Assignment 3

TTK4215	
Elias Olsen Almenningen	
Version: 1.0	Date: September 14, 2023

Contents

Problem 1

Consider the nonlinear system

$$\dot{x} = a_1 f_1(x) + a_2 f_2(x) + b_1 g_1(x)u + b_2 g_2(x)u \tag{1}$$

where $u, x \in \mathcal{R}$; f_i, g_i are known nonlinear functions of x; and a_i, b_i are unknown constant parameters and $i = 1, 2$. The system is such that $u \in \mathcal{L}_\infty$ implies $x \in \mathcal{L}_\infty$. Assuming that x, u can be measured at each time t, design an estimation scheme for estimating the unknown parameters online.

Taking the Laplace transform of (1) yields

$$sx = a_1 f_1(x) + a_2 f_2(x) + b_1 g_1(x)u + b_2 g_2(x)u \tag{2}$$

Since $\dot{x}$ cant be measured, we need to filter the signals through the filter $\Lambda(s) = s + \lambda_0$. Using the hint given in the assignment, we can write the system as an SPM:

$$z = \theta^{*\top}\phi \tag{3}$$

where

$$z = \frac{s}{\Lambda(s)}[x] \tag{4}$$

$$\theta^* = [a_1, a_2, b_1, b_2]^\top \tag{5}$$

$$\phi = \left[\frac{1}{\Lambda(s)}[f_1(x)], \frac{1}{\Lambda(s)}[f_2(x)], \frac{1}{\Lambda(s)}[g_1(x)u], \frac{1}{\Lambda(s)}[g_2(x)u],\right]^\top \tag{6}$$

Problem 2

Design an online estimation scheme to estimate the coefficients of the numerator polynomial

$$Z(s) = b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_1 s + b_0 \tag{7}$$

of the plant

$$y = \frac{Z(s)}{R(s)}u \tag{8}$$

when the coefficients of $R(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0$ are known. Repeat the same problem when $Z(s)$ is known and $R(s)$ is unknown.

a) $R(s)$ is known

Assuming we can measure both y, u, rewriting (8) s.t. the known is on the left hand side and unknowns on the right hand side:

$$yR(s) = Z(s)u \tag{9}$$

Then the filter can be defined as $\Lambda(s) = s^n + \lambda_{n-i}s^{n-1} + \cdots + \lambda_1 s + \lambda_0$. Writing the estimation scheme as an SPM:

$$z = \frac{s^n}{\Lambda(s)}[y] + \frac{a_{n-1}s^{n-1}}{\Lambda(s)}[y] + \cdots + \frac{a_1 s}{\Lambda(s)}[y] + \frac{a_0}{\Lambda(s)}[y] \tag{10}$$

$$\theta^* = [b_{n-1}, b_{n-2}, \ldots, b_1, b_0]^\top \tag{11}$$

$$\phi = \left[\frac{s^{n-1}}{\Lambda(s)}[u] + \frac{s^{n-2}}{\Lambda(s)}[u] + \cdots + \frac{s}{\Lambda(s)}[u] + \frac{1}{\Lambda(s)}[u]\right]^\top \tag{12}$$

b) $Z(s)$ is known

Assuming we can measure both y, u, rewriting (8) s.t. the known is on the left hand side and unknowns on the right hand side:

$$Z(s)u = yR(s) \tag{13}$$

Then the filter can be defined as $\Lambda(s) = s^n + \lambda_{n-i}s^{n-1} + \cdots + \lambda_1 s + \lambda_0$. Writing the estimation scheme as an SPM:

$$z = \frac{b_{n-1}s^{n-1}}{\Lambda(s)}[u] + \cdots + \frac{b_1 s}{\Lambda(s)}[u] + \frac{b_0}{\Lambda(s)}[u] \tag{14}$$

$$\theta^* = [1, a_{n-1} \ldots, a_1, a_0]^\top \tag{15}$$

$$\phi = \left[\frac{s^n}{\Lambda(s)}[y] + \cdots + \frac{s}{\Lambda(s)}[y] + \frac{1}{\Lambda(s)}[y]\right]^\top \tag{16}$$

this is wrong however, since there can't be a known parameter in θ^*. Therefore, rewriting

$$z = \frac{b_{n-1}s^{n-1}}{\Lambda(s)}[u] + \cdots + \frac{b_1 s}{\Lambda(s)}[u] + \frac{b_0}{\Lambda(s)}[u] - \frac{s^n}{\Lambda(s)}[y] \tag{17}$$

$$\theta^* = [a_{n-1}\ldots, a_1, a_0]^\top \tag{18}$$

$$\phi = \left[\frac{s^{n-1}}{\Lambda(s)}[y]\cdots + \frac{s}{\Lambda(s)}[y] + \frac{1}{\Lambda(s)}[y]\right]^\top \tag{19}$$

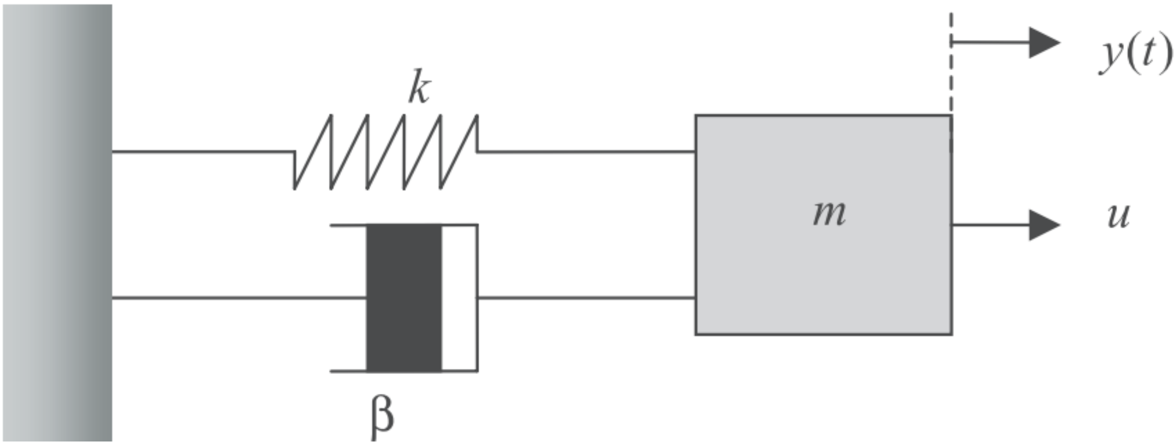

Figure 1: Mass-spring-damper system

Problem 3

Consider the mass–spring–damper system shown in Figure 1, where β is the damping coefficient, k is the spring constant, u is the external force, and $y(t)$ is the displacement of the mass m resulting from the force u.

a)

Setting up Newtons 2nd law:

$$\sum f = ma \tag{20}$$

$$u - \beta\dot{y} - Ky = m\ddot{y} \tag{21}$$

$$m\ddot{y} + \beta\dot{y} + ky = u \tag{22}$$

If one pulls the mass to far to the right, the spring is no longer linear, making the model wrong, but for small displacements the dynamic behaviour is as described.

b) Gradient algorithm

Transform (22) into an SPM:

$$z = {\theta^*}^\top \phi \tag{23}$$

by taking the Laplace transform, defining the filter $\Lambda(s) = s^2 + \lambda_1 s + \lambda_0$ and rearranging the equation gives:

$$\frac{ms^2y + \beta sy + k}{\Lambda(s)} = \frac{u}{\Lambda(s)} \tag{24}$$

making

$$z = \left[\frac{u}{\Lambda(s)}\right] \tag{25}$$

$$\theta^* = [m, k, \beta]^\top \tag{26}$$

$$\phi = \left[\frac{s^2}{\Lambda(s)}[y] \, \frac{s}{\Lambda(s)}[y] + \frac{1}{\Lambda(s)}[y]\right]^\top \tag{27}$$

Define estimate of z:

$$\hat{z} = \theta^\top \phi \tag{28}$$

Normalizing the estimation error:

$$\epsilon = \frac{z - \hat{z}}{m^2} \tag{29}$$

where $m = 1 + n_s^2$, where n_s is chosen s.t. $\frac{\phi}{m} \in \mathcal{L}_\infty$. Normal is $n_s^2 = \alpha\phi^\top\phi$, $\alpha > 0$
Parameter estimation error:

$$\tilde{\theta} = \theta - \theta^* \tag{30}$$

The adaptive law is chosen as

$$\dot{\theta} = \Gamma\epsilon\phi \tag{31}$$

d)

Using the handed out script and filling in the blank spaces, and setting the filter constants, and adaptive gain to 1 gave the following result:

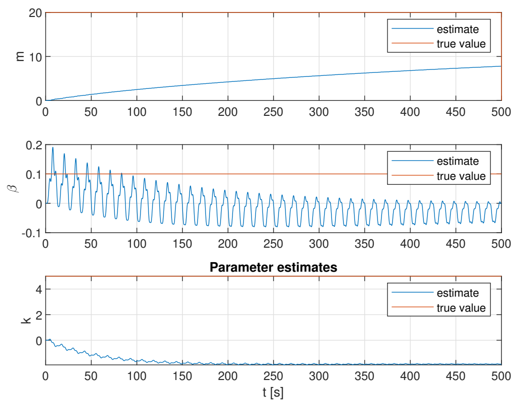

Figure 2: $\lambda_1 = \lambda_0 = \Gamma = 1$

It's clear to see that the parameter estimates does not converge towards the true values. The result of setting the filter constants to 0.5 and increasing the gain to 10 can be seen in Figure 3:

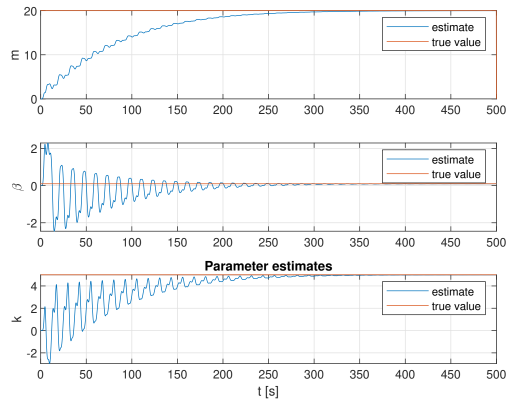

Figure 3: $\lambda_1 = \lambda_0 = 0.5$ and $\Gamma = 10$

Here the estimated parameters converge towards the actual parameter values. In both cases the input was $u = sin(t)$.

e)

When the mass changes after 20 seconds, and keeping the same parameters as in d), the mass estimate only goes to 20, as seen in Figure 4. There could be several different reasons for

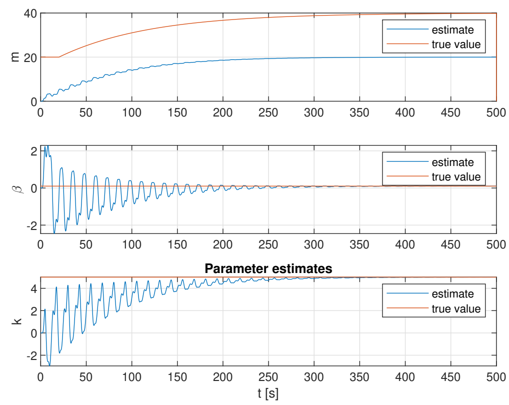

Figure 4: $\lambda_1 = \lambda_0 = 0.5$ and $\Gamma = 10$

this. The first that comes to mind is that u is not P.E or sufficiently rich. Another could be that the algorithm used can't follow changes in the model.