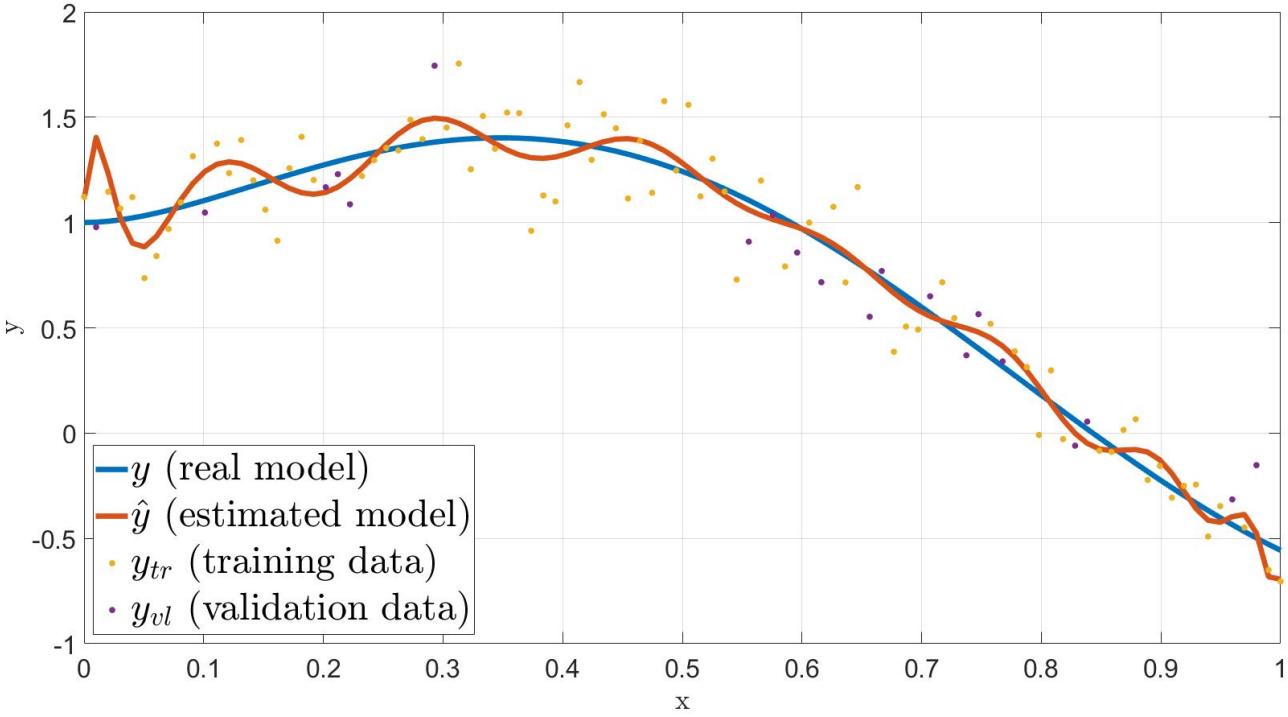

Figure 14: Estimated model for $m = 20$.

To calculate $\hat{\theta}$ we use the least squares estimation again. For convenience, the solving equations are called up

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T y_{tr}$$

$$\hat{y} = \Phi\hat{\theta}$$

Two cases are reported for $m = 1$ (Figure 13) and $m = 20$ (Figure 14).

Now we have to choose which one is the best model. How can we do that? We have to calculate the average variance of the training and validation data. The model with the lowest validation variance will be the model to choose.

Through the average of 1000 simulations we have obtained the following graph (Figure 15):

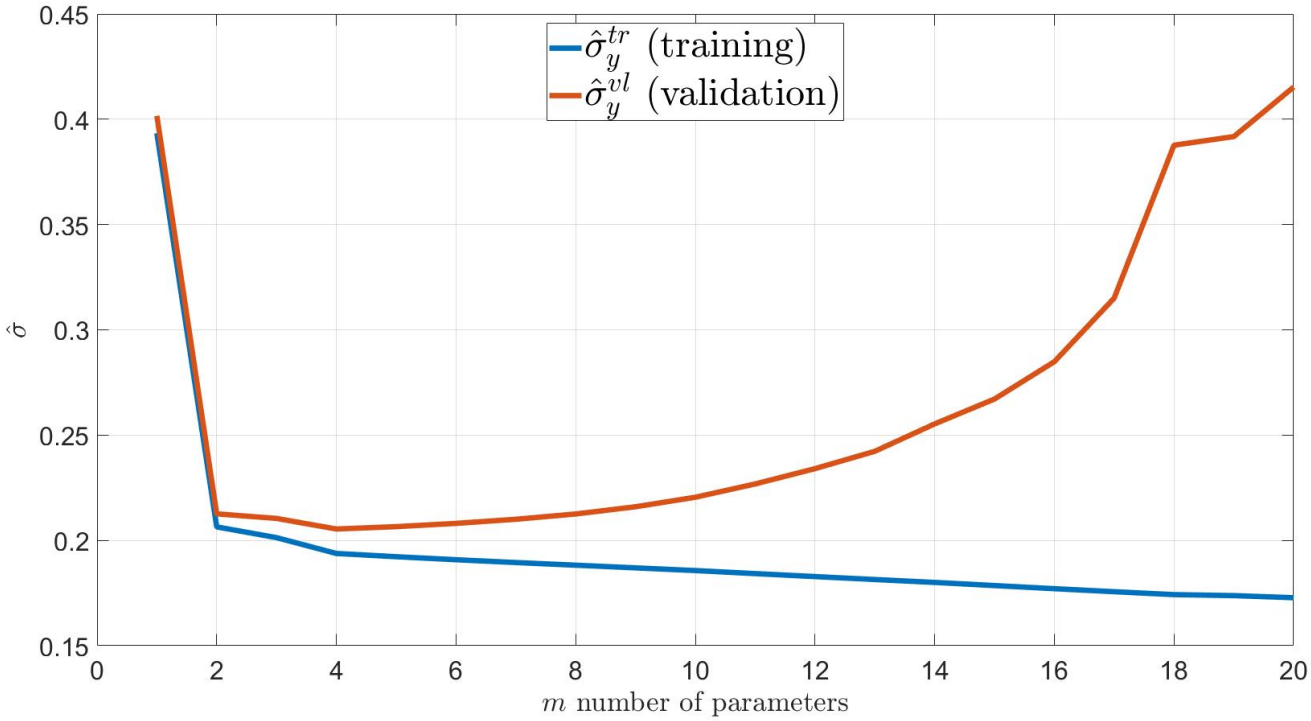

Figure 15: Average prediction error on training and validation set.

As announced, we can see how the minimum variance is obtained for $m = 4$, while for $m > 4$ overfitting occurs.

Finally we can plot the estimated model with $m = 4$ with the confidence bound and compare it with the real model (Figure 16).

We see that the estimated model is very similar to the real one, which always remains within the confidence margin.

3 Identification of discrete time linear systems

In this last part we will estimate the damping coefficient of a DC motor. Then we will design a PD position controller.

Consider the current-controller model of a DC motor given by

$$Y(s) = \frac{K_i K_t}{s(Js + b)} U(s) = P(s)U(s)$$

where K_i, K_t, J are known but the damping coefficient b is unknown. If the previous system is discretized with sampling period T, the exact discretization is

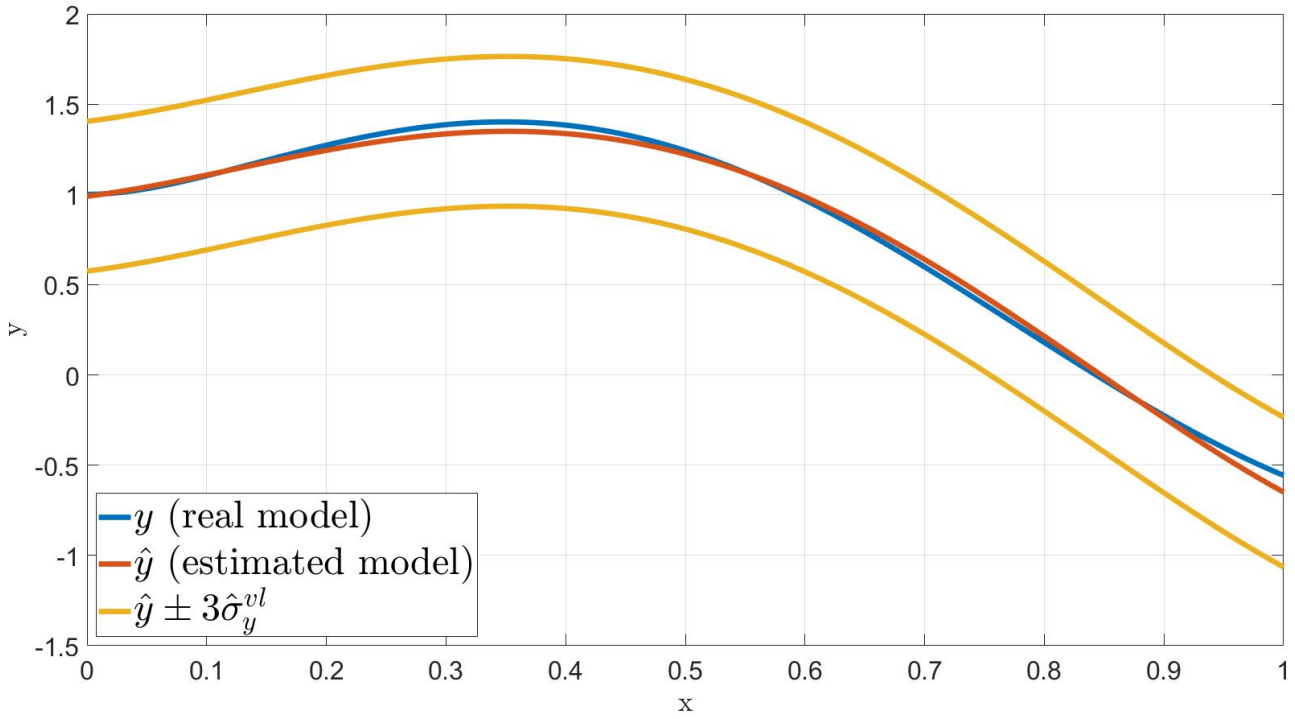

Figure 16: Comparison between real model and estimated model.

$$Y(z) = \frac{b_1 z + b_0}{(z-1)(z+a_0)} U(z)$$

where

$$\tau = \frac{J}{b}, \quad b_0 = \frac{K_i K_t}{b}(\tau - (T+\tau)e^{-\frac{T}{\tau}}), \quad b_1 = \frac{K_i K_t}{b}((T-\tau) + \tau e^{-\frac{T}{\tau}}), \quad a_0 = -e^{-\frac{T}{\tau}}$$

Consider the following filtered input-output signals

$$U_f(z) = \frac{1}{z} U(z), \quad Y_f(z) = \frac{z-1}{z} Y(z)$$

and note that

$$Y_f(z) = \frac{b_1 z + b_0}{z + a_0} U_f(z) = \frac{b_1 + b_0 z^{-1}}{1 + a_0 z^{-1}} U_f(z)$$

which can be equivalently be written in the time-domain as

$$y_f = -a_0 y_f(t-1) + b_1 u_f(t) + b_0 u_f(t-1) = \underbrace{[-y_f(t-1)u(t)u(t-1)]}_{\phi(t)^T} \underbrace{[a_0 b_1 b_0]^T}_{\theta}$$

3.1 Parameter identification

For the estimation of the damping coefficient b we have to use a recursive least squares method. We introduce the new matrices

$$P = Z^{-1}$$

$$K_t = P_t \phi_t$$

Consider equations 1.5 and 1.6. By replacing Z with P we can write:

$$\begin{aligned}
z_{t-1} &= Z_{t-1}\hat{\theta}_{t-1} \\
\hat{\theta}_t = Z_t^{-1} z_t &= (Z_{t-1} + \phi_t \phi_t^T)^{-1}(z_{t-1} + \phi_t y_t) = (P_{t-1}^{-1} + \phi_t \phi_t^T)^{-1}(P_{t-1}^{-1}\hat{\theta}_{t-1} + \phi_t y_t) \\
&= (P_{t-1}^{-1} + \phi_t \phi_t^T)^{-1}(P_{t-1}^{-1}\hat{\theta}_{t-1} + \phi_t y_t + \phi_t \phi_t^T \hat{\theta}_{t-1} - \phi_t \phi_t^T \hat{\theta}_{t-1}) \\
&= \hat{\theta}_{t-1} + (P_{t-1}^{-1} + \phi_t \phi_t^T)^{-1}\phi_t(y_t - \phi_t^T \hat{\theta}_{t-1}) \\
&= \hat{\theta}_{t-1} + P_t \phi_t (y_t - \phi_t^T \hat{\theta}_{t-1}) = \hat{\theta}_{t-1} + K_t(y_t - \phi_t^T \hat{\theta}_{t-1}) \\
&= \hat{\theta}_{t-1} + K_t \epsilon_t
\end{aligned} \tag{3.1}$$

The residual ϵ_t can be interpreted as the error in predicting the signal y_t one step ahead based on the estimate $\hat{\theta}_{t-1}$. The equation may seem easy to solve. In fact, for the calculation of P, it is necessary to invert the matrix

$$P_t = Z_t^{-1} = (Z_{t-1} + \phi_t \phi_t^T)^{-1} = (P_{t-1}^{-1} + \phi_t \phi_t^T)^{-1}$$

The following matric inversion lemma is useful: let A, C, and $C^{-1} + DA^{-1}B$ be nonsingular square matrices. Then $A + BCD$ is invertible, and

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Replacing

$$A = P_{t-1}^{-1}$$

$$B = \phi_t$$

$$C = 1$$

$$D = \phi_t^T$$

we can write

$$\begin{aligned}
P_t &= (P_{t-1} + \phi_t \phi_t^T)^{-1} = P_{t-1} - P_{t-1}\phi_t(1 + \phi_t^T P_{t-1}\phi_t)^{-1}\phi_t^T P_{t-1} \\
&= \left(I - \frac{P_{t-1}\phi_t \phi_t^T}{1 + \phi_t^T P_{t-1}\phi_t}\right) P_{t-1}^{-1} \\
&= (I - K_t \phi_t^T) P_{t-1}^{-1}
\end{aligned} \tag{3.2}$$

$$K_t = P_t \phi_t = P_{t-1} - P_{t-1}\phi_t(1 + \phi_t^T P_{t-1}\phi_t)^{-1}\phi_t^T P_{t-1}\phi_t = \frac{P_{t-1}\phi_t}{1 + \phi_t^T P_{t-1}\phi_t}$$

Now we have all the equations to proceed with the estimate. However, one final point needs to be made: the matrix P_t is defined only when the matrix $\Phi_t^T \Phi_t$ is nonsingular. Since

$$\Phi_t^T \Phi_t = \sum_{i=1}^{t} \phi_i \phi_i^T$$

it follows that $\Phi_t^T \Phi_t$ is always singular if $t < n$ (n is the number of parameters). To obtain an initial

condition for P, it is necessary to choose $t = t_0$ so that $\Phi_{t_0}^T \Phi_{t_0}$ is nonsingular.

The initial conditions are then

$$P_{t_0} = (\Phi_{t_0}^T \Phi_{t_0}^T)$$

$$\hat{\theta}_{t_0} = P_{t_0} \Phi_{t_0}^T Y_{t_0}$$

The recursive equations can be used for $t > t_0$. In this case $t_0 = 3$.

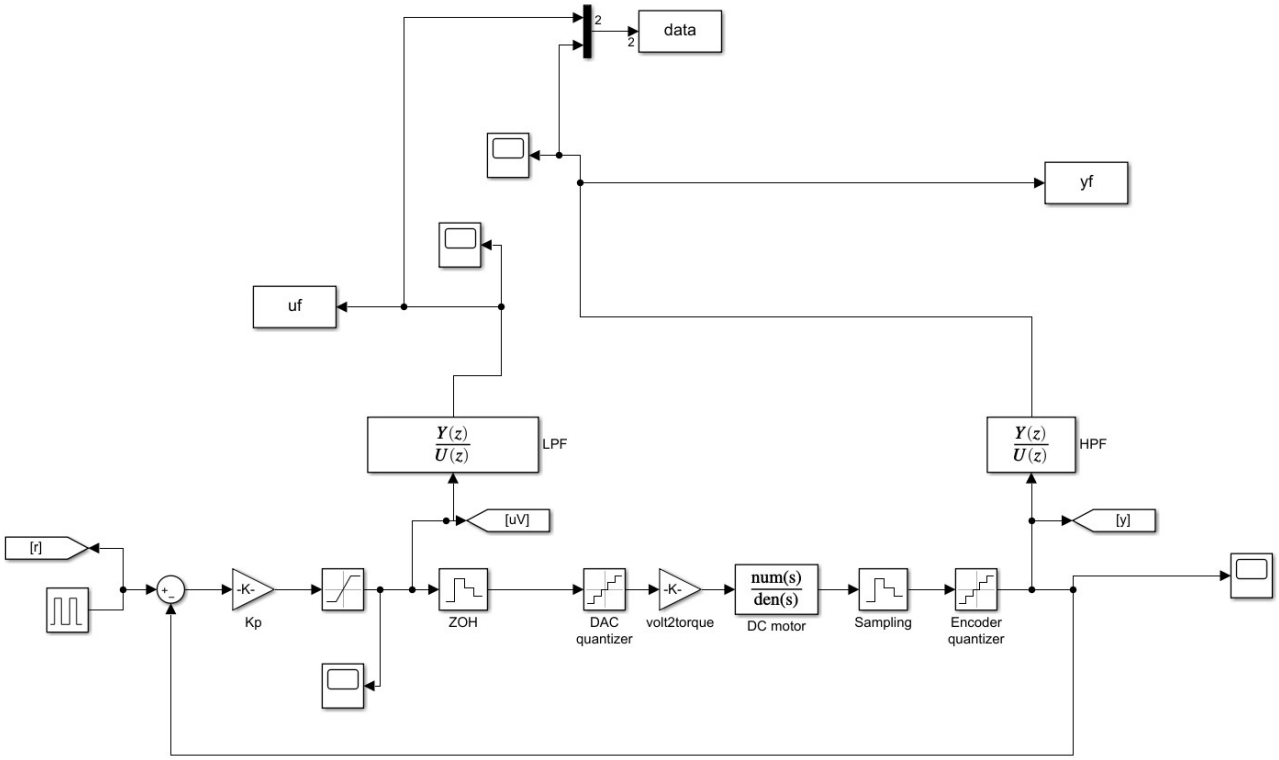

Figure 17: Simulink model.

The Simulink model in Figure 17 is provided for the project with the following parameters: $T = 0.001$, $J = 3.78 \times 10^5$, $b = 2.4299 \times 10^4$, $K_i = 2$, $K_t = 0.071$, where $K_{volt2torque} = K_i K_t$. The available data are $\{u_f(i), y_f(i)\}, \quad i = 1, ..., t$

Noting that $a_0 = -e^{-\frac{T}{\tau}}$ estimate the damping coefficient as

$$\hat{b}(t) = \frac{J}{T} \ln(-\hat{a}_0(t))$$

Recalling that

$$Var(\hat{\theta}(t)) \approx P(t)\sigma_y^2 = \left(\sum_{k=1}^{t} \phi(k)\phi^T(k) \right)^{-1} \sigma_y^2$$

and using as an estimate of the measurement noise variance

$$\hat{\sigma}_y^2(t) = \frac{1}{t} \sum_{k=1}^{t} (y_f(k) - \hat{y}_f(k;t))^2 = \frac{1}{t} \sum_{k=1}^{t} (y_f(k) - \phi^T(k)\hat{\theta}(t))^2$$

the following formula estimates the confidence bounds on the estimated parameter a_0 using the 1-1

element of the matrix $P_t\hat{\sigma}_y^2$, i.e.

$$\hat{\sigma}_{\hat{a}_0(t)} = \sqrt{P_{11}(t)\hat{\sigma}_y^2(t)}$$

Finally, the $3-\sigma$ confidence bounds for b can be calculated using the formula

$$\hat{b}_{bounds}(t) = \frac{J}{T}\ln(-\hat{a}_0(t) \pm 3\hat{\sigma}_{\hat{a}_0}(t))$$

Here is the plot of $\hat{b}$ and confidence bounds (Figure 18).

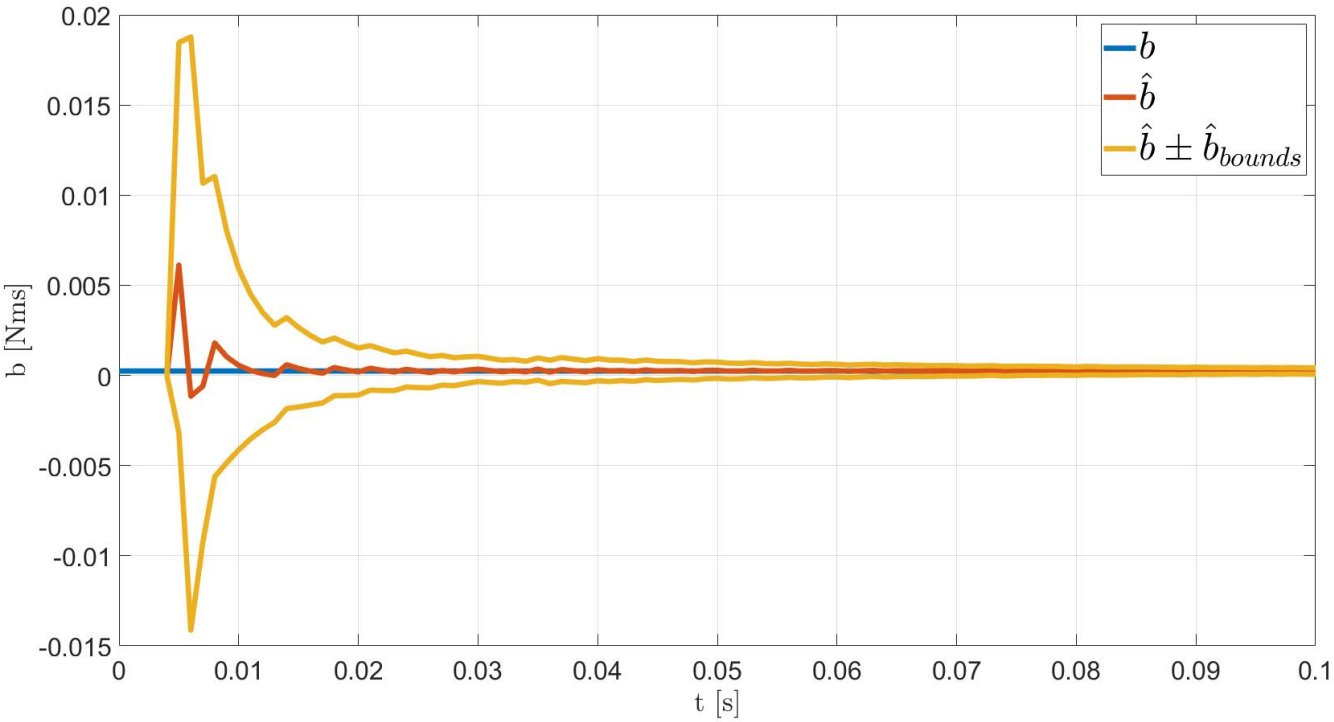

Figure 18: b and $\hat{b}$ with $3-\sigma$ confidence bounds.

The value found by the estimate is $\hat{b} = 2.4328 \times 10^4$, which is very close to the real value $b = 2.4299 \times 10^4$.

3.2 Identification-based Tuning for Control Design

The last part of this laboratory consists in designing a PD position controller

$$C(s) = K_P + K_D\frac{s}{1+\tau_c s}$$

for the continuous time transfer function $P(s)$ with maximum overshoot of 10% and settling time $t_s = 0.5s$. There are no specific requests on the steady state error

$$e = \frac{100}{1+P(0)C(0)}$$

which is automatically $e = 0$ because $P(0) = \infty$.

We indicate $G(s)$ as the open-loop control system and $W(s)$ as the closed-loop control system, i.e.

$$G(s) = P(s)C(s)$$

$$W(s) = \frac{G(s)}{1 + G(s)}$$

We indicate ω_c as the crossover frequency and ϕ_m as the phase margin of $G(s)$.

Through appropriate approximate conversion tables we can translate overshoot and settling time into phase margin and crossover frequency. In particular we obtain:

$$\phi_m = 60^\circ$$

$$\omega_c = 13.3 \ rad/s$$

To find the controller parameters we define a the magnitude and α the phase of $C(s)$. We can calculate a and α using the requirements in frequency domain:

$$|P(j\omega_c)||C(j\omega_c)| = 1 \quad \rightarrow \quad a = |C(j\omega_c)| = \frac{1}{|P(j\omega_c)|}$$

$$\angle G(j\omega_c) + 180^\circ = \phi_m \quad \rightarrow \quad \alpha = \phi_m - \phi_m^P$$

Finally, for a PD controller, the coefficients K_P and K_D are calculated as follows:

$$K_P = a\cos\alpha$$

$$K_D = \frac{a\sin\alpha}{\omega_c}$$

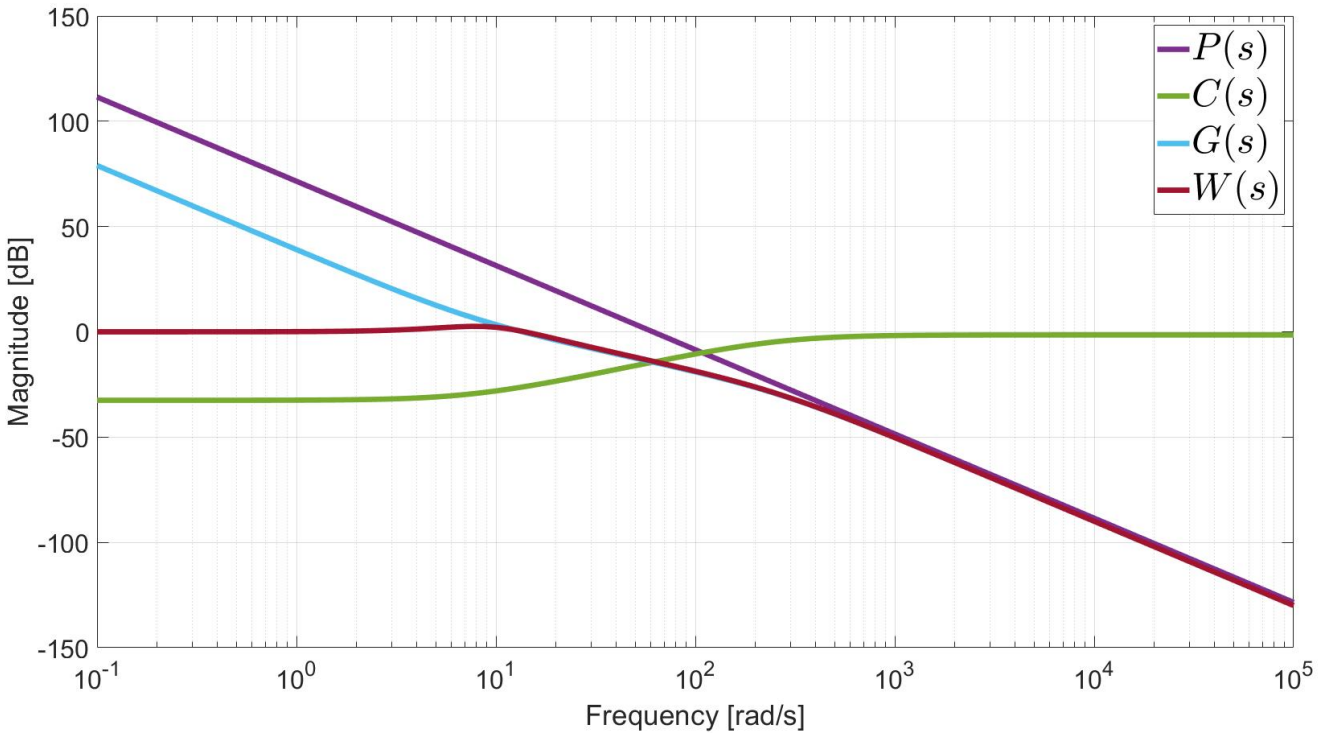

Figure 19: Bode magnitude plot of $P(s)$, $C(s)$, $G(s)$ and $W(s)$ with $b = 0$.

Now we have to consider a scenario where b is not known and a scenario after the identification of

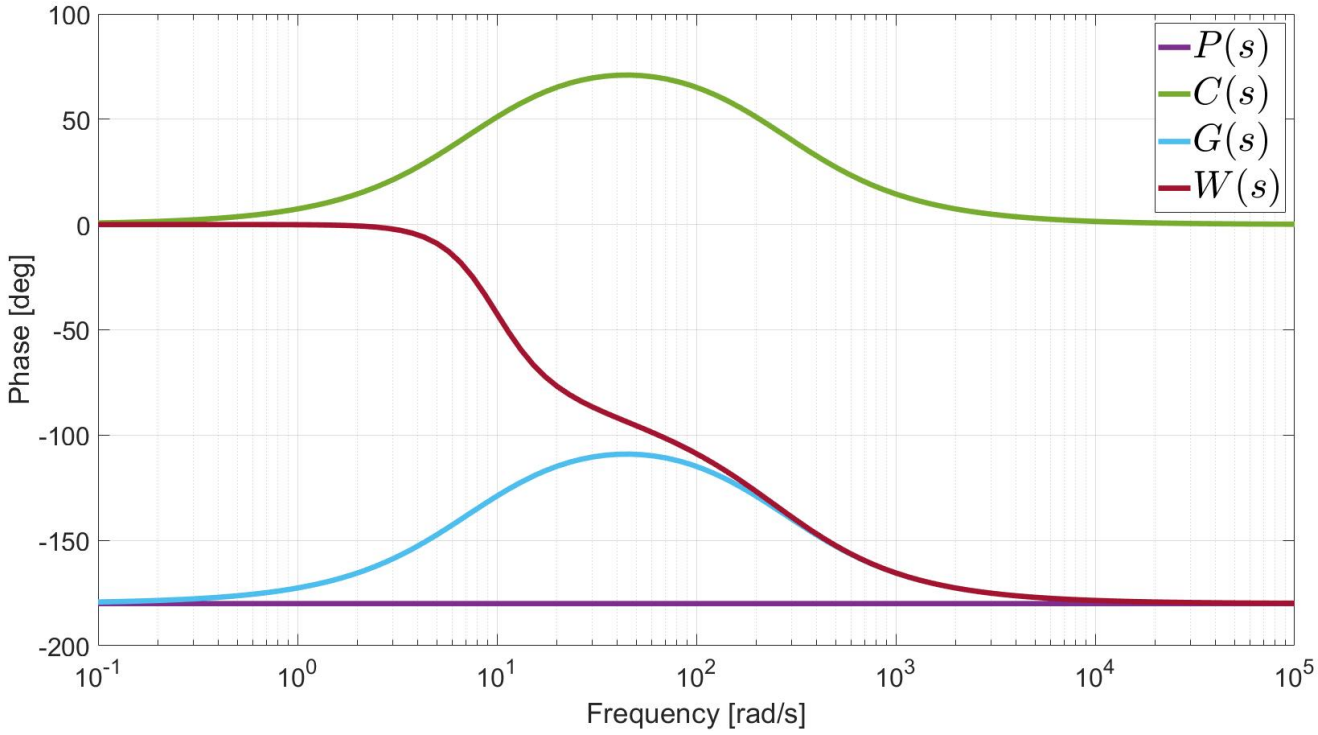

Figure 20: Bode phase plot of $P(s)$, $C(s)$, $G(s)$ and $W(s)$ with $b = 0$.

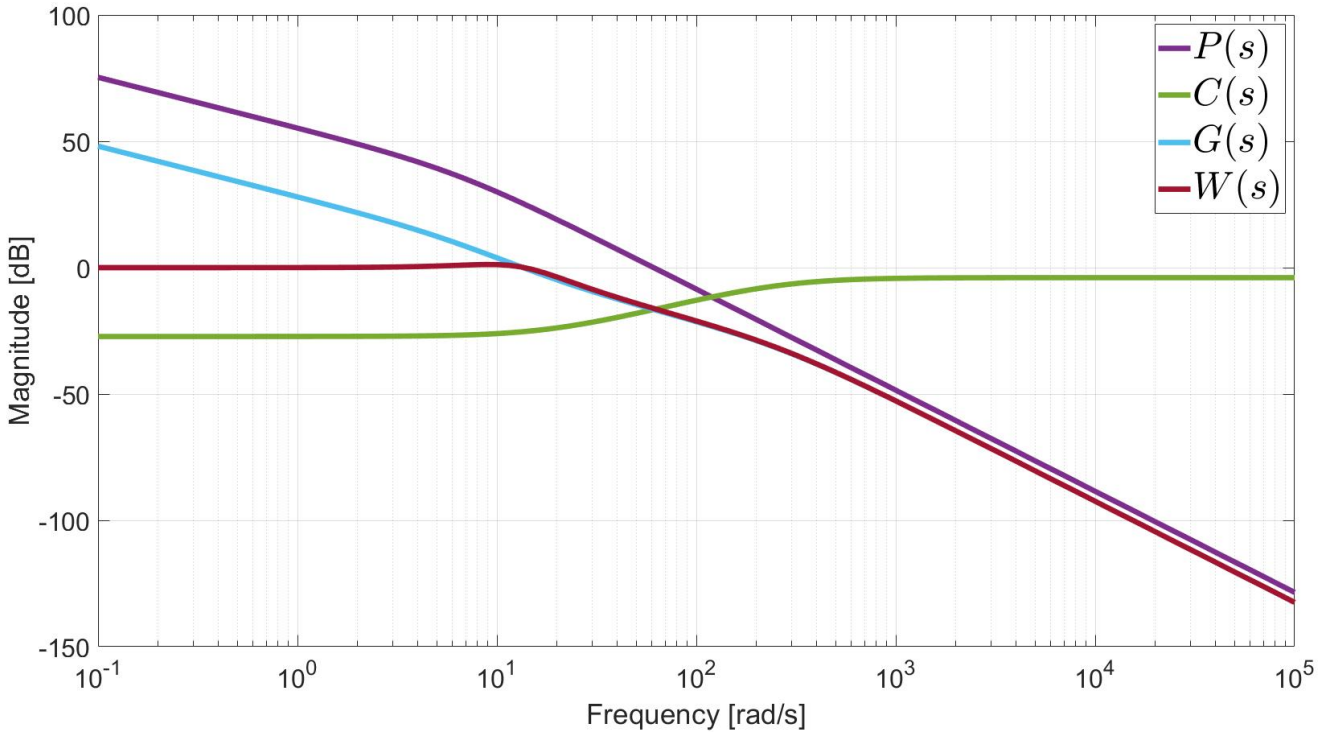

Figure 21: Bode magnitude plot of $P(s)$, $C(s)$, $G(s)$ and $W(s)$ with $b = \hat{b}$.

the damping coefficient. For the first scenario, since b is not known, the standard procedure is to be conservative and assuming $b = 0$, i.e. the transfer function to be used to design the PI controller is

$$P_{b=0}(s) = \frac{K_i K_t}{J s^2}$$

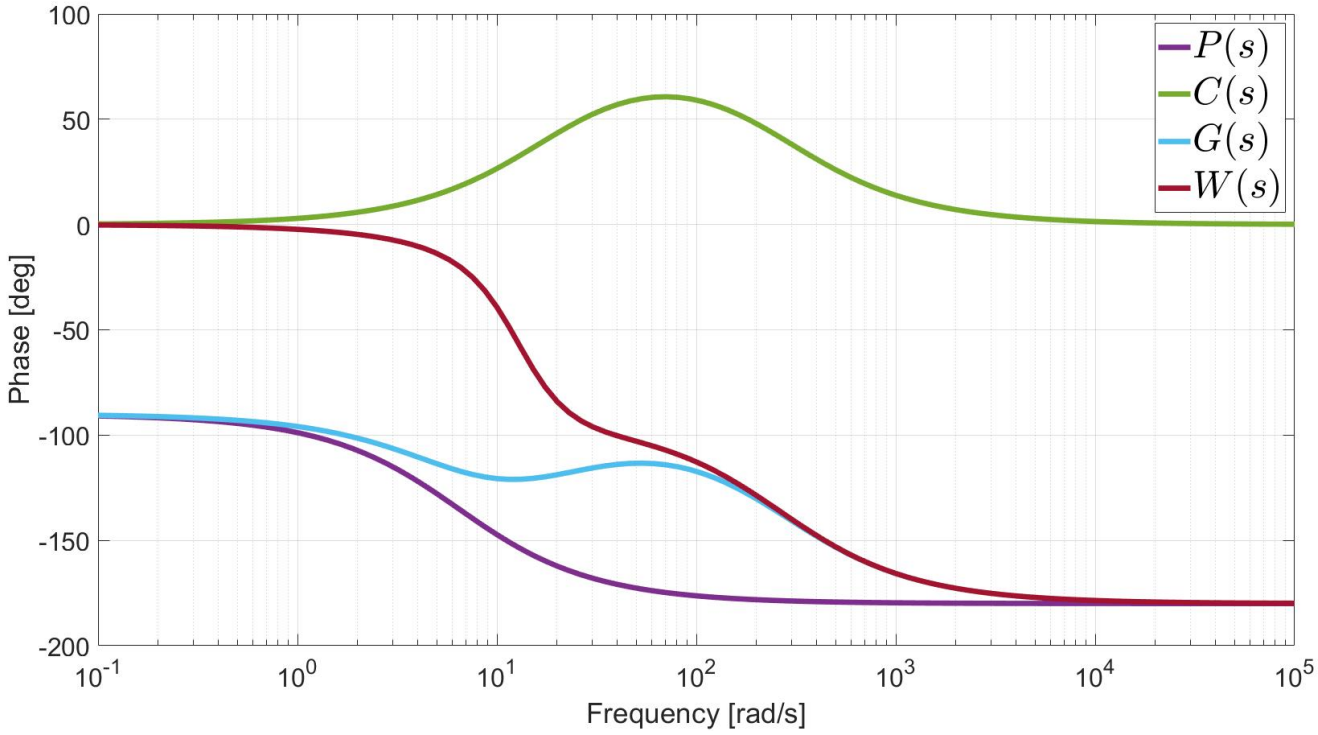

Figure 22: Bode phase plot of $P(s)$, $C(s)$, $G(s)$ and $W(s)$ with $b = \hat{b}$.

while in the second scenario we can use the transfer function with the estimated parameter

$$P_{b=\hat{b}}(s) = \frac{K_i K_t}{Js^2 + \hat{b}s}$$

Since the controller as it is designed is not physically realizable, it is necessary to add a pole with a higher frequency than ω_c. The chosen frequency is $\frac{1}{\tau_c} = 20\omega_c$. This choice is a compromise: In fact we want to have $\frac{1}{\tau_c} \gg \omega_c$ so that its contribution to the overall closed loop transfer function at the angular crossover frequency is negligible. However, at the same time we do not want to set $\frac{1}{\tau_c}$ too small to avoid large amplification of the measurement noise. The following parameters for the controller $C(s)$ have been obtained (Table 1):

$P_b(s)$	K_P	K_D	τ_c
$b = 0$	0.02366	0.00307	0.00375
$b = \hat{b}$	0.04344	0.00222	0.00375

Table 1: PD controllers parameters

The bode diagrams of $P_{b=0}(s)$ and $P_{b=\hat{b}}(s)$ functions are shown in Figure 19, 20, 21 and 22.

In the Simulink project the PD block replaces the Kp block, as shown in the Figure 23.

Finally, the responses of the real system using the two controllers are to be compared (Figure 24).

As we can see, the controller realized without knowing b does not respect the limits imposed by the project, since the settling time exceeds the limit of 0.5s. This makes us understand how important it is to correctly estimate the parameters for a proper functioning system. To be thorough, the overshoot and the settling time for the two systems considered are reported (Table 2).

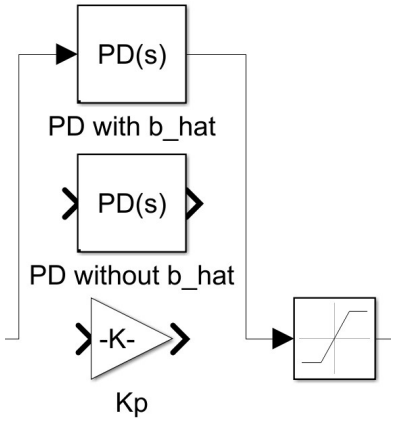

Figure 23: PD block location in Simulink project.

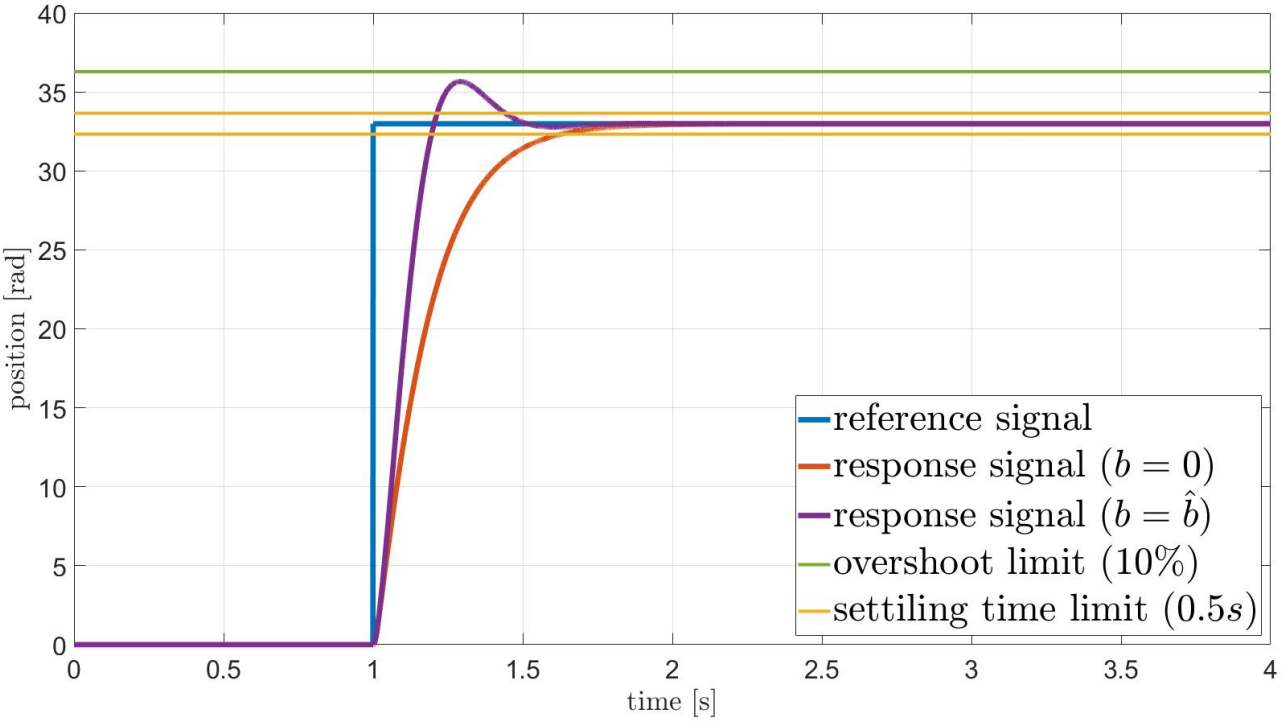

Figure 24: Comparison of signal responses with $b = 0$ and $b = \hat{b}$.

$P_b(s)$	Overshoot [%]	t_s [s]
$b = 0$	0	0.630
$b = \hat{b}$	8.1	0.444

Table 2: Overshoot and settiling time