Mechatronic systems
Assignment 1

Shmulik Edelman

July 2022

1 Nonlinear process

For the nonlinear process described with the following model:

$$hy^{(n)}(t) + \sum_{i=1}^{n} \left[a_i y^{(n-i)}(t) + b_i f_i(y^{(n-i)}) \right] = u(t) \tag{1}$$

I will choose the dynamic system of an inverted pendulum with a spring, K, and a damper, c_1, c_2, and the joint, the mass, M, of the beam with length of L is assumed to be at the center, and there is a motor with propeller at the end which generate trust force, F, perpendicular to the beam.

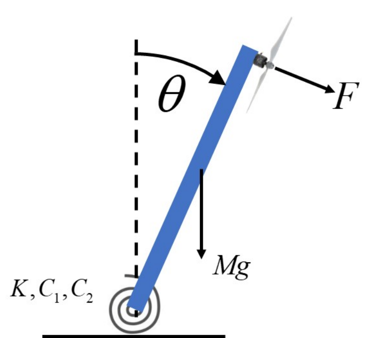


Figure 1: Dynamic system

The nonlinear model of the system is:

$$\boxed{I\ddot{\theta} + c_1\dot{\theta} + c_2\dot{\theta}^3 + k_1\theta - MgL\sin\theta = LF(t) = u(t)} \tag{2}$$

The spring is modeled as linear element and its torque is proportional to the angle. And the damper is modeled as nonlinear element:

$$\tau_{spring} = k_1 \cdot \theta$$
$$\tau_{damper} = c_1\dot{\theta} + c_2\dot{\theta}^3$$

With respect to equation 1 the coefficients are:

$$h = I = ML^2$$
$$a_1 = c_1 \qquad b_1 = c_2 \qquad f_1(\dot{\theta}) = \dot{\theta}^3$$
$$a_2 = k_1 \qquad b_2 = -MgL \qquad f_2(\theta) = \sin\theta$$

2 General adaptive control for nonlinear system with model reference

2.1 Control law and adaptation

We will choose a reference model with similar order to system model:

$$\ddot{y}_m^{(n)}(t) = \sum_{i=1}^{n}\left(a_{m,i}y_m^{(n-i)}(t)\right) + b_m r(t) \tag{3}$$

Similar to what was presented in the lectures we define the control for the model in equation 1: (Note that parameters with ˆ are estimated parameters of the real ones)

$$\boxed{u(t) = \hat{h}y_r^{(n)}(t) - kz(t) + \underbrace{\sum_{i=1}^{n}\left[\hat{a}_i y^{(n-i)}(t) + \hat{b}_i f(y^{(n-i)})\right]}_{nonlinear \quad model}} \tag{4}$$

Where:

$$y_r^{(n)} = y_m^{(n)} - \lambda_{n-2}e^{(n-1)} - ... - \lambda_0\dot{e} \tag{5}$$
$$z(t) = e^{(n-1)} + \lambda_{n-2}e^{(n-2)} + ... + \lambda_0 e \tag{6}$$
$$e(t) = y(t) - y_m(t) \tag{7}$$

We choose this type of control law that contains the system model with a nonlinear elements. This is done in purpose to eliminate the non linear elements of the model and remain only with linear system.

Substituting equation 4 into 1

$$hy^{(n)}(t) + \sum_{i=1}^{n}\left[a_i y^{(n-1)}(t) + b_i f(y^{(n-1)})\right] =$$
$$\hat{h}y_r^{(n)}(t) - kz(t) + \sum_{i=1}^{n}\left[\hat{a}_i y^{(n-1)}(t) + \hat{b}_i f(y^{(n-1)})\right]$$

$$
\begin{aligned}
&hy^{(n)}(t) + kz(t) = \\
&\qquad (\tilde{h}(t) + h)y_r^{(n)}(t) + \sum_{i=1}^{n} \left[(\hat{a}_i - a_i)\, y^{(n-1)}(t) + \left(\hat{b}_i - b_i\right) f(y^{(n-1)}) \right]
\end{aligned} \tag{8}
$$

Using next definitions for parameters error:

$$
\begin{aligned}
\tilde{a}_i(t) &= \hat{a}_i(t) - a_i \\
\tilde{b}_i(t) &= \hat{b}_i(t) - b_i \\
\tilde{h}_i(t) &= \hat{h}(t) - h \rightarrow \hat{h}(t) = \tilde{h}_i(t) + h
\end{aligned} \tag{9}
$$

$$
\begin{aligned}
&h\left(y^{(n)}(t) - y_r^{(n)}(t)\right) + kz(t) = \\
&\qquad \tilde{h}(t) y_r^{(n)}(t) + \sum_{i=1}^{n} \left[\tilde{a}_i(t) y^{(n-1)}(t) + \tilde{b}_i(t) f(y^{(n-1)}) \right]
\end{aligned} \tag{10}
$$

Using the derivative of $\dot{z}(t)$ from equation 6, we get:

$$
\boxed{h\dot{z} + kz(t) = \tilde{h}_i(t) y_r^{(n)}(t) + \sum_{i=1}^{n} \left[\tilde{a}_i(t) y^{(n-1)}(t) + \tilde{b}_i(t) f(y^{(n-1)}) \right]} \tag{11}
$$

Next step will be Laplace transform:

$$
\begin{aligned}
&s\mathscr{L}\{z(t)\} + \frac{k}{h}\mathscr{L}\{z(t)\} = \\
&\qquad \mathscr{L}\{\frac{1}{h}\left(\hat{h} y_r^{(n)}(t) - kz(t) + \sum_{i=1}^{n} \left[\hat{a}_i y^{(n-1)}(t) + \hat{b}_i f(y^{(n-1)}) \right] \right)\}
\end{aligned} \tag{12}
$$

$$
\begin{aligned}
\mathscr{L} z(t) = \underbrace{\frac{1}{s + k/h}}_{SPR} &\mathscr{L}\{\frac{1}{h}\,(\,\hat{h} y_r^{(n)}(t) - kz(t) + \\
&\sum_{i=1}^{n} \left[\hat{a}_i y^{(n-1)}(t) + \hat{b}_i f(y^{(n-1)}) \right]\}
\end{aligned} \tag{13}
$$

The SPR is a first order stable function that we know the adaptation laws for it:

$$
\boxed{
\begin{aligned}
\dot{\hat{h}}(t) &= -\gamma sgn(h) z(t) y_r^{(n)}(t) \\
\dot{\hat{a}}_i(t) &= -\gamma sgn(h) z(t) y^{(n)}(t) \\
\dot{\hat{h}}(t) &= -\gamma sgn(h) z(t) y_r^{(n)}(t)
\end{aligned}
} \tag{14}
$$

2.2 Stability proof

To proof the stability of the adaptive controller we will need to choose a Lyapunov function and proof that its monotonically reducing and bounded, and its second derivative is bounded as well.

Choosing the Lyapunov function:

$$\boxed{V(t) = |h|z(t)^2 + \gamma^{-1}\left(\tilde{h}^2 + \sum_{i=1}^{n} \tilde{a}_i(t)^2 + \tilde{b}_i(t)^2\right) \geq 0} \tag{15}$$

Differentiating the Lyapunov function:

$$\dot{V} = \frac{\partial V}{\partial t} = 2|h|z\dot{z} + \gamma^{-1}\left(2\tilde{h}\dot{h} + \sum_{i+1}^{n} 2\tilde{a}_i(t)\dot{\hat{a}}_i(t) + 2\tilde{b}_i(t)\dot{\hat{b}}_i(t)\right) \tag{16}$$

Using equation 11 and transform terms to get $\dot{z}$:

$$\begin{aligned}\dot{z}(t) =& \frac{1}{h}\left(\tilde{h}_i(t)y_r^{(n)}(t) + \sum_{i=1}^{n}\left[\tilde{a}_i(t)y^{(n-i)}(t) + \tilde{b}_i(t)f(y^{(n-i)})\right]\right. \\ & -kz(t))\end{aligned} \tag{17}$$

Now substituting $\dot{z}$ into equation 16:

$$\begin{aligned}\dot{V} =& 2z\frac{|h|}{h}\left(\tilde{h}_i(t)y_r^{(n)}(t) + \sum_{i=1}^{n}\left[\tilde{a}_i(t)y^{(n-1)}(t) + \tilde{b}_i(t)f(y^{(n-1)})\right]\right) - kz(t)) + \\ & \gamma^{-1}\left(2\tilde{h}\dot{h} + \sum_{i+1}^{n} 2\tilde{a}_i(t)\dot{\hat{a}}_i(t) + 2\tilde{b}_i(t)\dot{\hat{b}}_i(t)\right)\end{aligned} \tag{18}$$

Using the adaptation laws from equations 14 the term of $\dot{V}$ will reduce to:

$$\boxed{\dot{V} = -2|k|z^2 \leq 0} \tag{19}$$

As can be seen, this term is negative which means the the Lyapunov function that we chose is monotonically decreasing, and it also is bounded from above by its initial value:

$$0 \leq V(t) \leq V(t_0) \tag{20}$$

Next step will be proving that the Lyapunov function second derivative is bounded too.

$$\boxed{\ddot{V} = -4|k|z\dot{z}} \tag{21}$$

Substituting $\dot{z}$: from equation 17 into equation 21

$$\ddot{V} = -4|k|z\frac{1}{h}\left(\underbrace{\tilde{h}_i(t)y_r^{(n)}(t)}_{1} + \underbrace{\sum_{i=1}^{n}\left[\tilde{a}_i(t)y^{(n-i)}(t) + \tilde{b}_i(t)f(y^{(n-i)})\right]}_{2}\underbrace{-kz(t)}_{3}\right) \tag{22}$$

To prove that $\ddot{V}(t)$ is bounded we need to examine each of added elements

$$(1):\qquad -4|k|z\frac{1}{h}\tilde{h}_i(t)y_r^{(n)}(t)$$
$$(2):\qquad -4|k|z\frac{1}{h}\sum_{i=1}^{n}\tilde{a}_i(t)y^{(n-i)}(t) + \tilde{b}_i(t)f(y^{(n-i)})$$
$$(3):\qquad 4|k|z\frac{1}{h}kz(t)$$

We proved that $z(t)$ is bounded and since it counteracted from $e(t), y_r^{(n)}(t)$ (equations 5-7) they are bounded as well. $Kandh$ are constants that don't affect the convergence or divergence of those elements. The parameters $\tilde{a}_i, \tilde{b}_i$ and $y^{(n-i)}$ are values which converge since the error is bounded.

Thus if $\ddot{V}(t)$ is bounded, then from the Lemma of Barbalat $\dot{V}(t) \to 0$, hence $z(t) \to 0$.

3 Adaptive controller - known parameters

From the general form of the control law in equation 4, assuming all parameters known, the control law is:

$$u(t) = hy_r^{(n)}(t) - Kz(t) + \sum_{i=1}^{n}\left[a_i y^{(n-1)}(t) + b_i f(y^{(n-1)})\right] \tag{23}$$

For our system shown in equation 2 we will choose the control signal to be:

$$\boxed{u(t) = h\ddot{\theta}_r(t) - Kz(t) + c_1\dot{\theta} + c_2\dot{\theta}^3 + k_1\theta + MgL\sin\theta} \tag{24}$$

And a reference model:

$$\ddot{\theta}_m(t) = -k_1\dot{\theta}_m(t) - k_2\theta_m(t) + k_3 r(t) \tag{25}$$

Substituting the control law into our system in equation 2:

$$\begin{aligned} I\ddot{\theta} + c_1\dot{\theta} + c_2\dot{\theta}^3 + k_1\theta - mgL\sin\theta =& I\ddot{\theta}_r(t) - kz(t) + \\ & c_1\dot{\theta} + c_2\dot{\theta}^3 + k_1\theta + mgL\sin(\theta) \end{aligned}$$

$$I\ddot{\theta} = I\ddot{\theta}_r(t) - kz(t) \tag{26}$$

$$I\dot{z}(t) + kz(t) = 0 \tag{27}$$

3.1 Python simulation and implementation

We are using python and scipy.integrate.odeint package to solve the differential equations. This package uses the Adams/BDF method for solving the differential equations.

To use this package, we need to define a vector time for which the solver will return the state results, initial conditions, and constant parameters we want to pass into the function. Inside the function, we define two vectors $\dot{\bar{X}}$ and $\dot{\bar{X}}_m$, which represent the dynamic model of our system and the dynamic model of the reference system.

$$\bar{X}_{initial} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ \theta_m \\ \dot{\theta}_m \end{bmatrix} \qquad \bar{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ \theta_m \\ \dot{\theta}_m \end{bmatrix} \qquad \dot{\bar{X}} = \begin{bmatrix} x_2 \\ \dot{x}_2 \\ x_4 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix}$$

where:

$$\ddot{\theta} = \frac{1}{I}\left(u(t)L - \left(c_1\dot{\theta} + c_2\dot{\theta}^3 + k_1\theta - MgL\sin\theta\right)\right)$$
$$\ddot{\theta}_m = -k_{m,1}\dot{\theta}_m(t) - k_{m,2}\theta_m(t) + k_{m,3}r(t)$$

The chosen parameters of the system are:

$$c_1 = 4 \qquad c_2 = 0.4 \qquad k_1 = 2 \qquad M = 2 \qquad L = 1.5$$
$$K_{m,1} = 2 \qquad K_{m,2} = 10 \qquad K_{m,3} = 10$$

3.1.1 Step function input

Let us examine system response with different step inputs and initial conditions. Starting from a step function input and zero initial conditions:

$$r_1(t) = \begin{cases} 0, & t < 1 \\ 10^\circ, & t \geq 1 \end{cases} \qquad \bar{X}_{initial} = \begin{bmatrix} 0, & 0, & 0, & 0 \end{bmatrix}^T$$

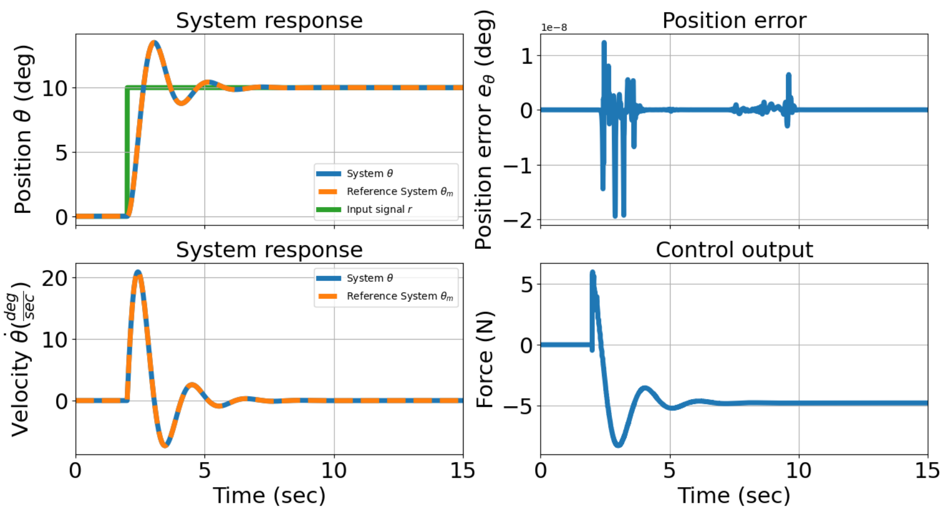


Figure 2: System response to step input $r_1(r)$

Let us separate the results into two subjects:

1. Reference system behavior - depends on the design of the reference system - this isn't the focus of this exercise.

2. system behavior - depends on the controller and how "good" the system tracks the reference system.

Looking at the upper left graph showing the system response position, we can see that the designed controller follows the system reference with a negligible error of maximum $|e_{max}| = 2^{-8}$. This result is not surprising since we designed the controller "knowing" the model parameters.

Now we will examine the system response to a step input function but with different initial conditions:

$$r_2(t) = \begin{cases} 0, & t < 5 \\ 10^\circ, & t \geq 5 \end{cases} \qquad \bar{X}_{initial} = \begin{bmatrix} 0, & 0, & -2, & 0 \end{bmatrix}^T$$

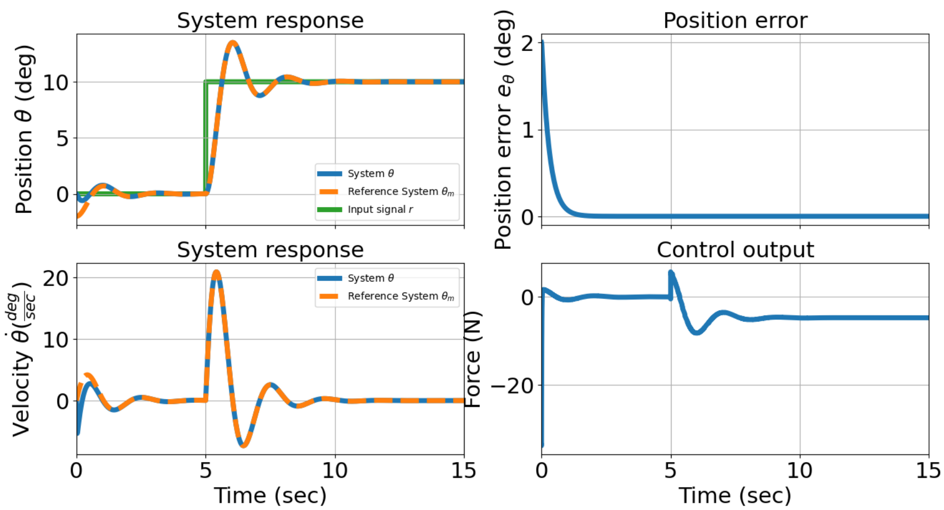


Figure 3: System response to step input $r_2(r)$

Examining the upper left graph for the position $\theta(t)$ of the beam, we see that the reference system and system have an offset of 2 degrees at the beginning of the motion. After about 2.5 seconds, the system "catches" the reference, and their error converges to zero, as expected. At the time $t = 5sec$, we see the step input signal, and the response is very similar to the previous one.

3.1.2 Wave function input

Lets examine system response a wave function and initial condition:

$$r_3(t) = 10\sin(0.1t) + 5\sin(0.2t) + 5\cos(0.5t - \frac{\pi}{2}) \qquad \bar{X}_{initial} = \begin{bmatrix} 0, & 0, & 0, & 0 \end{bmatrix}^T$$

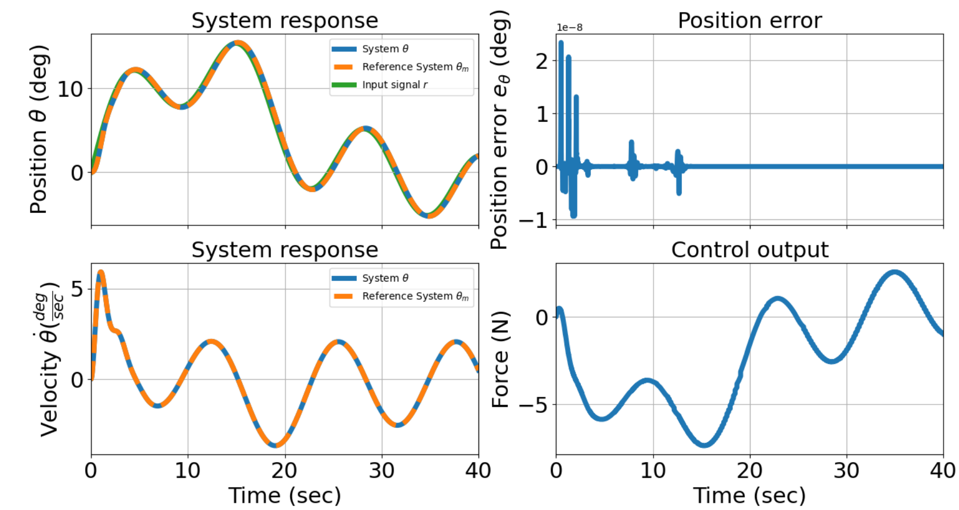


Figure 4: System response to wave input $r_3(r)$ with zero initial conditions

$$r_2(t) = 10\sin(0.1t) + 5\sin(t) + 5\cos(0.5t - \frac{\pi}{2}) \qquad \bar{X}_{initial} = \begin{bmatrix} 0, & 0, & 10, & 0 \end{bmatrix}^T$$

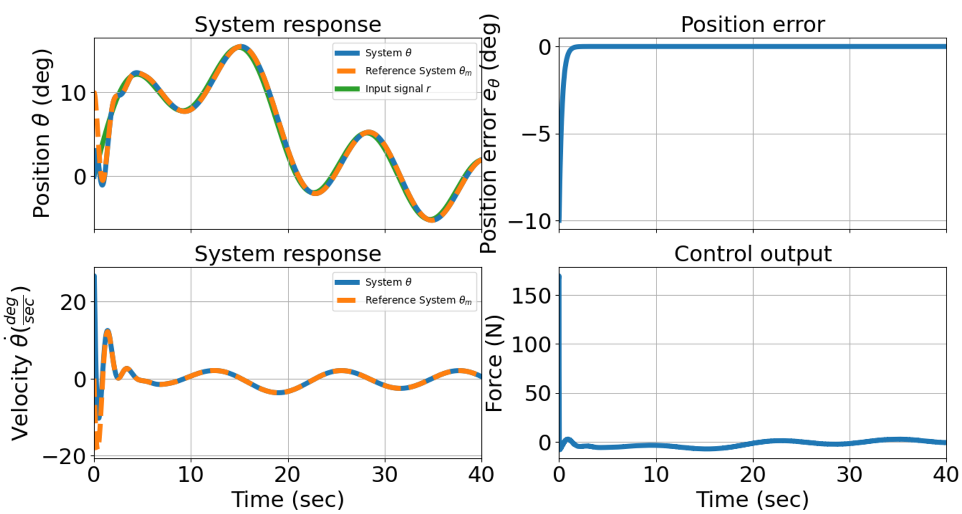


Figure 5: System response to wave input $r_3(r)$ with nonzero initial conditions

Examining both system responses to wave function, with zero and nonzero initial conditions, we see in both that the system follows the reference model almost perfectly. This is not surprising because the control signal we developed eliminated the nonlinear part of the model, and assuming that the physical parameters are known, the result would be almost perfect.

4 Adaptive controller - Unknown parameters

Unlike presented in part 3, we now will use the general form of the control law in equation 4, but assuming that all parameters are unknown,

$$u(t) = \hat{h} y_r^{(n)}(t) - Kz(t) + \sum_{i=1}^{n} \left[\hat{a}_i y^{(n-1)}(t) + \hat{b}_i f(y^{(n-1)}) \right] \tag{28}$$

$$\boxed{u(t) = \hat{h}\ddot{\theta}_r(t) - Kz(t) + \hat{c}_1\dot{\theta} + \hat{c}_2\dot{\theta}^3 + \hat{k}_1\theta + \widehat{MgL}\sin\theta} \tag{29}$$

Using the adaptation laws shown in equation 14 we can iteratively calculate them:

$$\begin{aligned}
\dot{\hat{h}}(t) &= -\gamma sgn(h) z(t) \ddot{y}_r(t) \\
\dot{\hat{c}}_1(t) &= -\gamma sgn(h) z(t) \dot{\theta}(t) \\
\dot{\hat{c}}_2(t) &= -\gamma sgn(h) z(t) \dot{\theta}^3(t) \\
\dot{\hat{k}}_1(t) &= -\gamma sgn(h) z(t) \theta(t) \\
\dot{\widehat{MgL}}(t) &= -\gamma sgn(h) z(t) \sin(\theta(t))
\end{aligned}$$

4.1 Python simulation and implementation

The implementation of that is done inside the *odeint* function. which mean in each iterations of solving the differential equation (using *odeint*) we calculate and "adjust" the parameters.

The final value of the those parameters don't have converge into the real and "true" parameters. They only have to converge into a state where the error between the reference model and the real model converge into 0 (zero).

The state vector of that passes into the *odeint* function is:

$$\bar{X}_{initial} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ \theta_m \\ \dot{\theta}_m \\ \hat{h} \\ \hat{c}_1 \\ \hat{c}_2 \\ \hat{k}_1 \\ \widehat{MgL} \end{bmatrix} \qquad \bar{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ \theta_m \\ \dot{\theta}_m \\ \hat{h} \\ \hat{c}_1 \\ \hat{c}_2 \\ \hat{k}_1 \\ \widehat{MgL} \end{bmatrix} \qquad \dot{\bar{X}} = \begin{bmatrix} x_2 \\ \dot{x}_2 \\ x_4 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\theta}_m \\ \ddot{\theta}_m \\ \dot{\hat{h}} \\ \dot{\hat{c}}_1 \\ \dot{\hat{c}}_2 \\ \dot{\hat{k}}_1 \\ \dot{\widehat{MgL}} \end{bmatrix}$$

Lets examine the system response for the input signal of a wave function with initial states all 0:

$$r_4(t) = 10\sin(0.1t) + 5\sin(t) + 5\cos(0.5t - \frac{\pi}{2})$$

In the next figure we can see the system response: Position ($\theta(t)$), velocity ($\dot{\theta}(t)$), error ($e_\theta(t)$) and control signal ($F(t)$).
The adaptation parameters are:

$$Kz = 200 \qquad \lambda = 20 \qquad \gamma = 10$$

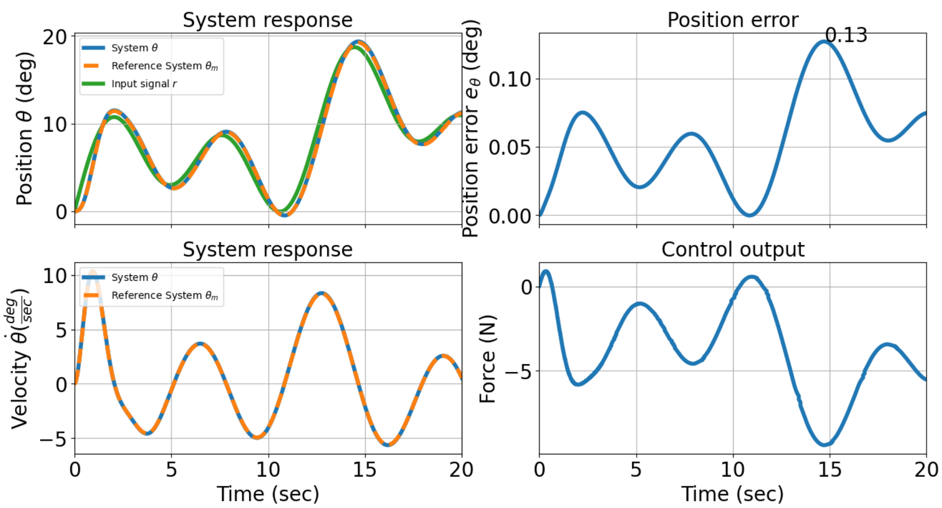


Figure 6: System response to wave input with unknown parameters

Looking at the upper left graph, we can see that at the beginning of the simulation, even though the initial state of the system and signal are identical, there is a delay in the tracing. This delay is not significant, but it is distinguishable and is present during the whole simulation time. It's also notable that the system is tracing the reference model with minimal error. From the upper right graph, we see that the max error is 0.13 (deg).

In figure 7, we can see the real parameters of the model and the convergence of the estimated parameters that are used in the controller. As mentioned previously, the parameters don't have to converge to real values for the system to track the model. Figure 8 magnifies the are were we see the convergence of the parameters from figure 7.

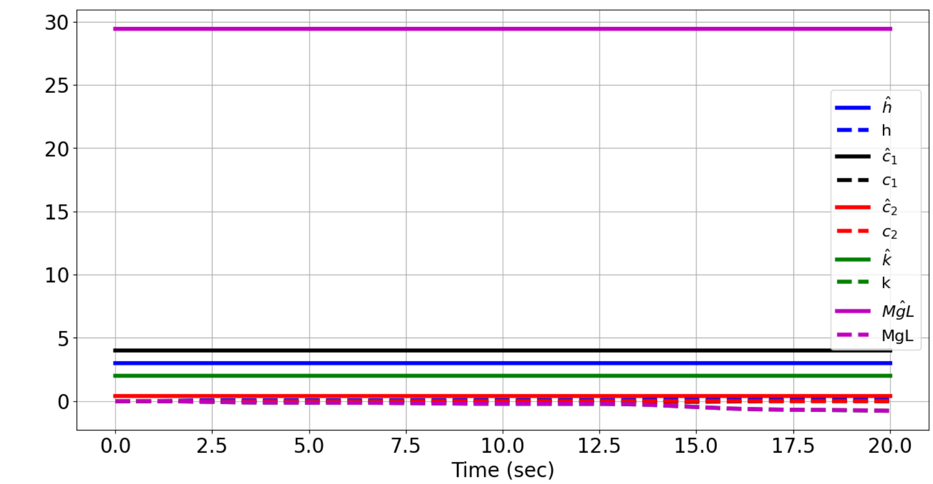


Figure 7: Adaptation of the unknown parameters

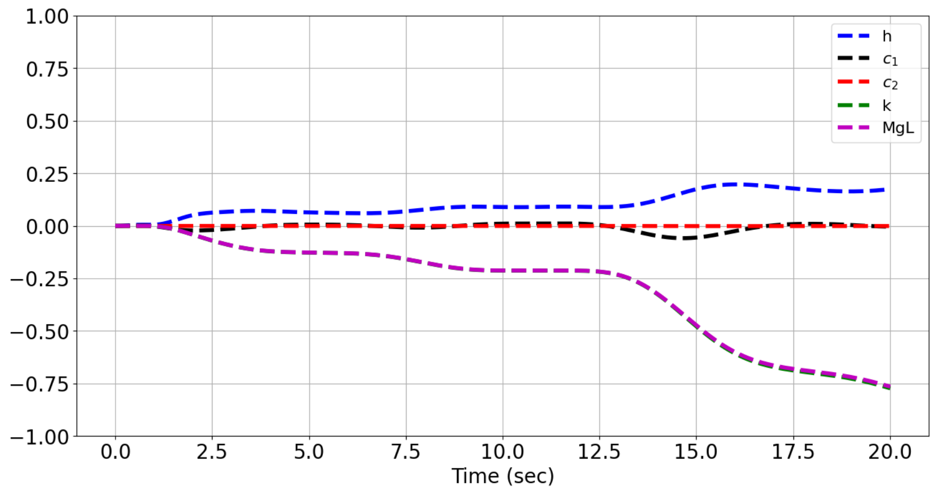


Figure 8: Adaptation of the unknown parameters - close up

To further more investigate the affect of the parameters $K_z, \gamma and \lambda$, we simulated the system with a mesh of configurations. In figures 9 - 11 we see the error and the control signal of the system.

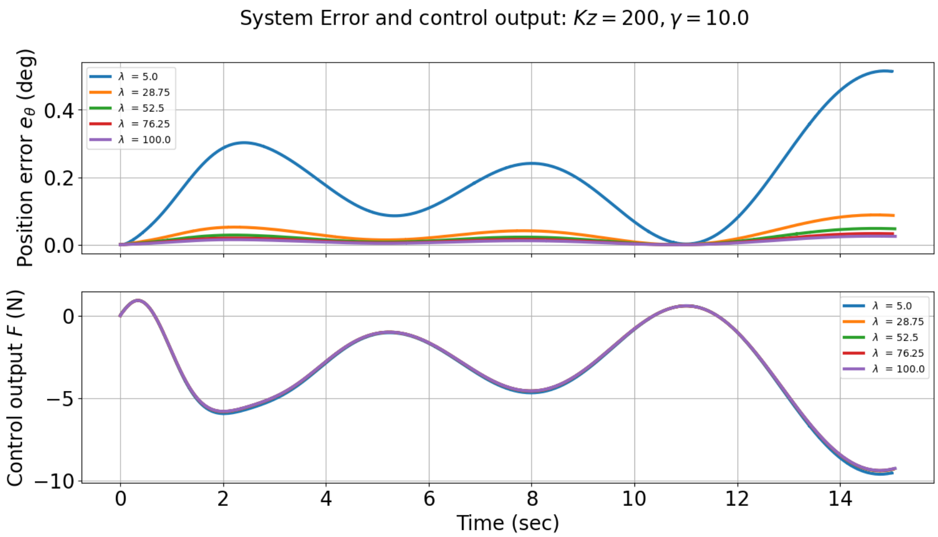


Figure 9: system response to different values of $\lambda \in [5, 28.8, 52.5, 76.3, 100]$

fix this caption

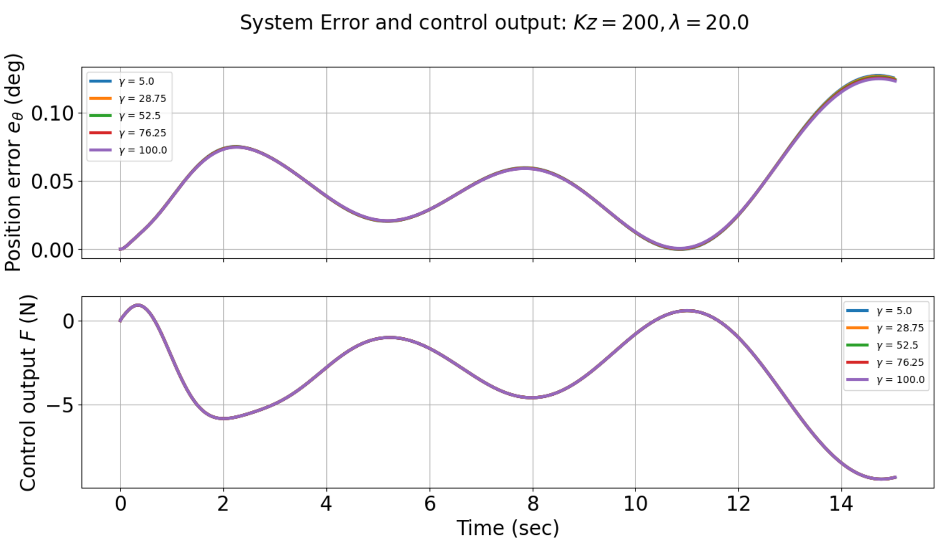


Figure 10: system response to different values of $\gamma \in [5, 28.8, 52.5, 75.3, 100]$

fix this caption

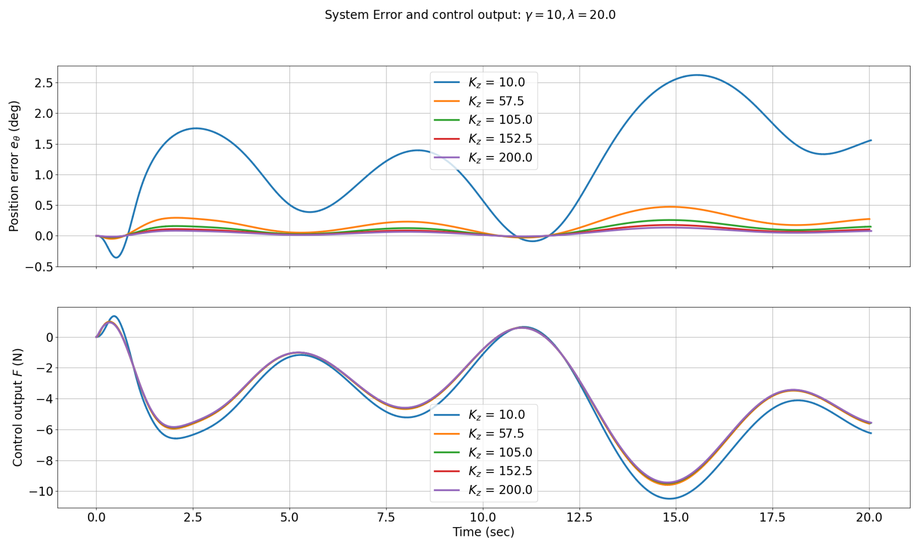


Figure 11: system response to different values of $K_z \in [10, 57.5, 105, 152.5, 200]$

What is common in all the mesh testing we did is that as we increase the parameters of the adaptations laws, system error is reduced. But its not linear, which means there is a lower limit to the error no matter of how more we will increase them.

More interesting results we will see the part 5.2 where we can see that if we will increase the adaptation laws parameters to much we will get an oscillation of the system response and even divergence.

5 Indirect digital control design

Out system is a continuous-time physical system. Its natural (not controlled) behavior is not dependent on the controller. On the other hand, the control law is designed on a micro-controller that is limited by the frequency it does calculations. In this part, we will simulate both system and controller with different frequencies. We will define the simulated system to be run with:

$$f_{system} = 1,500Hz$$

And the controller will be run on different frequencies to test the overall system response:

$$f_{controller} \in [2, 3, 4, 5, 6, 10, 12, 15, 20, 25, 30, 50, 60, 75, 100, \\ 125, 150, 250, 300, 375, 500]Hz$$

In part 3 we designed a continuous controller, 29, and a reference model, 3:

$$
\begin{aligned}
u(t) =& h\ddot{\theta}_r(t) - kz(t) + a_1\dot{\theta} + b_1\dot{\theta}^3 + a_2\theta + b_2\sin\theta \\
\ddot{\theta}_m =& -k_1\dot{\theta}_m - k_2\theta_m + k_3 r(t)
\end{aligned}
$$

To implement it on a micro-controller, we must convert the controller law and the reference model to a discrete in time. For example, using backward derivative approximation where $T(sec) = 1/f(Hz)$ is the controller's period, and K is the iteration:

Note that the simulation solution of the system is exactly as presented in part 3.

$$
\boxed{
\begin{aligned}
u(k) =& h\frac{\theta_r(k) - 2\theta_r(k-1) + \theta_r(k-2)}{T^2} - kz(t) + \\
& a_1\frac{\theta(k) - \theta(k-1)}{T} + \\
& b_1\frac{\theta(k) - \theta(k-1)}{T}^3 + \\
& a_2\theta(k) + \\
& b_2\sin\theta(k)
\end{aligned}
}
\tag{30}
$$

$$
\frac{\theta_{m,k} - 2\theta_{m,k-1} + \theta_{m,k-2}}{T^2} = -k_2\theta_{m,k} - k_1\frac{\theta_{m,k} - \theta_{m,k-1}}{T} + k_3 r(t)
$$

$$
\begin{gathered}
\theta_{m,k} - 2\theta_{m,k-1} + \theta_{m,k-2} = -k_2T^2\theta_{m,k} - k_1T(\theta_{m,k} - \theta_{m,k-1}) + k_3 r(t)T^2 \\
\theta_{m,k}\left(1 + k_2T^2 + k_1T\right) = \theta_{m,k-1}(2 + k_1T) - \theta_{m,k-2} + k_3 r(t)T^2 \\
\boxed{\theta_{m,k} = \frac{\theta_{m,k-1}(2 + k_1T) - \theta_{m,k-2} + k_3 r(t)T^2}{1 + k_2T^2 + k_1T}}
\end{gathered}
\tag{31}
$$

We also need to discretize the adaptation laws from equations 14:

$$
\boxed{
\begin{aligned}
\hat{h}(k+1) &= \left(-\gamma z(k)\ddot{\theta}_r(k)\right)T + \hat{h}(k) \\
\hat{c}_1(k+1) &= \left(-\gamma z(k)\dot{\theta}(k)\right)T + \hat{c}_1(k) \\
\hat{c}_2(k+1) &= \left(-\gamma z(k)\dot{\theta}(k)^3\right)T + \hat{c}_2(k) \\
\hat{k}(k+1) &= \left(-\gamma z(k)\theta(k)\right)T + \hat{k}(k) \\
\widehat{MgL}(k+1) &= \left(-\gamma z(k)\sin\theta(k)\right)T + \widehat{MgL}(k)
\end{aligned}
}
\tag{32}
$$

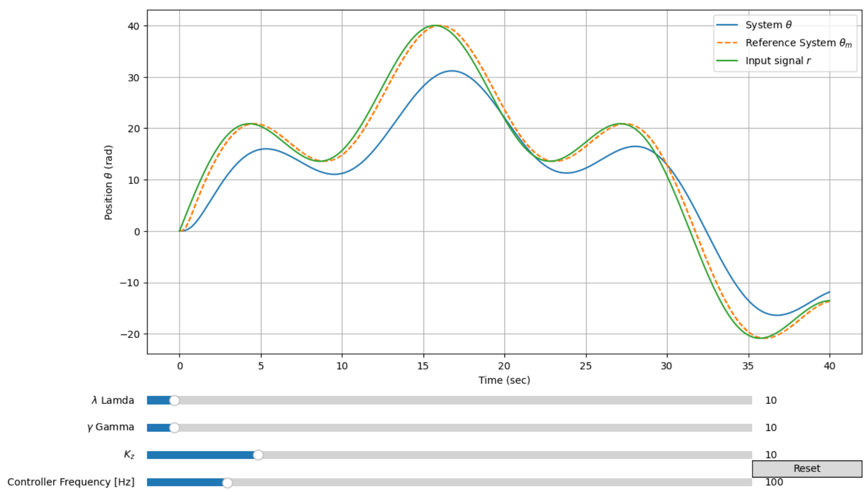


Figure 12: Discrete time controller simulation - system response

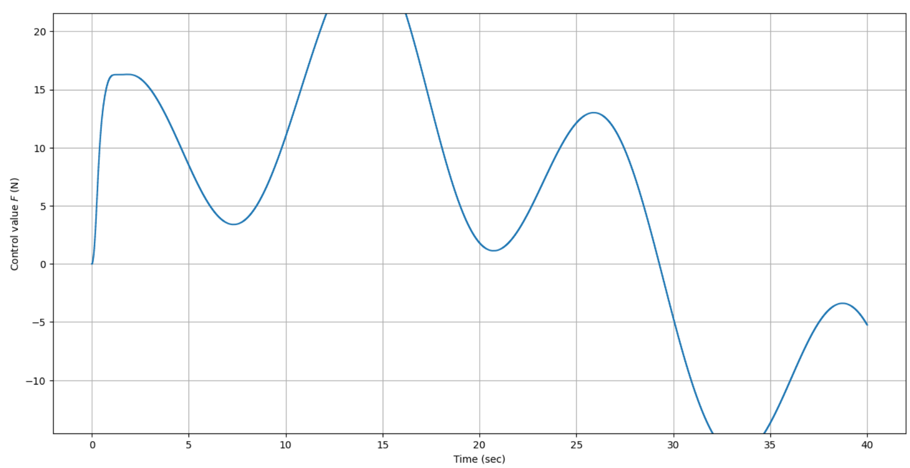


Figure 13: Discrete time controller simulation - system response

5.1 Python simulation and implementation

The implementation of the system simulation was similar to the previous parts, using *odeint*. The main difference was in the implementation of the control and reference calculation, which was written by me using the discrete equations 30-

32. Examining the graph's closeup view of the system response, control signals, and error, we can see the main differences between the discrete and continuous responses. First, we see the steps width of the responses and control signal in the graphs. As expected, as we increase control frequency f, and reduce the time steps T, we get closer to the continuous-time response.

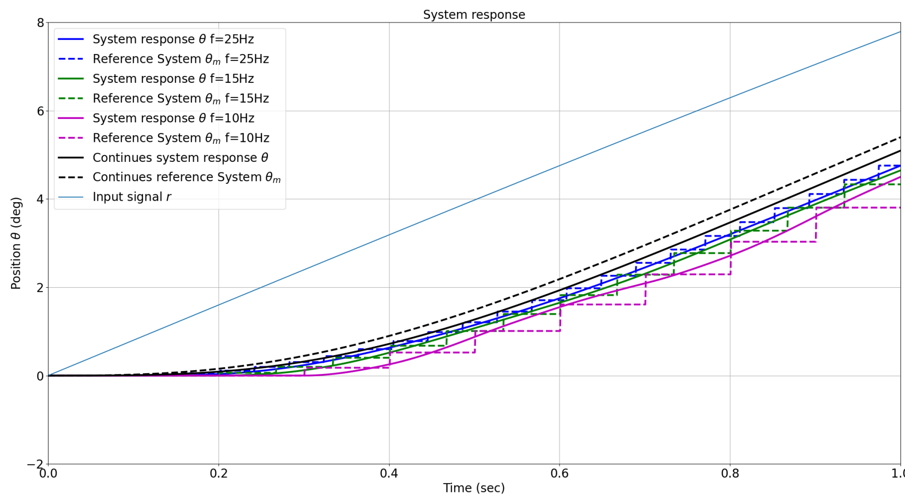


Figure 14: Discrete time controller simulation - closeup - system response

One of the most dangerous small control frequencies is that the control signal has large step size which can affect and harm the used hardware (e.g., motors).

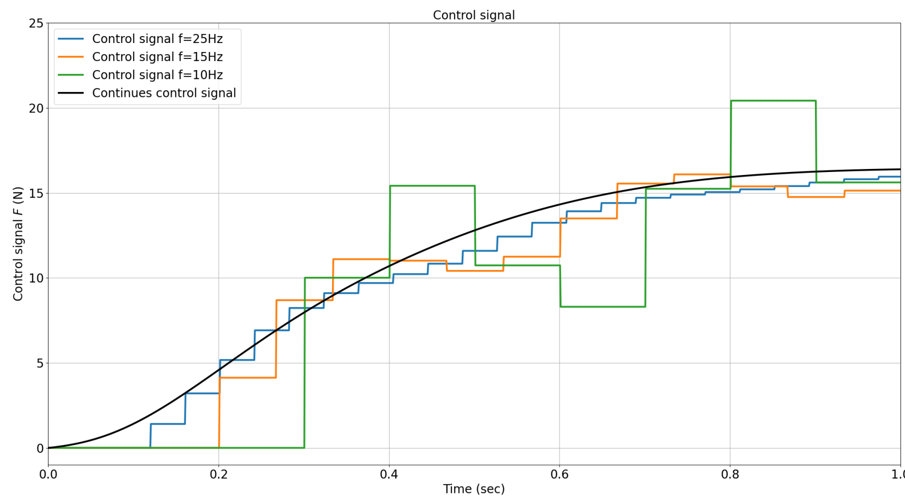


Figure 15: Discrete time controller simulation - closeup - control signal

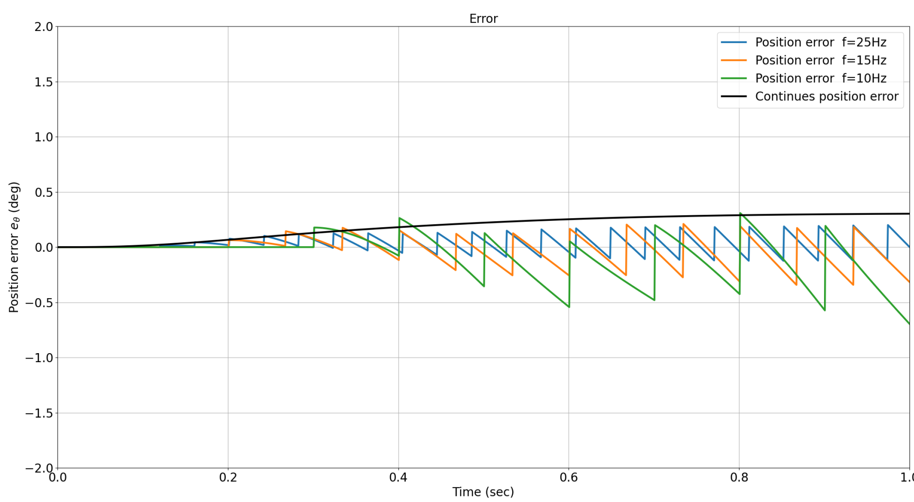
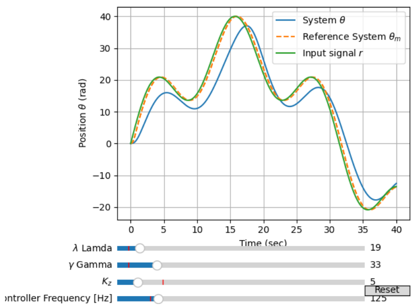
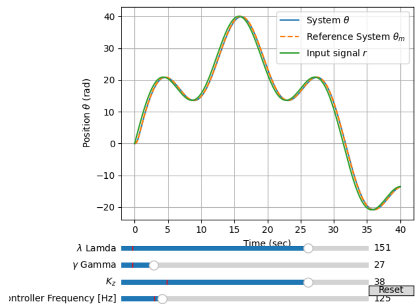


Figure 16: Discrete time controller simulation - closeup - system error

In the next figures we will examine the importance of adjusting the adaptation laws for small frequencies. The tested frequency is $f_{controller} = 125Hz$. In figures ?? and ?? we how choosing λ or K_z to small can cause a delay and reduces apliture of the system response.





More over, choosing λ or K_z to large will cause over reaction, meaning we get an oscillation of the system response and divergence of the response.

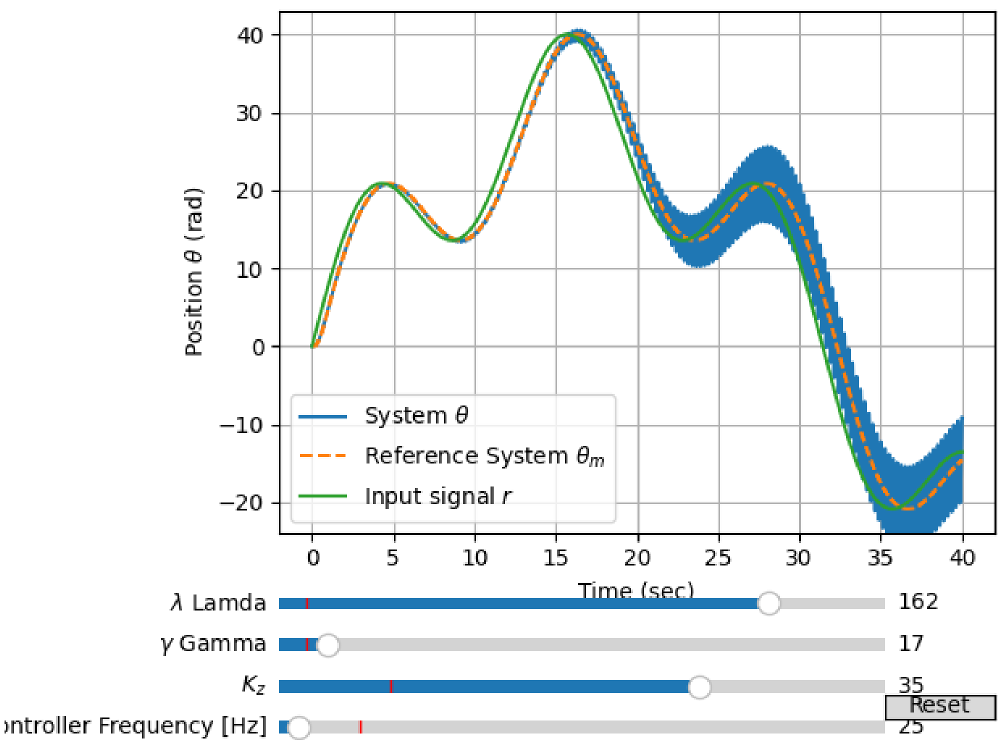


Figure 18: System response oscillation

5.2 Python advanced simulation

As discussed in the previous part, adjusting the parameters K_z, γ, and λ isn't an easy task, especially if we need to be able to get good results in a range of micro-controller frequencies. I build an interactive python GUI with several sliders to control the parameters and see the results in real-time. This small but significant GUI makes the adjusting parameters task easier.

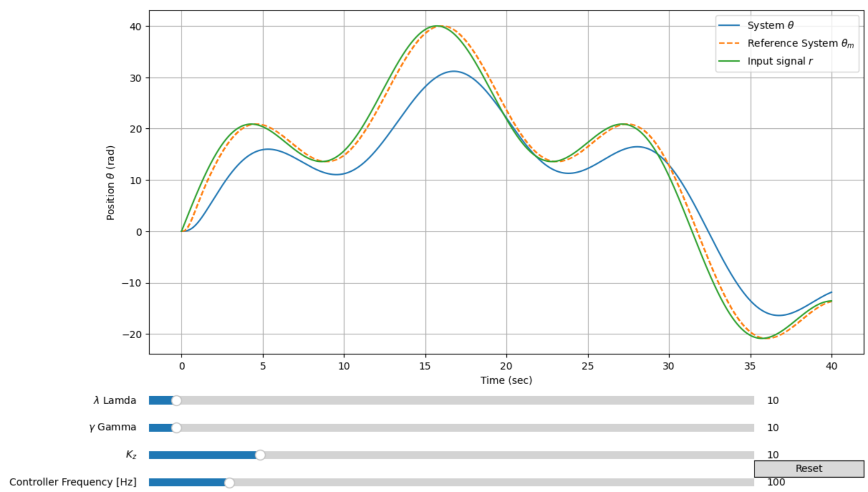


Figure 19: Python simulation with GUI - system response

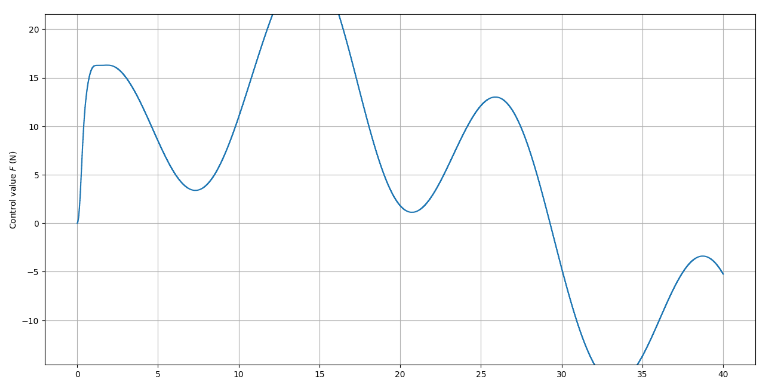


Figure 20: Python simulation with GUI - control signal

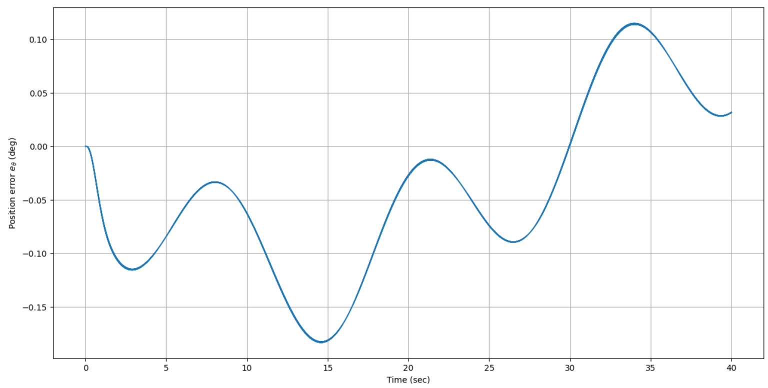


Figure 21: Python simulation with GUI - position error

6 Appendices

1. This project was written using Overleaf, the link to view this project: https://www.overleaf.com/read/qwggtkjcgbmg
2. All the home assignments and python files are available at my GitHub: github.com/shmulike/Mechatronic-systems-BGU