

Mechanical Engineering Project - Furuta Pendulum System Control

Alon Ben-David

Igal Vornov

2024-2025

Preface

This is the preface of the document. It introduces the topic and purpose.

Contents

Preface	1
1 Introduction and system modeling	4
1.1 Introduction	4
1.2 System modeling	4
1.2.1 The test system	4
1.2.2 Defining the Mechanical System	5
1.2.3 Defining the DC-motor	9
1.3 Block Diagrams of the System	9
1.3.1 Block Diagram of the whole system, under bottom eq.	9
1.3.2 Block Diagram of the system with no pendulum	11
2 System Identification	13
2.1 Lab System Structure	13
2.2 Transfer Functions and Process of Identification	14
2.2.1 Identification For system without pendulum	15
2.2.2 Identification For system with pendulum	16
3 First Control Scheme - Tracking for arm angle	18
3.1 Tracking Controller	19
3.1.1 Design of Tracking Controller using LS	19
3.1.2 Compare proportional and LS controller	19
3.2 Time-Optimal Control Based Reference Signal	20
3.2.1 Calculating the Bang-Bang Control Signal	20
3.2.2 Comparison of tracking references by Loop-Shaping controller	22
3.3 Applying the 2DOF Control Scheme on the tracking Closed-Loop	24
3.3.1 The Scheme of a 2DOF Controller	24
3.3.2 Compare Tracking of 1DOF and 2DOF Controller	24
3.4 Damping Controller	25
3.4.1 The Control Scheme of a Cascade Damping + 2DOF Tracking	26
3.4.2 Design of the Damping Controller	26
3.4.3 Compare tracking after adding the damping controller	28

3.5	Damping Controller from LQG problem	29
3.5.1	LQG problem and it's solution	30
3.5.2	Loop-Shaping using an LQG solution	30
3.5.3	Design of damping controller based on LQG	31
3.5.4	Damping controller based on LS Vs. LQG	32
3.5.5	Damping controller based on LQG using different weights	33

Chapter 1

Introduction and system modeling

1.1 Introduction

In this project, our aim is to control a system consisting of a DC motor that applies torque to a Furuta pendulum, which includes a shaft, arm, and pendulum, with the input being the motor voltage. To control it using LTI methods, a linear model is required, so both the mechanical system and the brushed DC motor will be modeled as one linear model. Since the Furuta Pendulum is non-linear, it must be linearized around an equilibrium point. The project will involve modeling, linearization, parameter identification, and designing a control strategy for implementation.

1.2 System modeling

1.2.1 The test system

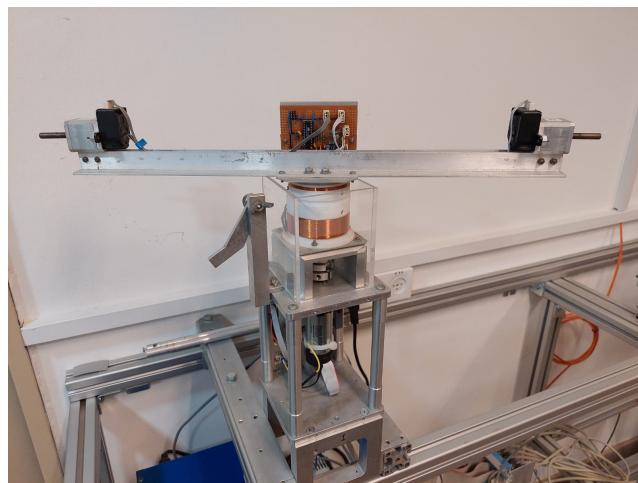


Figure 1.1: The Lab System

The system in Figure 1.1 is a Furuta pendulum, consisting of a DC motor connected, via a gearbox, to a rigid vertical shaft with a horizontal shaft above it. Also, to the end of the arm a rigid pendulum is connected. The

only input of the system is the Voltage given to the motor.

The system has 2 encoders we use: the encoder near the motor for the motor's shaft angle, and an encoder near the pendulum for reading its rotation angle.

While the movement of the arm alone is linear, the movement of the pendulum, and thus the whole system, is not. In order to control the system using linear control tools, a linearized model will be used.

1.2.2 Defining the Mechanical System

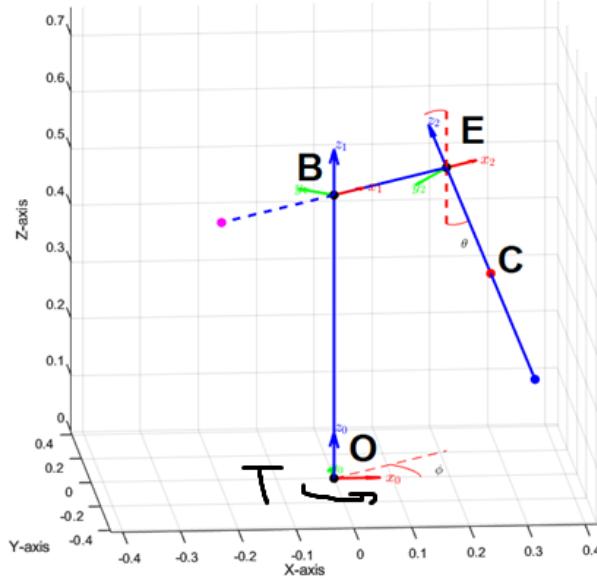


Figure 1.2: Model of the Mechanical System

The system in Figure 1.2 consists of a horizontal beam with a rigidly connected vertical beam, called the arm, which can rotate at an angle ϕ as one rigid body. A pendulum is attached to a free-rotation axis at the end of the arm, rotating with angle θ . A DC motor applies torque $M(t)[Nm]$ (**FIX DRAWING! M and not T**) in the vertical direction to rotate both the arm and the pendulum. The motor rotates the moment of inertia of the gearbox in addition to that of the arm and pendulum.

Parameters and Signals

The system has the following parameters:

- $r[m]$ - The distance from the axis BO to point E
- $l_c[m]$ - The distance from point E to the COM of the pendulum, point C
- $m_p[kg]$ - Mass of the pendulum.
- $J_a[kg \cdot m^2]$ - Moment of Inertia of the arm (rigidly connected vertical and horizontal beam) and gearbox, around BO axis.

- $J_p [kg \cdot m^2]$ - Moment of Inertia of the pendulum, around its COM with axis perpendicular to BE.
- $f_a, f_p [J/rad]$ - Damping coefficients in the rotation of the arm and pendulum, separately.

The following presents the systems signals and signals used while obtaining the systems equations with Euler-Lagrange equations:

- $\phi(t)$ - Angle of rotation of the arm.
- $\theta(t)$ - Angle of rotation of the pendulum.
- $q(t) = \begin{pmatrix} q_1(t) \\ q_2(t) \end{pmatrix} = \begin{pmatrix} \phi(t) \\ \theta(t) \end{pmatrix}$ - Column vector of the mechanical systems DOF.
- $T(t)$ - Total kinetic energy (KE) of the system, where $T_a(t)$ and $T_p(t)$ are the contributions of the arm and pendulum separately.
- $V(t)$ - Total potential energy (PE) of the system, where $V = V_a + V_p$ similarly to the KE.
- $L(t) = T(t) - V(t)$ - The Lagrangian of the system.
- $D(t)$ - The dissipation function of the system, expresses the contribution of damping forces in the energy equations.
- $Q(t) = \begin{pmatrix} Q_1(t) \\ Q_2(t) \end{pmatrix}$ - Column vector of the vector of generalized forces on the system.

Axis-Systems

The coordinate systems are defined as follows: the system (x_0, y_0, z_0) is static and located at the base of the shaft. The system (x_1, y_1, z_1) is attached to the arm and rotates by an angle ϕ around the $+z_0$ axis, while the system (x_2, y_2, z_2) is attached to the pendulum and rotates by an angle θ around the $-x_1$ axis.

With this system definition, the inertia matrix of the pendulum is given by:

$$I_{(c,2)} = \begin{bmatrix} J_p & 0 & 0 \\ 0 & J_p & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Where the inertia is being calculated around the COM of the pendulum, in an axis-system with the same orientation as axis-system 2.

Obtaining non-linear Equations

To derive the equations of motion for the mechanical system, we use the Euler-Lagrange method for analyzing physical systems:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} = Q_i, \quad i = \{1, 2\} \quad (1.1)$$

To acquire the 2 equations that solve for the angles of the arm and pendulum, for input $M(t)$ and IC, we need to find the KE and PE - $T(t)$ and $V(t)$.

After getting the energy's as a function of the system parameters and angles, we substitute them in equation 1.1 and get this 2 non-linear equations of motion:

$$\begin{cases} (1.2.1) \quad J_a \ddot{\phi} + 2 \left(\frac{1}{2} J_p + \frac{1}{2} m_p l_c^2 \right) (2 \sin(\theta) \cos(\theta) \dot{\theta} \dot{\phi} + \sin^2(\theta) 2 \ddot{\theta}) \\ \quad + \frac{1}{2} m_p (r^2 2 \ddot{\phi} - 2 r \dot{\theta} l_c \cos(\theta) + 2 r \dot{\theta}^2 l_c \sin(\theta)) \\ \quad + f_a \dot{\phi} = M(t) \\ (1.2.2) \quad \left(\frac{1}{2} J_p + \frac{1}{2} m_p l_c^2 \right) (2 \ddot{\theta}) - \frac{1}{2} m_p (2 r \ddot{\phi} l_c \cos(\theta) - 2 r \dot{\phi} \dot{\theta} l_c \sin(\theta)) \\ \quad - \left(\frac{1}{2} J_p + \frac{1}{2} m_p l_c^2 \right) (2 \sin(\theta) \cos(\theta) \dot{\phi}^2) \\ \quad - \frac{1}{2} m_p (2 r \dot{\theta} \dot{\phi} l_c \sin(\theta)) + m_p g l_c \sin(\theta) + f_p \dot{\theta} = 0 \end{cases} \quad (1.2)$$

Description in State Space and Equilibrium Points

To control the system using linear-control tools, we need to linearize the system around a chosen equilibrium point. We define the following state vector for the system:

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Using x we can get the equations of the system in a state-space representation.

Firstly, we can rewrite the equations in 1.2 like:

$$\begin{bmatrix} J_a + 2(J_p + m_p l_c^2) \sin^2(\theta) + m_p r^2 & -m_p r l_c \cos(\theta) \\ -m_p r l_c \cos(\theta) & J_p + m_p l_c^2 \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

Where $g_1 = M - f_a \dot{\phi} - 2 \left(\frac{1}{2} J_p + \frac{1}{2} m_p l_c^2 \right) 2 \sin(\theta) \cos(\theta) \dot{\theta} \dot{\phi} - \frac{1}{2} m_p 2 r \dot{\theta}^2 l_c \sin(\theta)$
and $g_2 = \left(\frac{1}{2} J_p + \frac{1}{2} m_p l_c^2 \right) (2 \sin(\theta) \cos(\theta) \dot{\phi}^2) + \frac{1}{2} m_p (2 r \dot{\theta} \dot{\phi} l_c \sin(\theta)) - m_p g l_c \sin(\theta) - f_p \dot{\theta} - \frac{1}{2} m_p 2 r \dot{\phi} \dot{\theta} l_c \sin(\theta)$.
This representation of the equations can be written as:

$$M_f \dot{x} = N_f$$

Where:

$$M_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & J_a + 2(J_p + m_p l_c^2) \sin^2(\theta) + m_p r^2 & -m_p r l_c \cos(\theta) \\ 0 & 0 & -m_p r l_c \cos(\theta) & J_p + m_p l_c^2 \end{bmatrix}$$

And:

$$N_f = \begin{bmatrix} x_3 \\ x_4 \\ g_3(x) \end{bmatrix}$$

Such that:

$$g_3(x) = \left(\frac{1}{2}J_p + \frac{1}{2}m_p l_c^2 \right) (2\sin(x_2)\cos(x_2)x_3^2) \\ + \frac{1}{2}m_p (2rx_3x_4l_C \sin(x_2)) - m_p g l_c \sin(x_2) - f_p x_4 - \frac{1}{2}m_p 2rx_3x_4l_C \sin(x_2)$$

Thus $\dot{x} = M_f^{-1}N_f$ and the representation of the system in state-space:

$$\begin{cases} \dot{x} = f(x, M(t)) = M_f^{-1}N_f \\ y(t) = h(x, M(t)) = Cx = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \phi \\ \theta \end{pmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{cases} \quad (1.3)$$

Now, to get the equilibrium points, we need to find the points (x_{eq}, M_{eq}) s.t.:

$$\dot{x}_{(x,M)=(x_{eq},M_{eq})} = f(x_{eq}, M_{eq}) = 0$$

And after solving the linear equations we get:

$$\begin{pmatrix} x \\ M \end{pmatrix}_{eq} = \begin{pmatrix} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \\ M \end{pmatrix}_{eq} = \begin{pmatrix} \phi_0 \\ \pi k \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \forall \phi_0 \in \mathbb{R}, k \in \mathbb{Z}$$

In the physical system, this corresponds to 2 unique equilibrium points:

- Bottom eq. where ϕ is arbitrary and $\theta = 0$, meaning the pendulum is pointing downwards.
- Top eq. where ϕ is arbitrary and $\theta = \pm\pi$, meaning the pendulum is pointing upwards.

Bottom linearization of the system

Now that we have the non-linear equations and the eq. points, we can get the linear system from the torque $M(t)$ to the angles with deviation from the bottom eq. ($x_\delta = x - x_{eq} = x$):

$$\dot{x} = Ax + BM$$

Where:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_p^2 gr l_c^2}{\alpha} & -\frac{f_a(J_p+m_p l_c^2)}{\alpha} & -\frac{f_p m_p r l_c}{\alpha} \\ 0 & -\frac{m_p g l_c (J_a+m_p r^2)}{\alpha} & -\frac{f_a m_p r l_c}{\alpha} & -\frac{f_p (J_a+m_p r^2)}{\alpha} \end{pmatrix}$$

And:

$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{J_p+m_p l_c^2}{\alpha} \\ \frac{m_p r l_c}{\alpha} \end{pmatrix}$$

Where $\alpha \equiv (J_a + m_p r^2)(J_p + m_p l_c^2) - m_p^2 r^2 l_c^2$.

After solving this state equation, we can get the angles directly from the components of the state vector.

In addition, it can be shown that this LTI system is equivalent to the following 2 Linear ODEs:

$$\begin{cases} (J_a + \frac{1}{2}m_p(2r^2))\ddot{\phi} + \frac{1}{2}m_p(-2rl_c)\ddot{\theta} + f_a\dot{\phi} = M \\ \frac{1}{2}m_p(-2rl_c)\ddot{\phi} + (\frac{1}{2}J_p + \frac{1}{2}m_p l_c^2)2\ddot{\theta} + f_p\dot{\theta} + m_p g l_c \theta = 0 \end{cases} \quad (1.4)$$

1.2.3 Defining the DC-motor

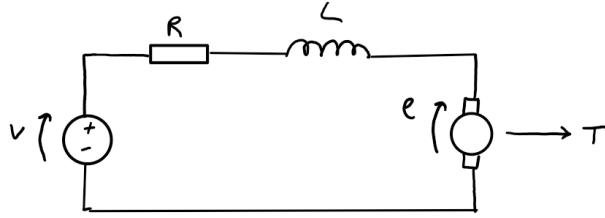


Figure 1.3: Model of DC motor

The figure in 1.3 depicts the model of the actuator used in the project - a Brushed DC motor. The model is of a linear electrical circuit that has input voltage $V(t)$, Resistance and Inductance of the motor R, L respectively and the output torque and current is $M_m(t), I(t)$ respectively. (**FIX, NOT $T(t)$!**)

Let us label the parameters:

- $R[\Omega]$ - Resistance of the rotor and the stator.
- $L[H]$ - Inductance of wires in the stator.
- N_g - Gear-ratio of gearbox.
- $K_e \left[\frac{V \cdot s}{rad} \right]$ - Motor back EMF constant.
- $K_t \left[\frac{N \cdot m}{A} \right]$ - Motor torque constant.

Where the applied voltage is $V(t)[V]$, the motor current $I(t)[A]$ and the torque and angle output of the motor are $M_m(t)[Nm], \phi_m(t)$, respectively.

The equations of the DC motor are known and they are:

$$\begin{cases} RI(t) = V(t) - K\dot{\phi}_m(t) \\ M_m(t) = KI(t) \\ \frac{M(t)}{M_m(t)} = \frac{\dot{\phi}_m(t)}{\dot{\phi}(t)} = N_g \end{cases} \quad (1.5)$$

1.3 Block Diagrams of the System

1.3.1 Block Diagram of the whole system, under bottom eq.

Now that we have the ODEs of the motor and the linearized mechanical system, we can get the block diagram from the input voltage $V(t)$ to the output angles $\phi(t)$ and $\theta(t)$.

To obtain the block diagram of the system, let us first rearrange equations 1.4 and 1.5:

Firstly, we use the Laplace Transform on the second equation in 1.4 and get:

$$\frac{\theta(s)}{\phi(s)} = \frac{m_p r l_c s^2}{(J_p + m_p l_c^2)s^2 + f_p s + m_p g l_c} := P_{\theta\phi}(s) \quad (1.6)$$

And after using the Laplace Transform on the first equation, we can get:

$$s\phi(s) = \frac{1}{(J_a + m_p r^2)s + f_a} (M(s) + m_p r l_c s^2 \theta(s)) \quad (1.7)$$

Before drawing the block diagram, we also need the equations from the motor. From the voltage equality in 1.5, we get:

$$I(s) = \frac{V(s) - K s \phi_m(s)}{R} \quad (1.8)$$

And from the torque equation, while also using the gear ration, we get:

$$M(s) = N_g M_m(s) = N_g K I(s) \quad (1.9)$$

From these four equations (1.6, 1.7, 1.8, 1.9), we can derive the block diagram of the system, which describes the relation between the voltage input $V(t)$ and the angles $\phi(t)$, $\theta(t)$. The resulting block diagram is shown in Figure 1.4.

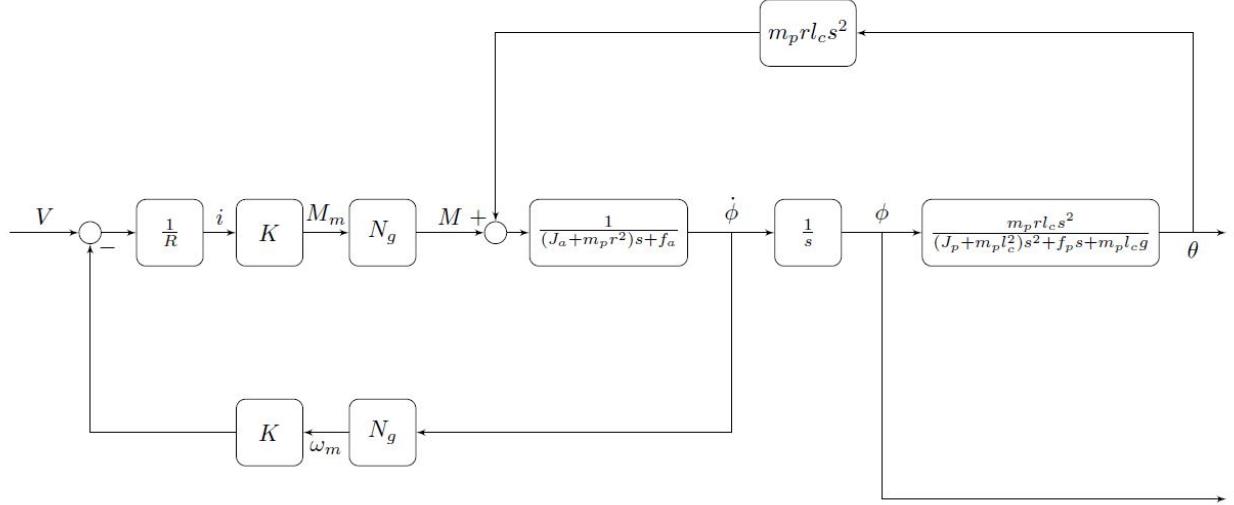


Figure 1.4: Block Diagram of the system under bottom equilibrium

Using algebraic manipulation in the s -domain, we can derive the process from the voltage input to the arm angle:

$$P_{\phi V}(s) = \frac{N_1(s)}{D_1(s)} \quad (1.10)$$

where:

$$N_1(s) = \frac{K_T N_g}{R} ((J_p + m_p l_c^2)s^2 + f_p s + K_a m_p l_c g) \quad (1.11)$$

$$\begin{aligned}
D_1(s) = & \left((J_a + m_p r^2)(J_p + m_p l_c^2) - (m_p r l_c)^2 \right) s^4 \\
& + \left((J_a + m_p r^2)f_p + (J_p + m_p l_c^2)f_a + \frac{KN_g}{R} KN_g (J_p + m_p l_c^2) \right) s^3 \\
& + \left((J_a + m_p r^2)m_p l_c g + f_a f_p + \frac{KN_g}{R} KN_g f_p \right) s^2 \\
& + \left(f_a m_p l_c g + \frac{KN_g}{R} KN_g m_p l_c g \right) s
\end{aligned} \tag{1.12}$$

Notice the integrator in the system.

Now we can simplify the block diagram for our control purposes. The simplified Block Diagram of the system is shown in figure 1.5.

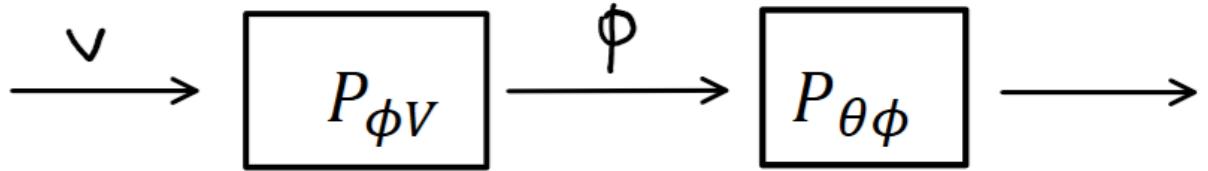


Figure 1.5: Simplified Block Diagram of the System

1.3.2 Block Diagram of the system with no pendulum

In addition, if we remove the pendulum from the system, the equations of the mechanical system change into one simple equation:

$$J_a \ddot{\phi}(t) + f_a \dot{\phi}(t) = M(t) \xrightarrow{\mathcal{L}} s\phi(s) = \frac{1}{J_a s + f_a} M(s) \tag{1.13}$$

From this equation and the motor equations, we can get the simplified block-diagram shown in figure 1.6.

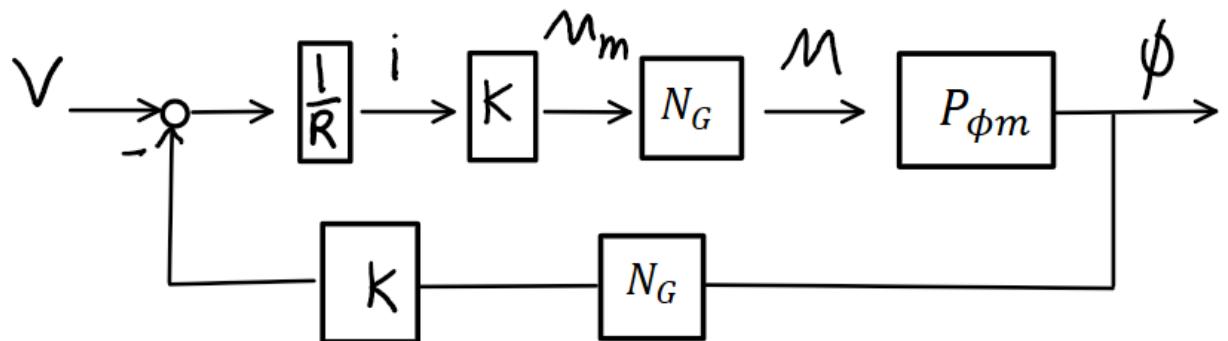


Figure 1.6: Block Diagram of the System without pendulum

Where $P_{\phi M}^{np}(s) = \frac{1}{J_a s + f_a}$ is the process from the torque applied on the mechanical system to the arm angle, when there is no pendulum. The symbol "np" will be used to indicate process in the no-pendulum case.

In this case, the process from the voltage to the arm angle:

$$P_{\phi V}^{np}(s) = \frac{\frac{1}{R} K N_g}{s(J_a s + (f_a + \frac{1}{R} (K N_g)^2))} \quad (1.14)$$

Notice the integrator in the system.

Now that we have the transfer functions of the system, we can go to the system identification step.

Chapter 2

System Identification

2.1 Lab System Structure

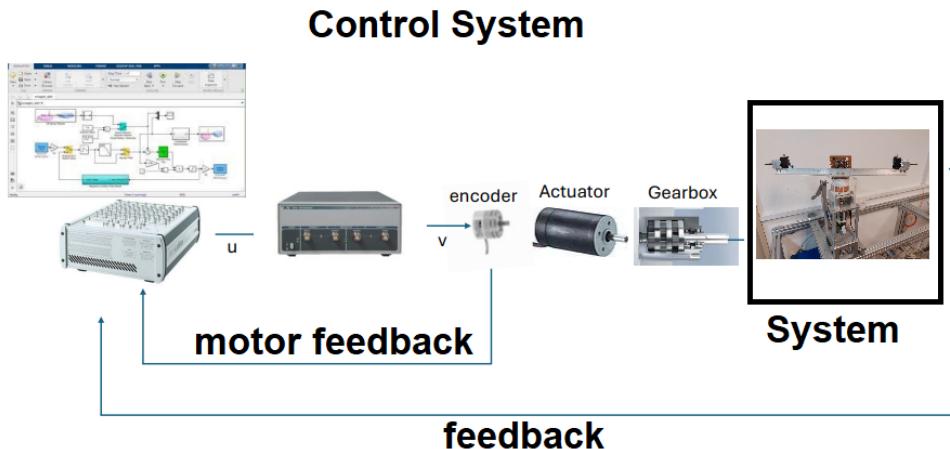


Figure 2.1: The Control System Whole Structure

As demonstrated in figure 2.1, the lab setup consists of the mechanical system driven by a DC motor at its base. We implement the control scheme in Simulink and compile the code using the dSPACE interface. During operation, the computer activates the dSPACE box, which sends a voltage signal to a 12x voltage amplifier. The amplified signal $V(t)$ is then applied to the motor.

Because of that 12 factor, in the Simulink scheme we divide by it in order to use the voltage applied on the motor as our control signal.

The encoders on the system measure the pendulum angle $\theta(t)$ deviating from initial rotation, and the motor shaft angle $\phi_m(t)$. Their measurements are in degrees and we work with a model that treats them in radians. Thus, we convert the angles into radians using the $-\pi/180$ factor for our control purposes. It is negative in order to get the voltage connection in the direction specified in the model.

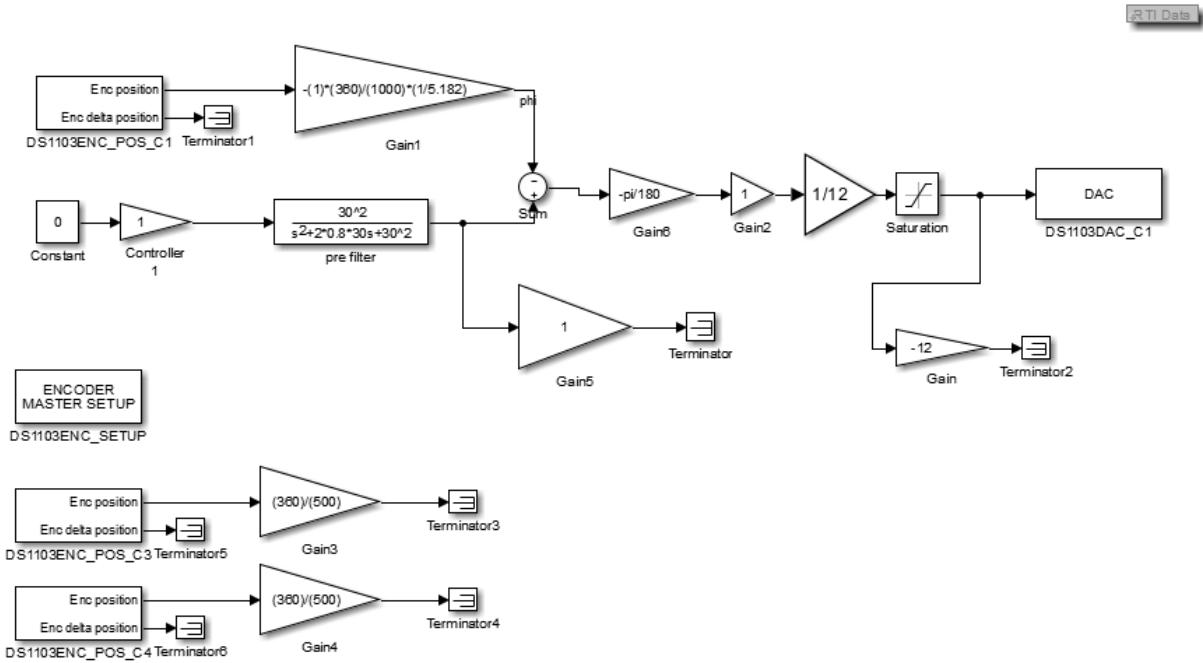


Figure 2.2: The First Simulink File

In the Lab, we design the control scheme using a Simulink file on Matlab like in figure 2.2, which shows a control scheme of negative-unity feedback on $\phi(t)$ with $C(s) = 1$ used on the system. We use blocks that represent the readings from the encoders of the arm angle $\phi(t)$ and the pendulum angle $\theta(t)$.

In the Simulink file we make the control scheme we wish to implement and send the voltage command to the dSPACE box.

The Simulink has a saturation block that bounds the voltage amplitude on the motor in:

$$-12[V] < V(t) < 12[V]$$

From rotating the arm manually and seeing the readings on the encoders, we got that $N_g = 5.182$. We used this gear-ratio on the Simulink file to get $\phi(t)$ instead of $\phi_m(t)$.

2.2 Transfer Functions and Process of Identification

Before making our first control scheme, we need to find this 3 transfer functions:

1. $P_{\phi}V(s)$ - The TF from $V(t)$ to $\phi(t)$ for system with pendulum, under bottom equilibrium
2. $P_{\theta\phi}(s)$ - The TF from $\phi(t)$ to $\theta(t)$ for system with pendulum, under bottom equilibrium
3. $P_{\phi}^{np}(s)$ - The TF from $V(t)$ to $\phi(t)$ for system **without** pendulum, under bottom equilibrium

From the system's model, we see that the TF's from $V(t)$ to $\phi(t)$ are not stable in both cases, because they have an integrator in them. Therefore, it is not advised to apply a step voltage and use the measured arm angle to

identify the TF's, because in this scenario the angle will increase without a bound and it may be dangerous. Thus, we will use a simple control configuration of negative unity feedback that can stabilize both system's as shown in figure 2.3. On 2.3a we have the control configuration for when the system has no pendulum, and on 2.3b for the full system.

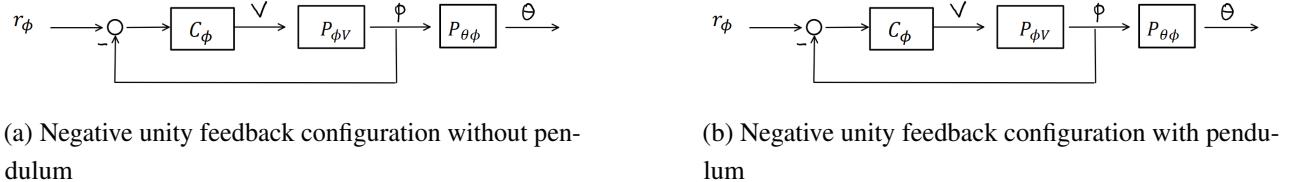


Figure 2.3: System identification control schemes with and without a pendulum.

In both cases, we will use this control scheme with a proportional controller $C_\phi(s) = k_p$. We will start with low gain in order to see that the closed loop is stable, than increase the gain as much as possible in order to perform system identification when the closed loop is getting close to instability.

We were shown that "bad controllers" tend to give good system identification results.

Then, from the known input and measured output, we get with the SI tool the TF from the reference $r_\phi(t)$ to $\phi(t)$, $T_\phi(s)$, and than we can extract the plant with:

$$T_\phi(s) = \frac{L_\phi(s)}{1 + L_\phi(s)} \Rightarrow P_{\phi V}(s) = \frac{T_\phi(s)}{1 - T_\phi(s)} \frac{1}{C_\phi(s)}$$

where $L_\phi(s) = P_{\phi V}(s)C_\phi(s)$.

On the contrary, because the TF between the angles $P_{\theta\phi}(s)$ is stable, we can get it straight from measuring the input and output, when doing the second experiment, with no additional control loop.

2.2.1 Identification For system without pendulum

Firstly, we closed a negative unity feedback loop on the system without the pendulum, as shown in the control scheme on 2.3a. We applied a step reference that was filtered using:

$$F(s) = \frac{30^2}{s^2 + 2 * 0.8 * 30s + 30^2} \quad (2.1)$$

Using the system identification toolbox on Matlab, we got that:

$$\widetilde{P}_{\phi V}^{np}(s) = \frac{2.978}{s(s + 1.316)} \quad (2.2)$$

The figure in 2.4 compares the simulation's response to the measured system's behavior with the same reference input. The simulation output resembles that of a 2nd-order undamped system, while the real system behaves similarly but only up to a small interval of angles before steady-state. Beyond that, Coulomb friction, which is non-linear and not modeled in the system's dynamics, causes the real angle to stop.

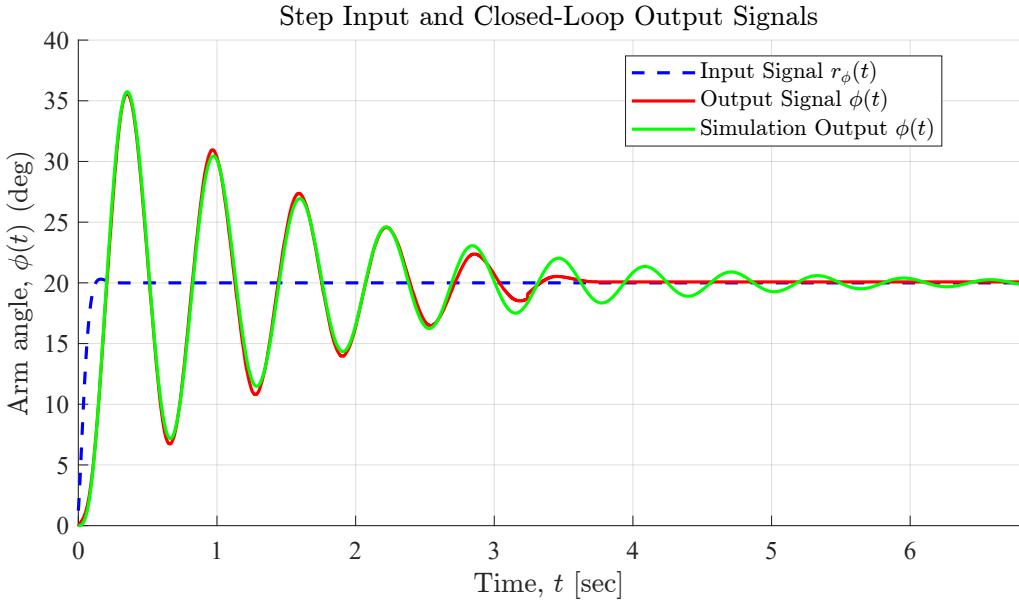
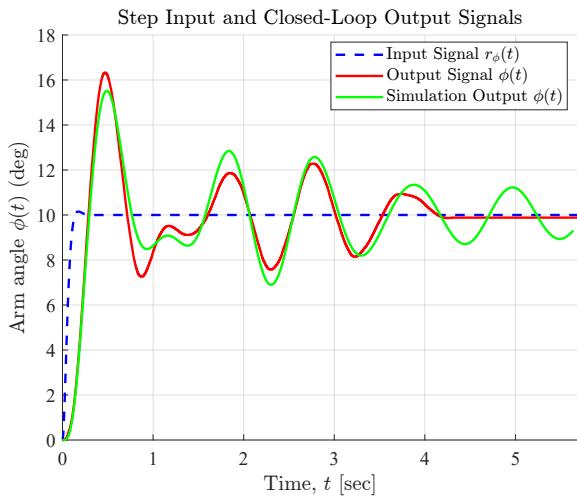


Figure 2.4: Shows $\phi(t)$ output from Lab and Simulation (with identified TF), for system without pendulum.

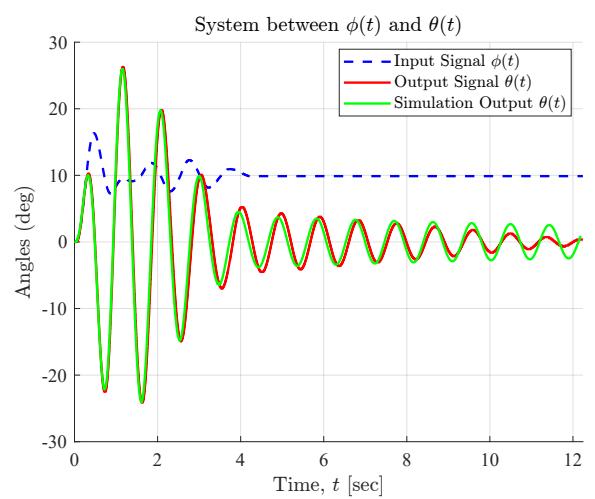
2.2.2 Identification For system with pendulum

Now, we close a loop on the system, as shown in the right of figure 2.3b. The left sub-figure shows $\phi(t)$ comparison and the right sub-figure shows $\theta(t)$ comparison. We apply a step reference filtered similarly to the previous experiment. Using the SI toolbox, we get:

$$(\text{Unused}) \widetilde{P_{\phi V}}(s) = \frac{2.568s^2 + 0.5587s + 111.2}{s(s^3 + 2.269s^2 + 49.12s + 83.56)}, \widetilde{P_{\theta\phi}}(s) = \frac{1.261s^2}{s^2 + 0.115s + 45.3} \quad (2.3)$$



(a) $\phi(t)$ output from Lab and Simulation with identified TF



(b) $\theta(t)$ output from Lab and Simulation with identified TF

Figure 2.5: Comparison of system identification performance, for system with pendulum

In figure 2.5 we can see the comparision between the output of the lab and SI between $r_\phi(t) \rightarrow \phi(t)$ and between $\phi(t) \rightarrow \theta(t)$.

The difference in the outputs in figure 2.5a, when simulating a step response, is because the SI also includes IC and we didn't add that. After that experiment, we tuned the coefficients and got this TF:

$$\widetilde{P_{\phi V}}(s) = \frac{3.08s^2 + 1.219s + 137.1}{s(s^3 + 1.451s^2 + 49.01s + 50.72)} \quad (2.4)$$

In conclusion, the TF's we will use for making our first control schemes will be: that in equation 2.2, the **second** on eq. 2.3 and from eq. 2.4.

Chapter 3

First Control Scheme - Tracking for arm angle

We will make our first control scheme, as shown in figure 3.1, in order to have tracking of the arm angle $\phi(t)$ after a desired reference signal $r_\phi(t)$.

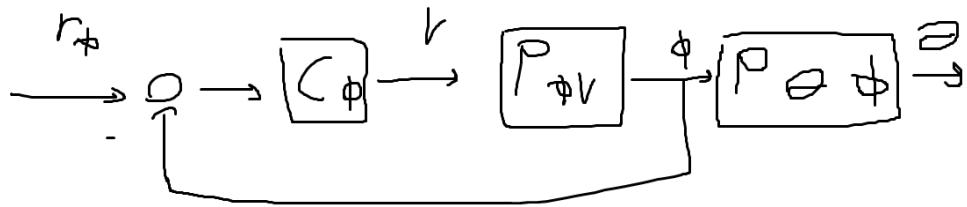


Figure 3.1: Unity-Feedback Control Scheme For Tracking of $\phi(t)$

In this chapter, we will increasingly modify the scheme in order to improve the tracking of the arm angle, while reducing the required control input $V(t)$.

We require the following from the closed-loop:

1. closed-loop stability
2. zero steady-state error for static reference $r_\phi(t) = 1(t)$
3. zero steady-state error for static disturbance $d(t) = 1(t)$
4. Adequate stability margins
5. The crossover frequency ω_c is treated as a tuning parameter

In the Lab itself we will input a step reference filtered through the filter $F(s)$ from equation 2.1.

3.1 Tracking Controller

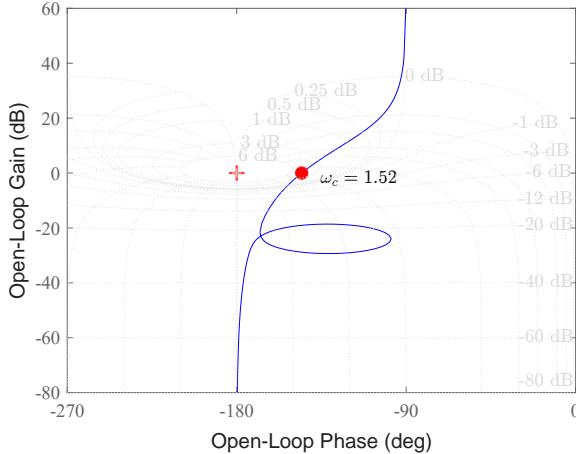
3.1.1 Design of Tracking Controller using LS

Firstly, we designed the tracking controller $C_\phi(s)$ for the control scheme in figure 3.1.

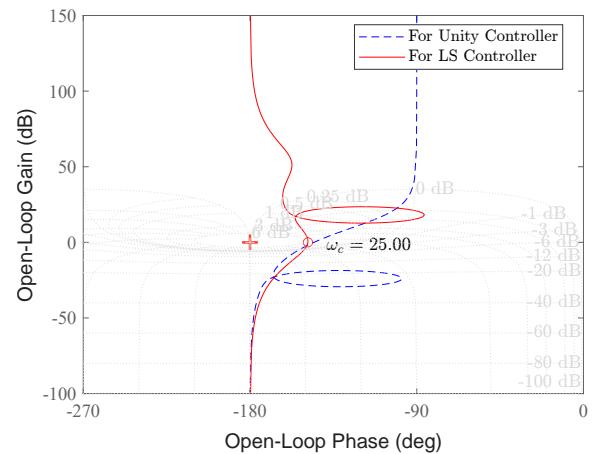
The original OL has $\omega_c = 1.52[\frac{\text{rad}}{\text{s}}]$. We designed a controller that has the following components: a gain to set $\omega_c = 25[\frac{\text{rad}}{\text{s}}]$, a lag controller with an integrator and a lead controller to set the PM. The resulting controller:

$$C_\phi(s) = \frac{349.3s^2 + 5217s + 2521}{s(s + 43.3)} \quad (3.1)$$

Closed loop stability is ensured through the Nichols diagram, zero error for static reference is obtained automatically from the integrator in $P_{\phi V}(s)$ and we added the integrator in $C_\phi(s)$ to reject static disturbances. The Open-Loop of the Process $P_{\phi V}(s)$ itself with $C_\phi(s) = 1$ and with our designed $C_\phi(s)$ is shown in figure 3.2.



(a) Open-Loop $L_\phi(s)$ with Unity-Controller $C_\phi(s) = 1$

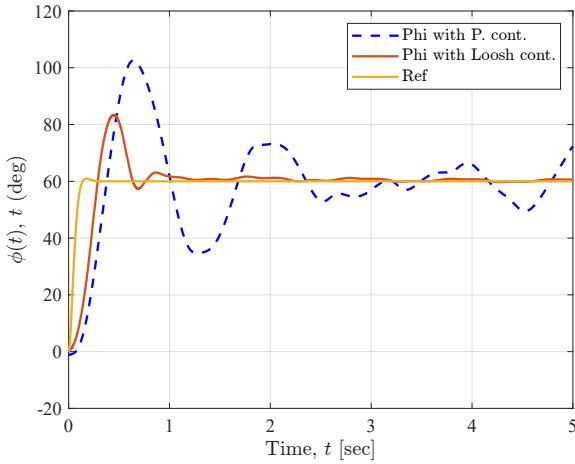


(b) Open-Loop $L_\phi(s)$ with $C_\phi(s)$ from Loop-Shaping

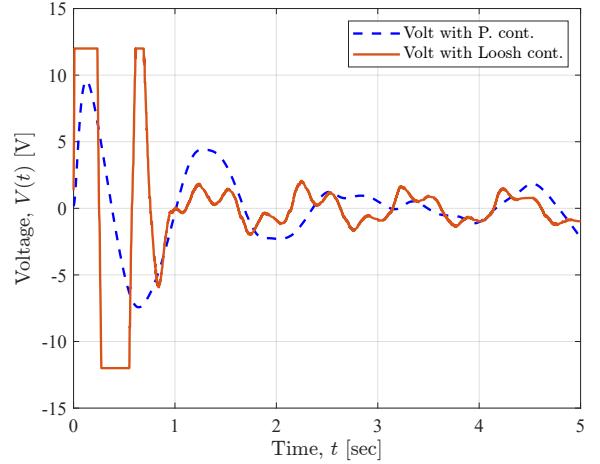
Figure 3.2: Comparison of Open-Loop with/without Loop-Shaping Controller

3.1.2 Compare proportional and LS controller

Now lets compare the performance of the Closed-Loop on tracking a step reference filtered through $F(s)$, between a constant controller and the Loop-Shaping controller.



(a) Comparison of tracking controllers: $C_\phi(s) = 10$ and $C_\phi(s)$ from Loop-Shaping



(b) Comparison of Voltage: $C_\phi(s) = 10$ and $C_\phi(s)$ from Loop-Shaping

Figure 3.3: Comparison of Closed-Loop performance with/without Loop-Shaping Controller

We can see that in figure 3.3a the LS controller better tracks the desired trajectory, with a smaller OS and a shorter convergence time. Even so, from 3.3b we see that the required control effort while using the LS controller is reaching saturation, unlike with the gain controller of 10.

Saturation is undesired due to it being non-linear and deviating from the designed linear-control behavior. Thus, we would like a **reference that is less demanding** of our closed-loop system.

3.2 Time-Optimal Control Based Reference Signal

In the previous section, we observed that a lightly-filtered step signal might be a difficult reference for the system to track.

In 1.10 we can see that the process $P_{\phi V}(s)$ is strictly proper and thus represents an inertial system. Hence, a step reference is physically unattainable by the system.

In this section we formulate a reference signal that requires a specific control signal from the system that is optimal for minimum time under symmetric boundary of the voltage.

3.2.1 Calculating the Bang-Bang Control Signal

Firstly, we formulate the problem.

Consider the system from the Voltage to the Arm angle:

$$P_{\phi V}(s) = \frac{\phi(s)}{V(s)}$$

We would like to find the voltage $V(t)$ that would take the angle $\phi(t)$ from ϕ_0 to ϕ_f under minimum time, assuming initial rest ($\dot{\phi}(t = 0) = 0$), for the condition of $|V(t)| \leq V_{max}$.

From Time-Optimal Control theory, we have these results for optimal $V(t)$:

1. Its non-zero for $0 < t < t_f$
2. It only applies maximum voltages $\pm V_{max}$
3. Number of switching required, for system with only real poles, is $n - 1$, where n is the systems order.
The voltage beginning and end switching from/to zero are not counted.

We see that because the system $P_{\phi V}(s)$ is of order 4, we need 3 voltage switching. Thus, we have five unknowns: initial voltage sign, switching times, and final time t_f . This problem might become challenging. As a result, we will simplify the problem by designing the optimal voltage from the assumption that the system behaves similarly to $P_{\phi V}^{np}(s)$ in eq. 1.14.

In this case, when also comparing it to the identified process in 2.2:

$$P_{\phi V}(s) \approx P_{\phi V}^{np}(s) = \frac{\frac{1}{R} K N_g}{s(J_a s + (f_a + \frac{1}{R} (K N_g)^2))} = \frac{2.978}{s(s+1.316)} = \frac{k}{s(\tau s + 1)}$$

Where: $\tau = 0.76$, $k = 2.263$

Meaning, the optimal voltage signal looks like:

$$V_{opt}(t) = \begin{cases} V_1 & 0 < t < t_{sw} \\ -V_1 & t_{sw} < t < t_f \\ 0 & t_f < t \end{cases}$$

Where t_{sw} is the switching time between 0 and t_f , and V_1 is the initial voltage with $|V_1| = V_{max}$ but with unknown sign.

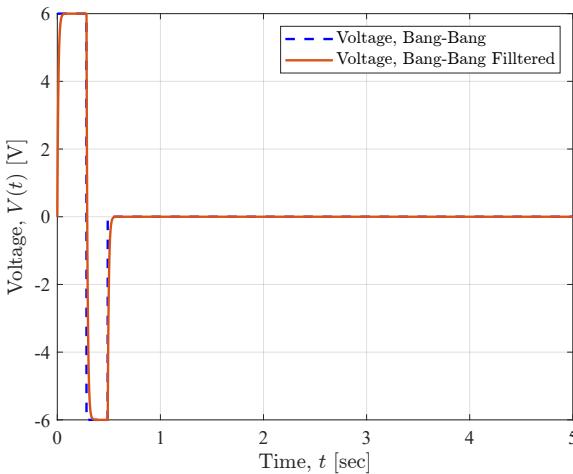
Now, to get the optimal voltage we need to find 3 unknowns: the initial voltage sign and the two switching times t_{sw} , t_f . This problem is less challenging than for the full system. By requiring that the boundary conditions on $\phi(t)$ apply and that the response of the angle is finite (via implementing FIR requirement), we get the following formulas for the switching times and initial voltage sign:

$$\left\{ \begin{array}{l} \text{sign}(V_1) = \text{sign}(\phi_f - \phi_0) \\ t_{sw} = \frac{|\phi_f - \phi_0|}{kV_{max}} + \tau \ln \left(1 + \sqrt{1 - e^{-\frac{|\phi_f - \phi_0|}{\tau k V_{max}}}} \right) \\ t_f = \frac{|\phi_f - \phi_0|}{kV_{max}} + 2\tau \ln \left(1 + \sqrt{1 - e^{-\frac{|\phi_f - \phi_0|}{\tau k V_{max}}}} \right) \end{array} \right. \quad (3.2)$$

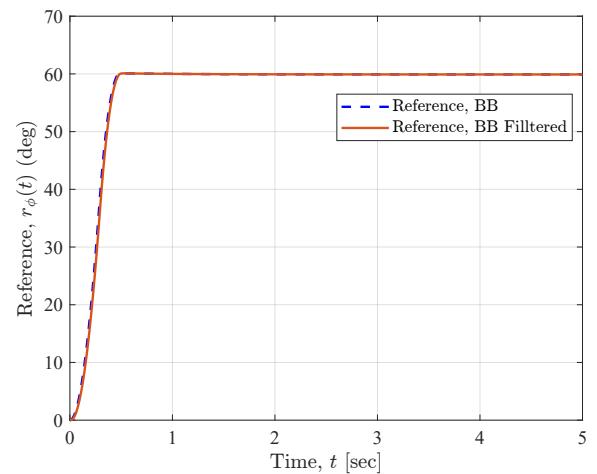
Now that we have the optimal voltage, we can get the arm angle reference from the process:

$$r_\phi(t) = P_{\phi V}\{V_{opt}(t)\}$$

And to make the optimal voltage less discontinuous, we filter it through: $F_{BB}(s) = \frac{\omega_0}{s+\omega_0}$, where $\omega_0 = 100[\frac{rad}{s}]$. For example, the optimal voltage and reference signal for $\phi_0 = 0, \phi_f = 60^\circ$ is in Figure 3.4.



(a) Shows $V_{opt}(t)$ and the filtered one, for $\phi_0 = 0$, $\phi_f = 60^\circ$ and $V_{max} = 6[V]$.



(b) Shows $r_\phi(t)$ and the filtered one, for $\phi_0 = 0$, $\phi_f = 60^\circ$ and $V_{max} = 6[V]$.

Figure 3.4: Comparison of Optimal Voltage and Reference ϕ for raw and filtered

We can observe that our filter on $V_{opt}(t)$ doesn't change the signal much.

3.2.2 Comparison of tracking references by Loop-Shaping controller

Firstly, we implement the control law as seen on figures 3.5.

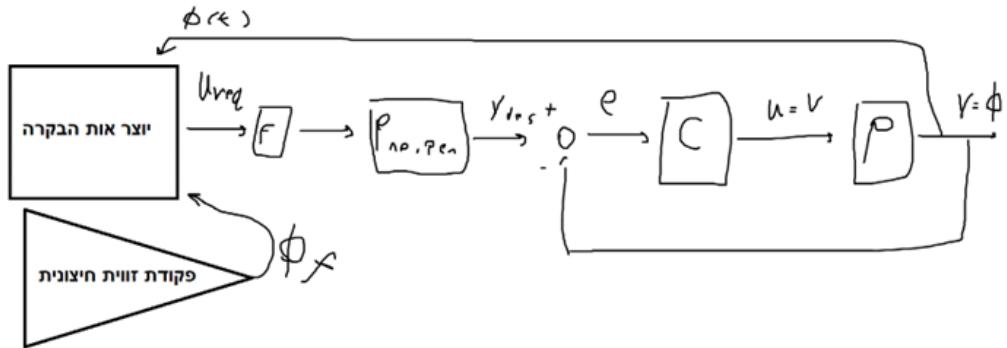
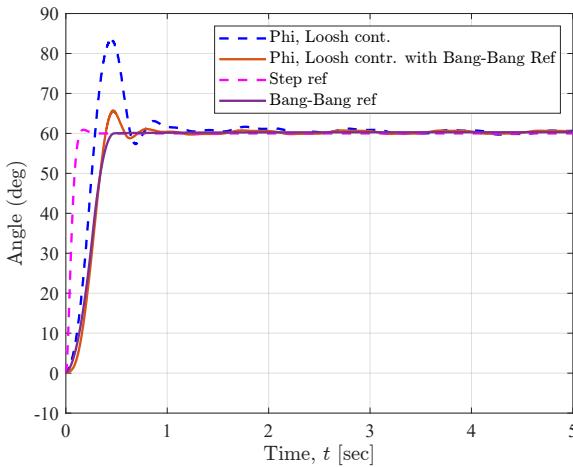
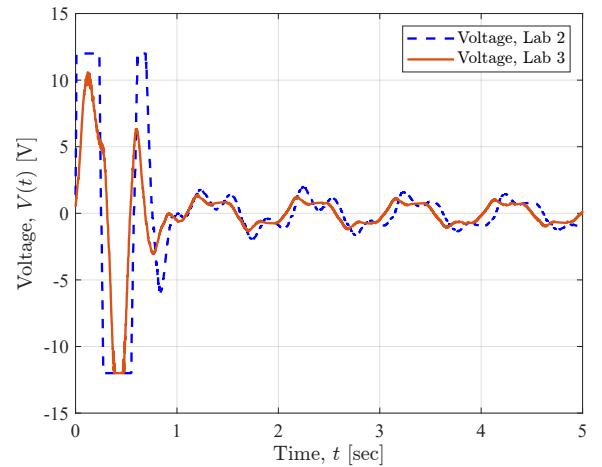


Figure 3.5: Tracking Closed-Loop with time-optimal reference, for 1DOF scheme. We see that the code calculates the required voltage, $V_{opt}(t)$, filterers it and gives the optimal trajectory $r_\phi(t)$ to the unity-feedback closed-loop.

Now we can observe the quality of the controller in the lab. In figure 3.6 we can see that the time-optimal based reference requires less voltage from the system and the tracking, for the same ϕ_f and controller, is better with less OS and smaller convergence time.



(a) Shows LS controller tracking $r_\phi(t) = 60^\circ 1(t)$ and Bang-Bang reference, in the lab



(b) Shows $V(t)$ in both cases of the reference signal, in the lab

Figure 3.6: Comparison of Optimal Voltage and Reference ϕ for raw and filtered

But, if we compare the voltage signal that the actuator applied in the lab, under optimal reference, and the optimal voltage, as seen in figure 3.7, we see that the tracking wasn't good enough in order for the actual applied voltage to be like the optimal one. The applied voltage doesn't respect the boundary's in the optimal voltage, contradicting our desire to bound the voltage below saturation.

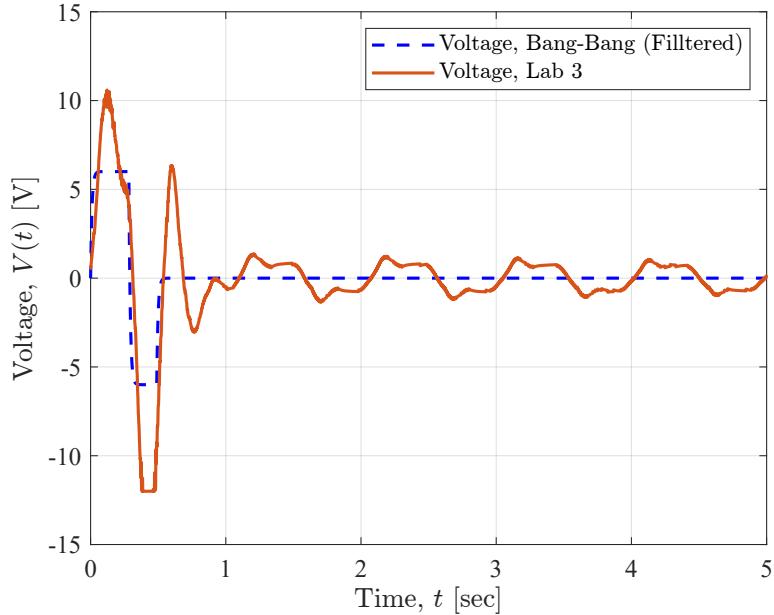


Figure 3.7: Comparing $V(t)$ applied in the lab under optimal reference w. $\phi_f = 60^\circ$ and the optimal voltage.

Consequently, we would like to modify the control scheme to improve the tracking.

3.3 Applying the 2DOF Control Scheme on the tracking Closed-Loop

In the previous section, We came to the conclusion that we need to improve the tracking of the reference, in order to ensure bounded and time-optimal control signal.

3.3.1 The Scheme of a 2DOF Controller

The scheme that we apply on the lab for this section is as in figure 3.8.

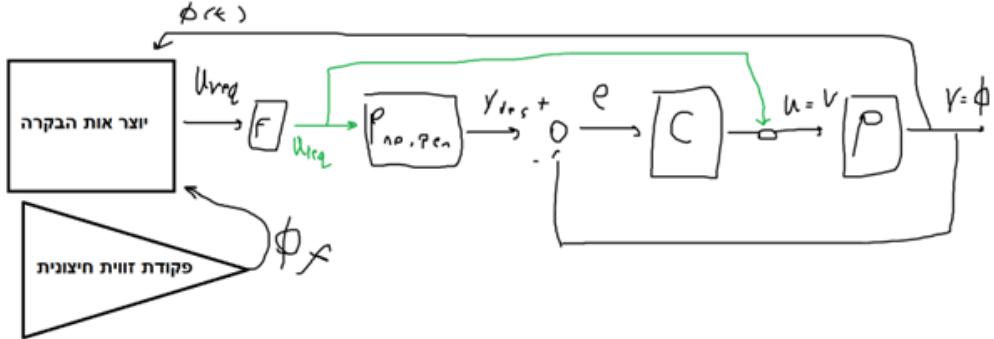


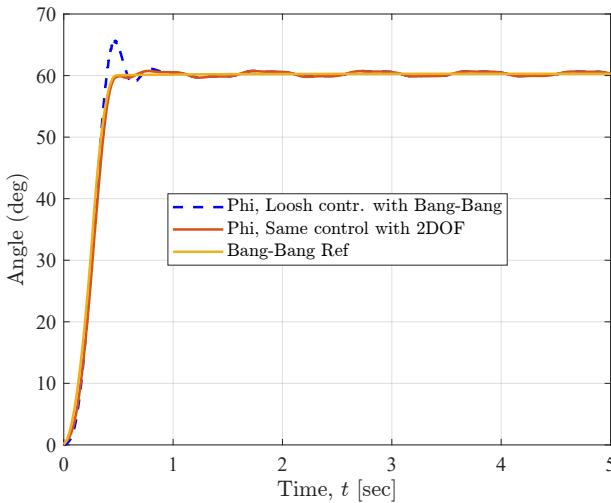
Figure 3.8: Tracking Closed-Loop with time-optimal reference, for 2DOF scheme. We see that the code calculates the required voltage, $V_{opt}(t)$, filterers it and gives the optimal trajectory $r_\phi(t)$ to the unity-feedback closed-loop, while also feed-forwarding the required voltage into the applied voltage on the system.

The motivation for using a 2DOF scheme is as follows:

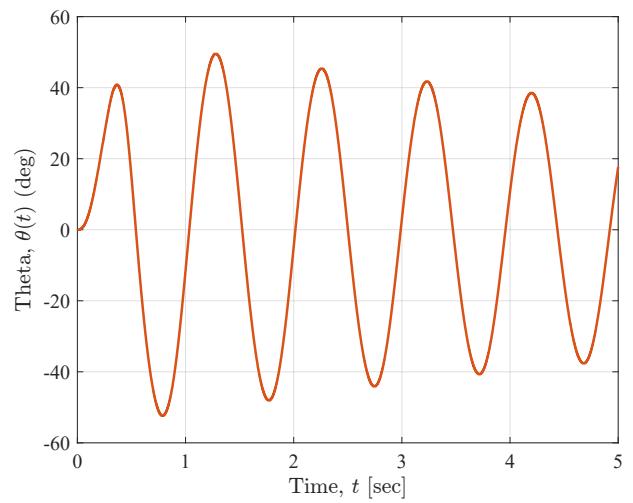
If $P_{\phi V}(s) = P_{\phi V}^{np}(s)$, and we feed-forward a control signal like in the scheme, we are guaranteed the optimal reference and voltage, meaning $\phi(t) = r_\phi(t)$ and $V(t) = V_{opt}(t)$. But, because the systems are not equal, the mismatch between the systems is dealt with by the CL in the 2DOF scheme, and we get the output and control signal to be similar to the desired signals.

3.3.2 Compare Tracking of 1DOF and 2DOF Controller

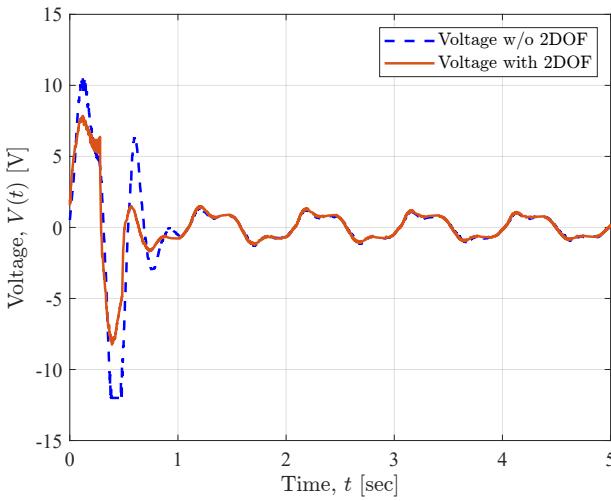
In figure 3.9, we can see that the tracking improves while using the 2DOF scheme, the voltage now looks more similar to the optimal one, as seen in 3.9d. However, the angle $\theta(t)$ has large resonance in this experiment and as a result the arm doesn't fully rest when reaching close to the target angle. The closed loop tries to damp those osculations but they do appear slightly in $\phi(t)$ on 3.9a.



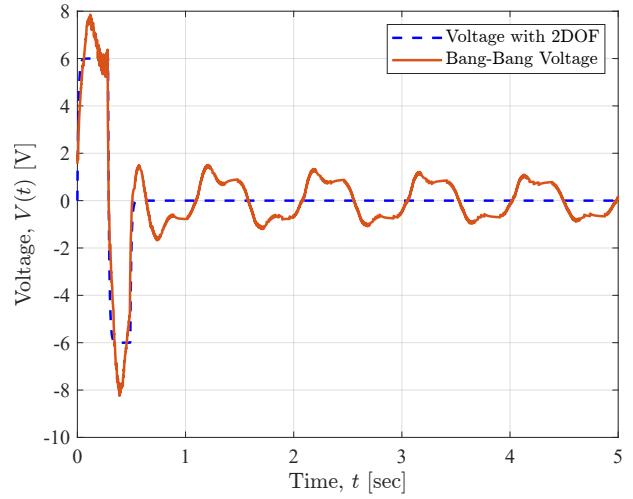
(a) Tracking performance: 1DOF vs. 2DOF LS controller.



(b) Response of the angle $\theta(t)$ for the 2DOF case.



(c) Comparison of $V(t)$ for 1DOF and 2DOF controllers.



(d) Comparison of $V(t)$ between Time-Optimal and 2DOF control.

Figure 3.9: Performance analysis of LS controller tracking a Bang-Bang reference for $\phi_0 = 0$, $\phi_f = 60^\circ$, and $V_{max} = 6[V]$. The subfigures show tracking performance between 1DOF and 2DOF, voltage application, and θ response in the 2DOF scheme.

In this case, we would like to damp the pendulum's osculations to accelerate the resting of $\phi(t)$ after the optimal voltage is applied.

3.4 Damping Controller

Now we will design the damping controller, on a closed-loop on top of the tracking closed-loop, with Loop-Shaping tools.

3.4.1 The Control Scheme of a Cascade Damping + 2DOF Tracking

The design of the cascade scheme is pragmatic and its intention is to provide a practical solution to our control problem - both tracking the desired arm angle $r_\phi(t)$ and damping the pendulum oscillations in $\theta(t)$, while only having one control input on the system.

In figure 3.10 we can see the scheme we will work with in the lab. This scheme adds another loop on top of the designed tracking CL, the outside CL with a feedback controller $C_\theta(s)$ is tasked with damping the angle $\theta(t)$ as much as possible without requiring too much voltage from the actuator, while also not interfering much with the tracking process. Adding the 2DOF structure doesn't change the process of designing the damping controller $C_\theta(s)$.

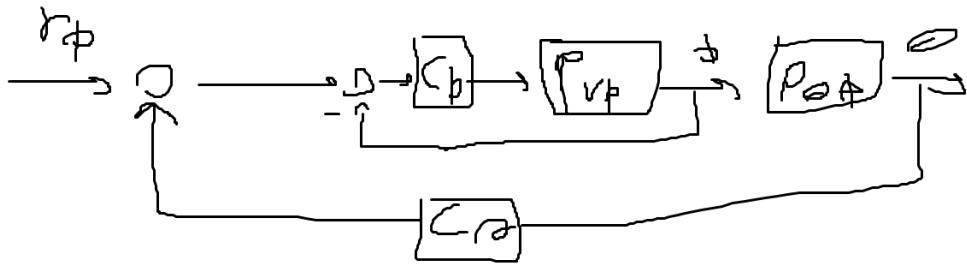


Figure 3.10: Scheme of tracking and damping closed-loop, for 1DOF tracking loop case.

3.4.2 Design of the Damping Controller

In the design of $C_\theta(s)$, we will consider the Cascade-Control scheme of 1DOF Tracking + Damping. Again, the addition of the 2DOF only improves tracking and doesn't change initial design. In figure 3.11 we see that for this case, the damping controller has the CL TF $T_{\phi r_\phi}(s)$ and $P_{\theta \phi}(s)$ to control.

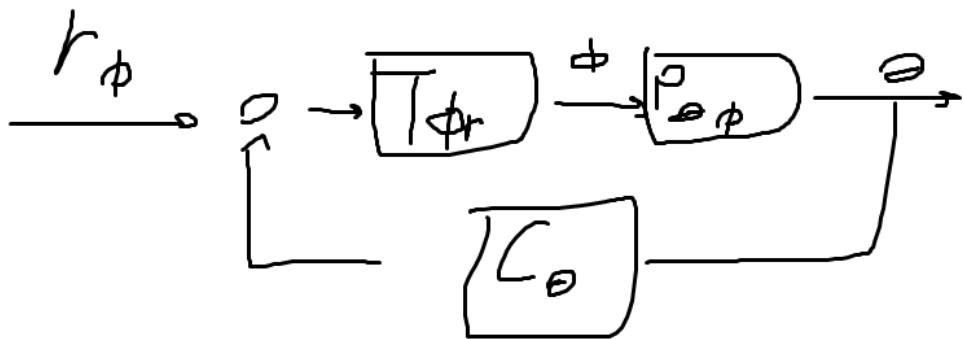


Figure 3.11: Simplified Scheme of tracking and damping closed-loop, for 1DOF tracking loop case.

We require the following from the damping closed-loop:

1. closed-loop stability
2. Adequate stability margins

3. The crossover frequency's $\omega_{c,1}, \omega_{c,2}$ are treated as a tuning parameter

Firstly, we take the tracking controller we designed $C_\phi(s)$ and get the tracking-loop closed-loop TF $T_{\phi r_\phi}(s) = \frac{P_{\phi V}(s)C_\phi(s)}{1+P_{\phi V}(s)C_\phi(s)}$. Let's denote $P_\theta(s) = T_{\phi r_\phi}(s)P_{\theta\phi}(s)$. So, from the block diagram, the CL TF from $r_\phi(t)$ to $\theta(t)$:

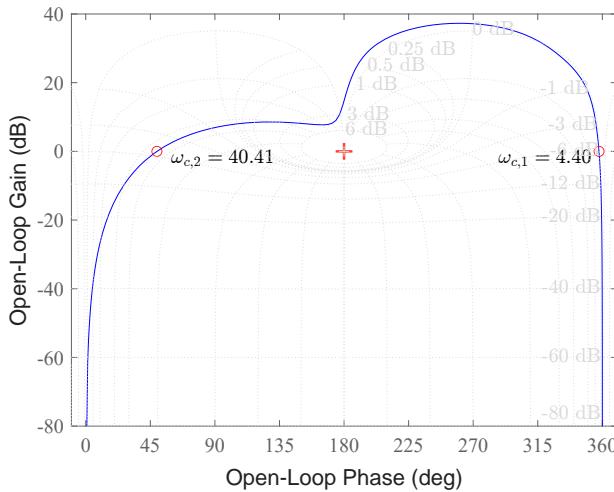
$$\theta = P_\theta(r_\phi + C_\theta\theta) \rightarrow \theta = \frac{P_\theta}{1 - P_\theta C_\theta}r_\phi$$

Thus, the open loop we will use for control-design for this closed-loop will be:

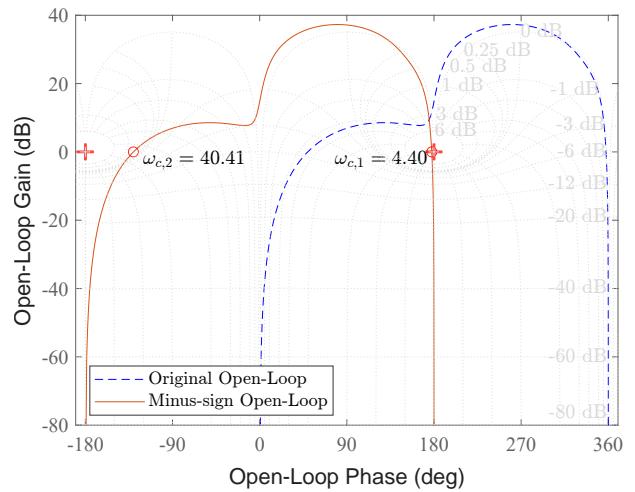
$$L_\theta(s) = -P_\theta(s)C_\theta(s)$$

In figure 3.12, we can see the procedure of designing the damping controller. Lets denote the two crossover frequency's $\omega_{c,1}$ and $\omega_{c,2}$.

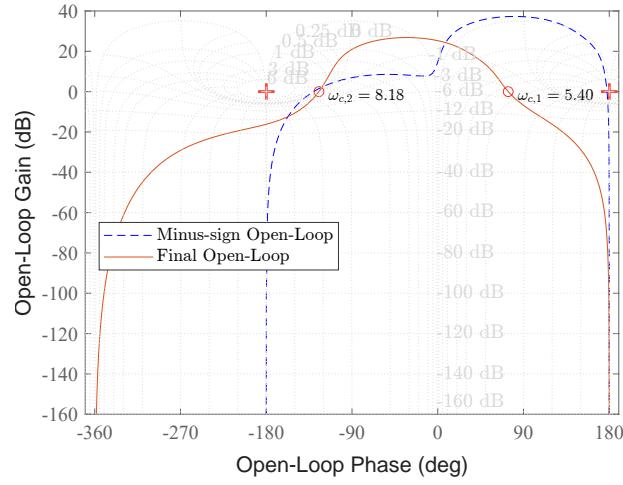
In the start, in figure 3.12a, the closed-loop isn't stable. For this reason, in 3.12b we made the controller $C_\theta(s) = -1$ and we got a stable CL, but with small PM near the $\omega_{c,1}$. Finally, we added a 2nd-order LPF on the first crossover frequency $\omega_{c,1}$ to both increase the PM near $\omega_{c,1}$ and get the whole open-loop closer to the critical point near -180° . In figure 3.12c We see the final open-loop.



(a) Damping open-loop $L_\theta(s)$ with unity controller, meaning $C_\theta(s) = 1$.



(b) Compare damping open-loop before and after a minus sign.



(c) Compare damping open-loop after minus sign and final.

Figure 3.12: Damping controller design procedure, from original to final controller, including the crossover frequencies.

The resulting controller:

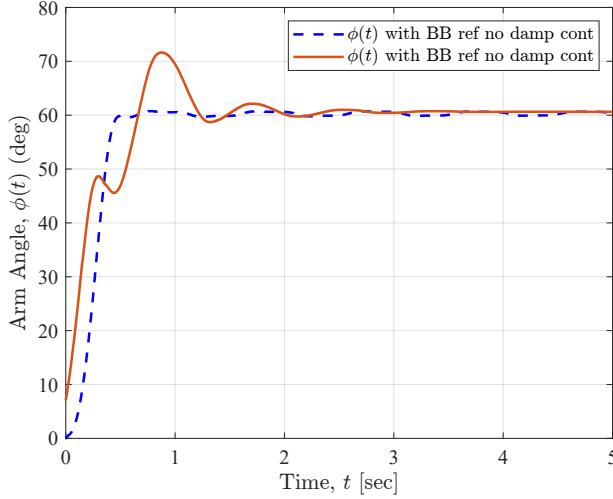
$$C_\theta(s) = \frac{-19.36}{s^2 + 8.8s + 19.36} \quad (3.3)$$

Closed loop stability is ensured through the Nichols diagram, with the use of the LPF in the controller we ensured desired PM and GM and we got specific crossover frequency's.

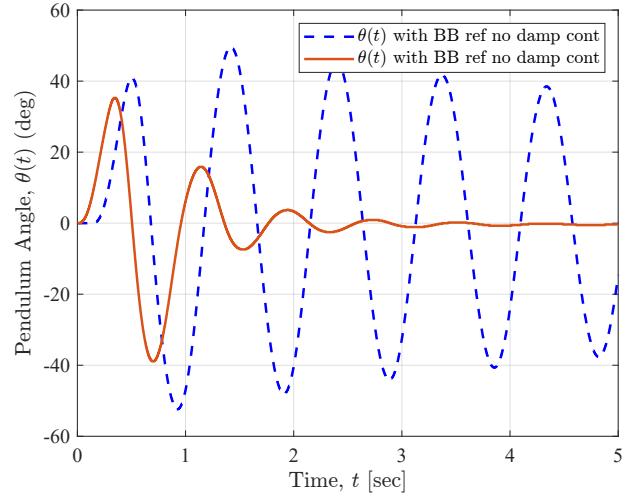
3.4.3 Compare tracking after adding the damping controller

In figure 3.13b we see that with the inclusion of the damping controller, the angle $\theta(t)$ fades to zero after about 3[sec], instead of oscillating with small damping like with the tracking 2DOF closed-loop. However, the price of good damping comes in the form of ruining the reference that the tracking loop gets, thus changing both the

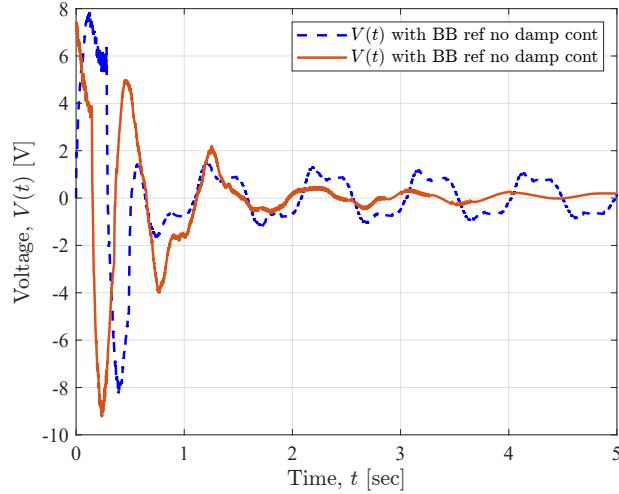
output $\phi(t)$ and the control effort $V(t)$. In figure 3.13a we see an added OS in $\phi(t)$, but no oscillations after about 3[sec], matching the damping in $\theta(t)$.



(a) Tracking performance: 1DOF vs. 2DOF LS controller.



(b) Response of the angle $\theta(t)$ for the 2DOF case.



(c) Comparison of $V(t)$ for 1DOF and 2DOF controllers.

Figure 3.13: Performance analysis of aas.

3.5 Damping Controller from LQG problem

Now we will design the damping controller, on a closed-loop on top of the tracking closed-loop, with LQG problem solution, instead of loop-shaping tools as shown in the previous section.

3.5.1 LQG problem and it's solution

Consider a strictly-proper SISO system $P(s)$ with its state-space matrices $\begin{bmatrix} A & B \\ C & D = 0 \end{bmatrix}$:

$$P : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Bd(t) \\ y(t) = Cx(t) + n(t) \end{cases}, \quad x(0) = 0 \quad (3.4)$$

Where $d(t), n(t)$ are white noise signals with intensities σ_d, σ_n , respectively.

We can define a white noise signal $f(t)$ deterministically by requiring that:

$$|F(j\omega)|^2 \equiv \sigma_f = \text{const} \geq 0$$

This definition is sufficient for the infinite-time LQG problem.

We want to find a stabilizing control signal $u(t)$, which will be a function of $y(t)$, that minimizes the following cost function:

$$J = \int_{t=0}^{\infty} \begin{pmatrix} x^T & u^T \end{pmatrix} \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} dt$$

It can be proven that the optimal solution for this problem is a state-feedback controller based on the LQR gains and a Luenberger estimator based on the Kalman-Filter gains (LQG+Kalman-Filter).

Meaning, the solution is an estimator-based state feedback:

$$\begin{cases} \hat{x} = A\hat{x} + Bu + L(y - C\hat{x}) \\ u = -K\hat{x} \end{cases} \Rightarrow \begin{cases} \hat{x} = (A - BK - LC)\hat{x} + Ly \\ u = -K\hat{x} \end{cases} \quad (3.5)$$

Where the SF gain K is obtained from: $P = \bar{X}, K = R^{-1}(S^T + B^T P)$, from the corresponding solution to the CARE problem of LQR.

And the estimator gain L is obtained from: $Q = \bar{Y}, L = \frac{1}{\sigma_n} QC^T$, from the corresponding solution to the CARE problem of LQE.

3.5.2 Loop-Shaping using an LQG solution

To better dampen the pendulum, a different approach can be used instead of traditional loop-shaping tools - LQG solution.

For a given process, $P(s)$, the LQG solution gives a control law, which is in large a SISO controller from $y(t)$ to $u(t)$, that ensured stability of the CL and optimal stability margins for dealing with white noises for which we specify their intensities.

This method of obtaining the controller is easier than loop-shaping because we don't have to use different controller elements with distinct and sometimes contradicting qualities, and we simply put the process in a function and get a stabilizing controller.

The process in which we will obtain an LQG-based $C(s)$, for process $P(s)$, is as follows:

1. Magnitude Shaping: We first introduce a shaper (or weight) in the form of the TF $W(s)$, which is designed to shape the open-loop's gain $L(s) = W(s)P(s) \equiv P_a(s)$ in a desired manner and add a necessary part into the controller. This includes:

- The desired crossover frequency ω_c .
- Desired gain function: High gains at low frequencies and low gains at high frequencies.
- Maybe an integrator or two are required.

2. Design an LQG controller for the augmented process $P_a(s)$: The augmented process is defined as follows:

$$P_a : \begin{cases} \dot{x}_a(t) = A_a x_a(t) + B_a u_a(t) + B d_a(t) \\ y(t) = C x_a(t) + n(t) \end{cases}, \quad x_a(0) = 0 \quad (3.6)$$

LQR control is designed with the following cost function weights:

$$J = \int_{t=0}^{\infty} (y^2 + u_a^2) dt = \int_{t=0}^{\infty} \begin{pmatrix} x_a^T & u_a^T \end{pmatrix} \begin{pmatrix} C_a^T C_a & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_a \\ u_a \end{pmatrix} dt \quad (3.7)$$

That is, equal weighting for the output and control effort in the augmented system. We will define unit-intensities for both white-noise signals..

3. Thus, we obtain a controller $C_a(s)$ that ensured CL stability for the process $P_a(s)$, with the following CL poles and open poles:

$$\frac{\Delta_{cl}(s)}{\Delta_0(s)} = \left| \begin{pmatrix} A_a - B_a K & B_a K \\ 0 & A_a - L C_a \end{pmatrix} \right| \frac{1}{D_{P_a}(s) D_c(s)} = 1 + L(s) \quad (3.8)$$

This approach provides a stable CL, but does not ensure a stable controller.

3.5.3 Design of damping controller based on LQG

Lets now see the controllers that we can get from the LQG Loop-Shaping process, for 3 different weights: $W(s) = \{0.1, 1, 10\}$.

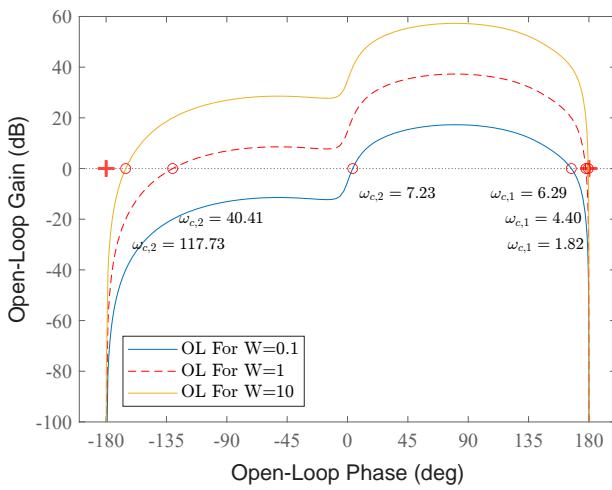
The TF we get from Matlab are:

$$\begin{aligned} C_{\theta,1}(s) &= \frac{-0.2298s^3 - 7.082s^2 - 192.4s - 2618}{s^4 + 21.89s^3 + 730.6s^2 + 1991s + 2.85 \times 10^4} \\ C_{\theta,2}(s) &= \frac{-24.13s^4 - 1003s^3 - 1.524 \times 10^4 s^2 - 1.808 \times 10^5 s - 4.087 \times 10^5}{s^5 + 94.25s^4 + 4628s^3 + 6.947 \times 10^4 s^2 + 3.021 \times 10^5 s + 7.029 \times 10^5} \\ C_{\theta,3}(s) &= \frac{-225.4s^4 - 1.581 \times 10^4 s^3 - 2.291 \times 10^5 s^2 - 9.792 \times 10^5 s - 3.34 \times 10^5}{s^5 + 316s^4 + 5.012 \times 10^4 s^3 + 7.311 \times 10^5 s^2 + 1.066 \times 10^6 s + 7.029 \times 10^5} \end{aligned} \quad (3.9)$$

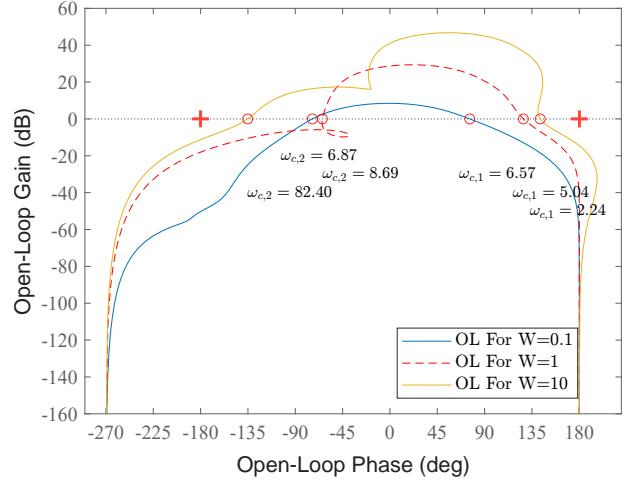
All this controllers are stable. Therefore, we can check the stability in Nichols using the simplified stability criteria.

In figure 3.14, we can see the different OL we get from each choice of weight. In figure 3.14a, we see just a

shift in amplitude when multiplying by $W(s) = \text{const.}$, but in figure 3.14b we see that when implementing the LQG solution on the augmented process, we get different OL's for the damping CL.



(a) OL after multiplying by the $W(s)$ gains.

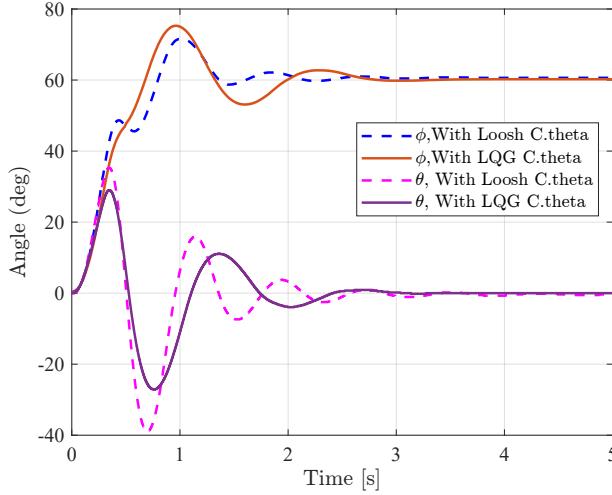


(b) OL after multiplying by the controller from the LQG solution, for each $W(s)$ gains.

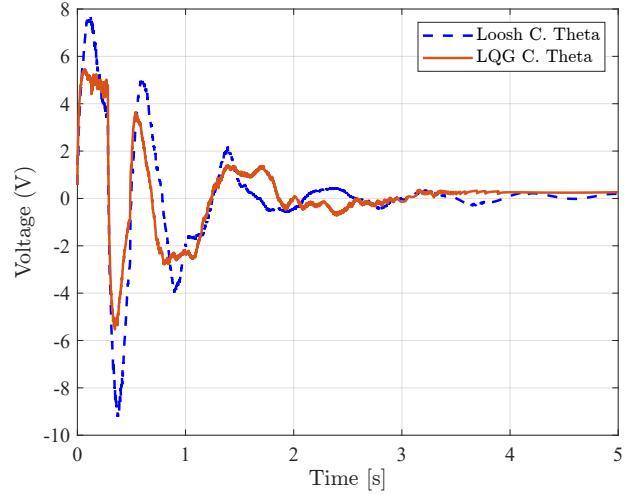
Figure 3.14: Design of Damping Controller $C_\theta(s)$ using LQG solution. On 3.14a is the OL for $C_\theta(s) = W(s)$ for the different weight. And on 3.14b is the OL after implementing a controller based on the LQG solution, for each case of $W(s)$.

3.5.4 Damping controller based on LS Vs. LQG

In figure 3as.



(a) asda.

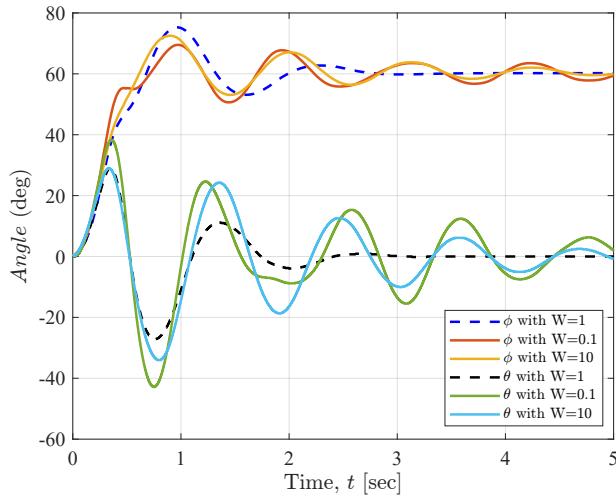


(b) ss.

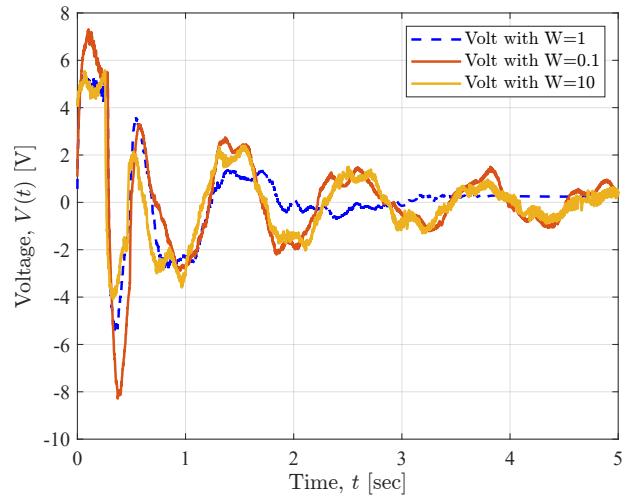
Figure 3.15: asda.

3.5.5 Damping controller based on LQG using different weights

Now, the LQG controller for different W's:



(a) asda.



(b) ssss.

Figure 3.16: asda.

**Now, go over the design of the tracking filter.
Than, show the tracking filter and use Igals graphs to compare stuff.**