Modeling & Simulation of Dynamic Systems

Gitopoulos Giorgos, 9344

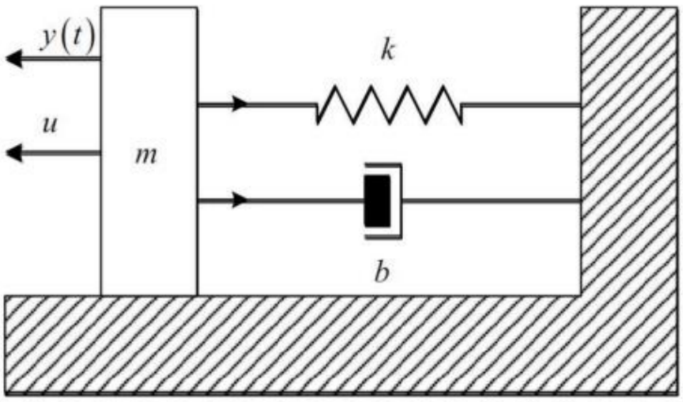
Electrical & Computer Engineering AUTh

1. Introduction

This project deals with parameter estimation of a dynamic system. After modeling the given system, we use a filter to bring it in a specific form, so that we can export the parameters estimation by minimizing the mean square error between the system states and the model states. We studied two separate systems and the analysis is presented below.

2. System 1: Damped Oscillation

In the first part, we model and simulate the following system:



According to Physics, the System 1 is described by the following differential equation:

$$m\ddot{y} = -ky - b\dot{y} + u \quad (1)$$

where m is the mass of the object, k the spring constant, b the damping constant, u the applied force and y the displacement of the object. We bring (1) to the form:

$$\ddot{y} + \frac{b}{m}\dot{y} + \frac{k}{m}y = \frac{1}{m}u \quad (2)$$

2.1. Linear configuration

We assume

$$\theta^* = \begin{bmatrix} \frac{b}{m} & \frac{k}{m} & \frac{1}{m} \end{bmatrix}^T, \quad \Delta = [\, -\dot{y} \;\; -y \quad u \,]^T$$

so that

$$\ddot{y} = \theta^{*T}\Delta \quad (3)$$

In order to get rid of the derivatives that are not practically measurable, we decide to filter (3) with $\frac{1}{\Lambda(s)}$, where $\Lambda(s) = (s+p_1)(s+p_2)$ and p_1, p_2 are selected as positive values, so that the filter is stable. After filtering, the model can be described by

$$y = \theta_\lambda^T \zeta \quad (4)$$

where

$$\theta_\lambda = [\, \theta_1^{*T} - \lambda^T \quad \theta_2^{*T} \,]^T,\ \theta_1^* = \left[\frac{b}{m}\ \frac{k}{m} \right]^T,\ \theta_2^* = \left[\frac{1}{m} \right]^T,\ \lambda = [\, p_1 + p_2 \ \ p_1 p_2 \,]^T,$$

so we find

$$\theta_\lambda = \left[\frac{b}{m} - (p_1 + p_2) \quad \frac{k}{m} - p_1 p_2 \quad \frac{1}{m} \right]^T \quad (5)$$

and

$$\zeta = \left[-\frac{\dot{y}}{\Lambda(s)} \quad -\frac{y}{\Lambda(s)} \quad \frac{u}{\Lambda(s)} \right]^T = \left[-\frac{sy}{\Lambda(s)} \quad -\frac{y}{\Lambda(s)} \quad \frac{u}{\Lambda(s)} \right]^T \quad (6).$$

Equation (4) is called linear configured form of the model. ζ is produced by measuring the input u and the output y of the system.

2.2. Minimization of mean squared error

Having a set of real system input - output measurements $y(t_i)$, $i = 1, 2, ..., N$, we are going to compute a θ_0 value that minimizes the mean squared error between the real system outputs $y(t_i)$ and the model outputs $\hat{y}(t_i)$:

$$\theta_0 = argmin_\theta \frac{1}{N} \sum_{i=1}^{N} \frac{(y(t_i) - \hat{y}(t_i))^2}{2} \quad (7).$$

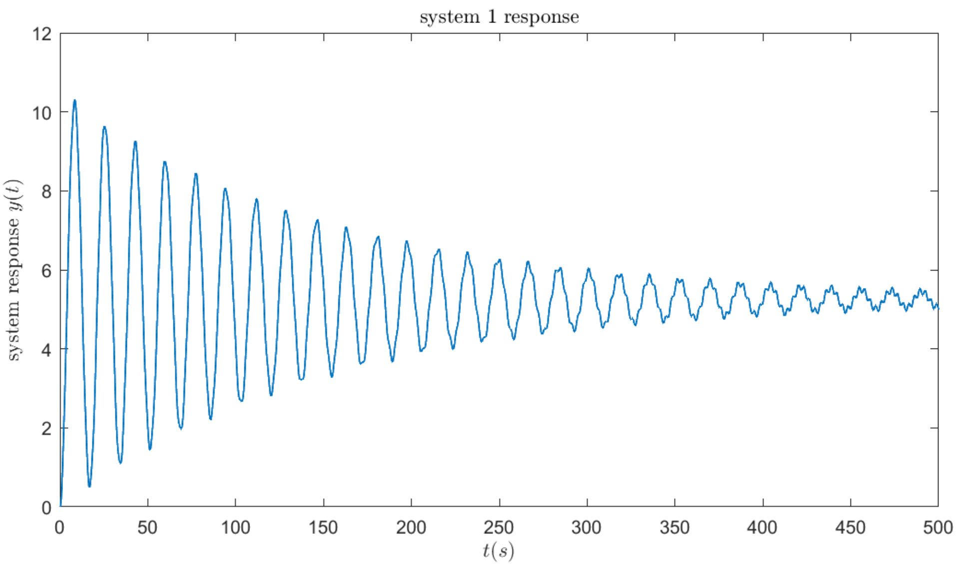
The solution of (7) is

$$\theta_0 = \left(\frac{1}{N} \sum_{i=1}^{N} \zeta(t_i) \zeta^T(t_i) \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} \zeta(t_i) y(t_i) \right).$$

We implemented this procedure as a Matlab function, where given the ζ matrix and a set of the real system outputs, θ_0 is computed and the optimized set of parameters $\hat{m}, \hat{k}, \hat{b}$ is estimated solving the equation system $\theta_\lambda = \theta_0$. In this way, we can estimate the model parameters, using a set of input - output measurements of the real system and a right choice of filter poles p_1, p_2.

2.3. Simulation in Matlab

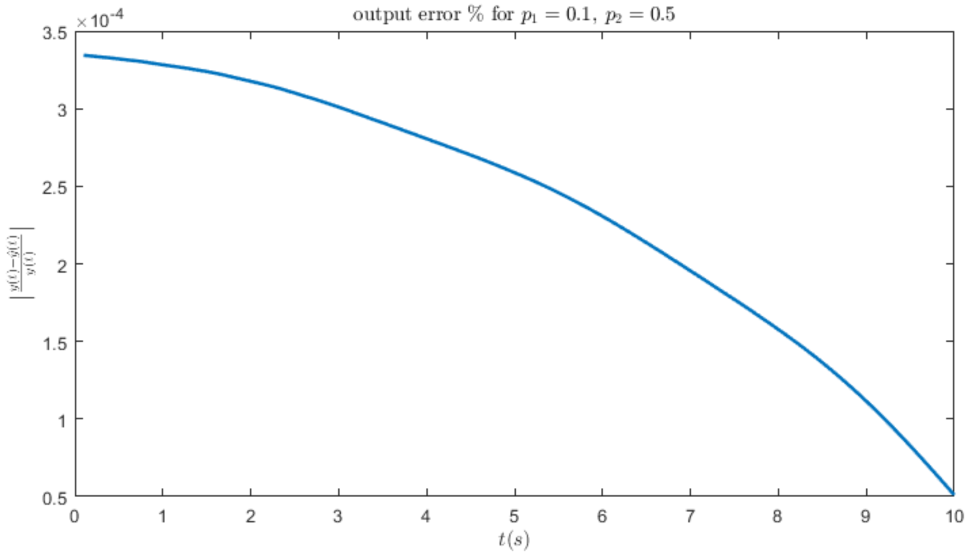
We assume the real system parameters: $m = 15kg, b = 0.2kg/sec, k = 2kg/sec^2$ and $u = 5sin(2t) + 10.5N$. The output $y(t)$ of the system with respect to time is presented in the next figure.



It is clear that it is about damped oscillation. Sampling for 10sec with 0.1sec step using *ode45* function and zero initial conditions, we store the outputs $y(t_i)$ in the matrix *state* and the inputs $u(t_i)$ in the matrix *in*. Then, for variable values of filter poles, we execute the process that described in [2.1] and [2.2], to export estimations for the parameters m, b, k. The optimized poles values will be found based on the error between the system and the model parameters as

$$e = \left|\frac{m - \hat{m}}{m}\right| + \left|\frac{b - \hat{b}}{b}\right| + \left|\frac{k - \hat{k}}{k}\right|.$$

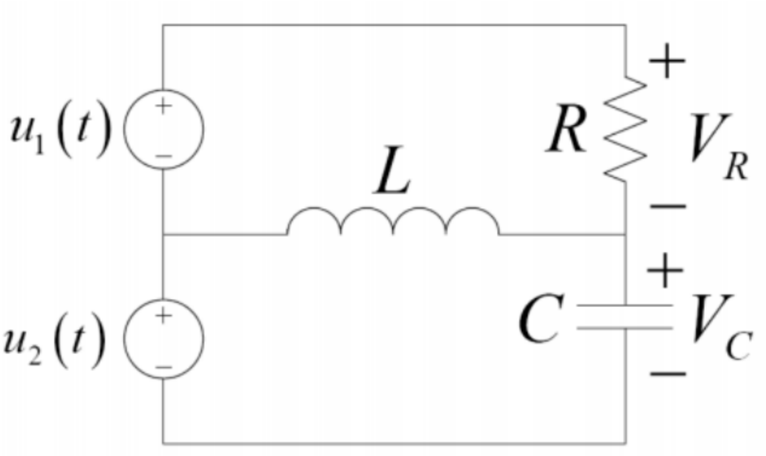
The script *sim1_error.m*, tests values of p_1 and p_2 between 0.1 and 5 with step 0.2 and finds out that the optimized poles selection is $p_1 = 0.1,\ p_2 = 0.5$ leading to $e = 0.0013 = 0.13\%$. The estimated parameters are $\hat{m} = 14.9950,\ \hat{b} = 0.2002,\ \hat{k} = 1.9997$ and the output error for the selected filter with respect to time is presented below:



Note that the sampling time is shorter than the oscillation period and for a longer sampling time the optimized poles may vary. However, the estimation of the parameters is really close to the real parameters and the result is satisfying.

3. System 2: Electrical Circuit

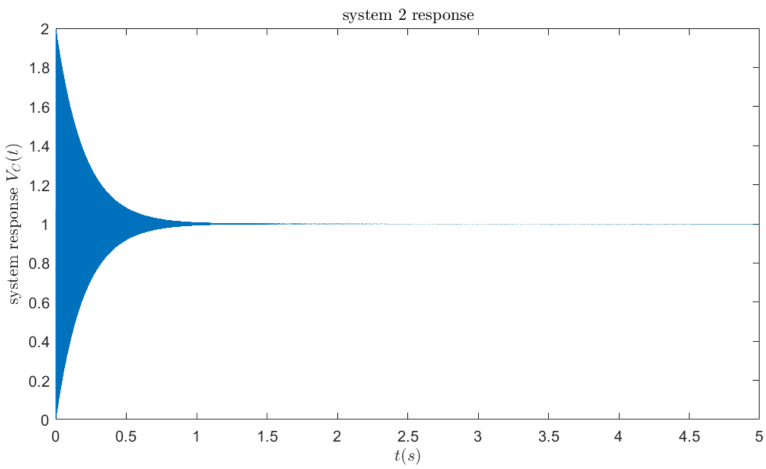
The second system that we study is the following:



where $u_1 = 2sin(t)\ V$ and $u_2 = 1\ V$. Also, we can measure both V_R and V_C. Applying Kirchhoff Laws in the electrical circuit we reach in the differential equation of the system:

$$\ddot{V_C} + \frac{1}{RC}\dot{V_C} + \frac{1}{LC}V_C = \frac{1}{RC}\dot{u_2} + \frac{1}{LC}u_2 + \frac{1}{RC}u_1 \quad (8)$$

Considering V_C as output, the system output with respect to time is presented below:



3.1. Linear configuration

In the same way as in the first part, we assume

$$\theta^* = \begin{bmatrix} \frac{1}{RC} & \frac{1}{LC} & \frac{1}{RC} & \frac{1}{LC} & \frac{1}{RC} & 0 \end{bmatrix}^T, \quad \Delta = [\, -\dot{V_C} \quad -V_C \quad \dot{u_2} \quad u_2 \quad \dot{u_1} \quad u_1 \,]^T$$

so that

$$\ddot{y} = \theta^{*T}\Delta \quad (9)$$

We filter (9) with $\frac{1}{\Lambda(s)}$, where $\Lambda(s) = (s + p_1)(s + p_2)$. After filtering, the model can be described by

$$y = \theta_\lambda^T \zeta \quad (10)$$

where

$$\theta_\lambda = [\,\theta_1^{*T} - \lambda^T \quad \theta_2^{*T}\,]^T,\ \theta_1^* = \left[\,\frac{1}{RC}\ \ \frac{1}{LC}\,\right]^T,\ \theta_2^* = \left[\frac{1}{RC}\ \ \frac{1}{LC}\ \ \frac{1}{RC}\ \ 0\right]^T,\ \lambda = [\,p_1 + p_2 \ \ p_1 p_2\,]^T,$$

so we find

$$\theta_\lambda = \left[\,\frac{1}{RC} - (p_1 + p_2) \quad \frac{1}{LC} - p_1 p_2 \quad \frac{1}{RC}\ \ \frac{1}{LC}\ \ \frac{1}{RC}\ \ 0\right]^T \quad (11)$$

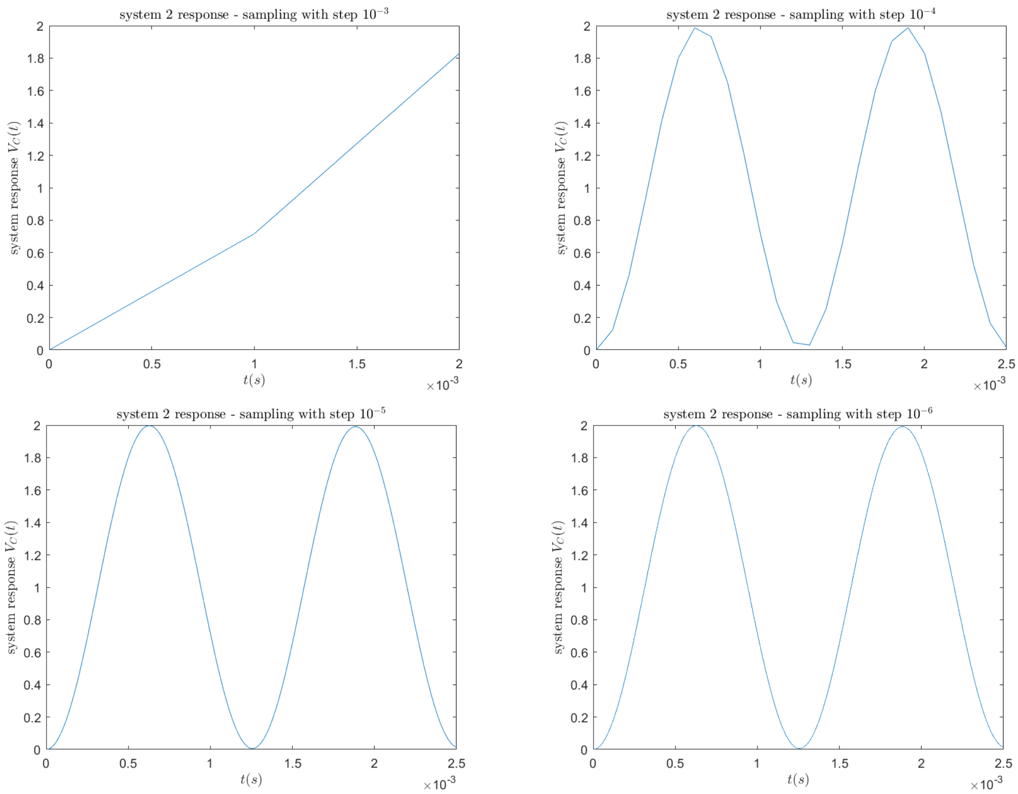
and

$$\zeta = \left[\,-\frac{\dot{V}_C}{\Lambda(s)} \quad -\frac{V_C}{\Lambda(s)}\ \ \frac{\dot{u}_2}{\Lambda(s)}\ \ \frac{u_2}{\Lambda(s)}\ \ \frac{\dot{u}_1}{\Lambda(s)}\ \ \frac{u_1}{\Lambda(s)}\,\right]^T = \left[\,-\frac{sV_C}{\Lambda(s)} \quad -\frac{V_C}{\Lambda(s)}\ \ \frac{su_2}{\Lambda(s)}\ \ \frac{u_2}{\Lambda(s)}\ \ \frac{su_1}{\Lambda(s)}\ \ \frac{u_1}{\Lambda(s)}\,\right]^T \quad (12).$$

Vector ζ can be computed using input and output measurements.

3.2. Parameters estimation

We had to select quite short step of sampling time in order to fit the high frequency of the system. The system output for different steps of sampling for the first two periods is presented bellow:



Step 10^{-5}sec provides a satisfying representation of the system, but to avoid deflection of possible future frequency changes, the step was set in 10^{-6}sec and the sampling time was 0 to 5sec.

We applied the mean squared error minimization method that described in [2.2] for various values of filter poles and exported the parameters estimation for every couple of pole values. Then, for each couple of pole values we computed the mean absolute error between the system output V_C and the model output $\hat{V}_C$. The mean absolute output error was defined as

$$e = \frac{1}{N}\sum_{i=1}^{N} \left|V_C(t_i) - \hat{V}_C(t_i)\right|$$

and the poles range that was tested is [200, 700]. The minimized error e was found for $p_1 = 370$ and $p_2 = 380$ and its value was $e = 0.000695\ V$. For this selection of poles, we end up in the parameters estimation $\hat{\frac{1}{RC}} = 9.9985$, $\hat{\frac{1}{LC}} = 24991099$. Now we are going to find the transport matrix of the system G. We have $V_C = GU$, where $U = [u_1(s)\ \ u_2(s)]^T$. Applying Laplace Transform in (8) we find

$$\left(s^2 + \frac{1}{RC}s + \frac{1}{LC}\right) V_C = \frac{1}{RC}s\, u_1 + \left(\frac{1}{RC}s + \frac{1}{LC}\right) u_2 \implies V_C = \begin{bmatrix} \frac{\frac{1}{R_C}s}{P(s)} & \frac{\frac{1}{R_C}s + \frac{1}{L_C}}{P(s)} \end{bmatrix} U, \ \ P(s) = s^2 + \frac{1}{RC}s + \frac{1}{LC}$$

so

$$G(s) = \begin{bmatrix} \frac{\frac{1}{R_C}s}{P(s)} & \frac{\frac{1}{R_C}s + \frac{1}{L_C}}{P(s)} \end{bmatrix} = \begin{bmatrix} \frac{9.9985s}{P(s)} & \frac{9.9985s + 24991099}{P(s)} \end{bmatrix}, \ \ P(s) = s^2 + 9.9985s + 24991099.$$

3.3. Technical error effect

Now, we suppose that three of our real system output measurements are false, adding technical error in three values of the V_C vector. We are going to analyze the effect of these errors in the estimation parameters that come out of the mean squared error minimizing method.
Using our selected optimal poles ($p_1 = 370,\ \ p_2 = 380$) and three randomly false V_C values, the results are the following:

$$\left(\hat{\frac{1}{RC}}\right)_{err} = 40.5812, \ \ \left(\hat{\frac{1}{LC}}\right)_{err} = 4744768.$$

The error $\%$ in parameters estimation is

$$e\left(\hat{\frac{1}{RC}}\right) = \left|\frac{9.9985 - 40.5812}{9.9985}\right| = 3.058 = 305.8\%, \ \ e\left(\hat{\frac{1}{LC}}\right) = \left|\frac{24991099 - 4744768}{24991099}\right| = 0.81 = 81\%.$$

while the output error of the model using these parameters is 0.0264, about 38 times the error that computed in [3.2] with the selected parameters.
It is clear that only three measurement errors lead to huge variation in estimated parameters and can absolutely ruin the model behavior. Following the parameters estimation method for the new set of measurements V_C will result in completely different output and the model behavior will not approach the real system.

4. Code functionality

As already mentioned, the code of the project was written in Matlab. You can run the code using *demo* file (instructions included).

4.1. System 1

Function *sim1* is used to simulate the System 1. It takes as arguments the values of the filter poles p_1, p_2. Given the real system parameters, we use *ode45* function to sample the differential equation of the system and export some measurements of the real system. Afterwards, we apply the filter $L(s)$ to configure the model as described in [2.1] and using the minimization of mean squared error method that was described in [2.2] and implemented by *mean_sq_err* function, we export the estimation of the system parameters. *sim1* function returns the error of the estimated parameters in comparison with the real parameters as mentioned in [2.3]. Function *sim1_err*, tests *sim1* for a selected range of poles values and finds the optimal selection with respected to the computed error.

4.2. System 2

The System 2 is simulated by *sim2* function. This function needs the filter poles as arguments too, just like *sim1*. The measurements now are exported and stored using the *v.p* file. There is an option to add technical error ([3.3]). The model is filtered and as previously, we estimate the parameters using *mean_sq_err* function. The error is now computed as described in [3.2] and returned by the function. *sim2_err* works just like *sim1_err* and finds the best poles values.