Laboratory

The Centrifugal Ring Positioner

Control System Design

MATH-H407

Group 26

Authors: Omidreza Kamalian
Saad Rekiek

Professor: Emanuele GARONE

Assistant: Maxime BUSSIOS

Academic Year 2023-2024

Contents

1 Introduction . . . 1
  1.1 Plant description . . . 1
  1.2 Instrumentation . . . 2
  1.3 Modelling approach . . . 3
2 DC motor analysis . . . 3
  2.1 Governing equations . . . 4
  2.2 System identification . . . 5
3 DC motor control . . . 7
  3.1 Proportional controller . . . 7
  3.2 Velocity control . . . 8
4 Ring position analysis . . . 8
  4.1 Governing equations . . . 8
  4.2 Linearization of the differential equation . . . 10
  4.3 Transfer function of the ring position . . . 12
5 Ring positionner control . . . 12
  5.1 PI controller . . . 12
  5.2 Discretization of the controller . . . 13
  5.3 Ring position control for a step response . . . 14
  5.4 Ring position control for a sinusoidal response . . . 15
6 Conclusion . . . 16
7 Appendices . . . 16
  7.1 Final Code Matlab . . . 16

1 Introduction

In our control system design lab, we're thoroughly exploring the analysis and control of a specific system. This system is a mobile hollow cylinder that moves along a shaft connected to a DC motor. Our goal is to use centrifugal force to precisely control the ring's position at any point along the shaft. The system itself has two main parts: the ring and the motor. We've thoroughly studied each part, identifying areas that need improvement and figuring out their respective transfer functions. Using this info, we've built models for both components in Simulink. After validating these models with the real system, we've developed a primary control for the motor and a secondary control for the ring. This approach, outlined in this report, helps us evaluate how effective our control design truly is.

1.1 Plant description

The plant considered is constituted of a Maxon RE25 gearmotor with different characteristics: ϕ 25mm, 24 VDC, 10W, 43.9mNm/A, reduction ratio 35, ring mass 0.0239 kg, tilt angle 20°, ring friction coefficient 0.3 kg/s, initial ring position 8 cm and the corresponding angular velocity 6.89 rad/s. They are important for the identification step and the resolution around the point of equilibrium.

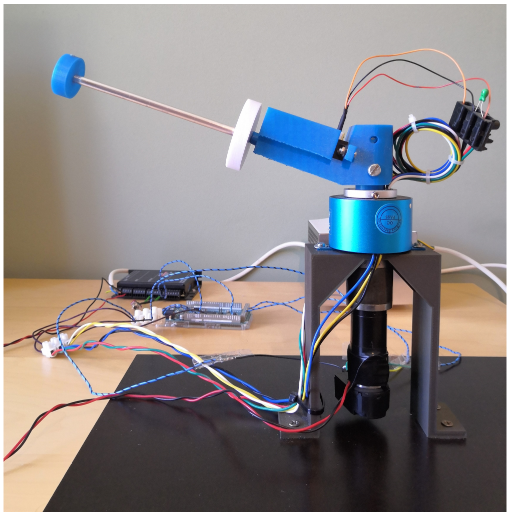

Figure 1.1: The centrifugal ring positioner

1.2 Instrumentation

The system has been enhanced with the integration of two sensors:

- Optical Incremental Encoder: This sensor is designed to provide real-time feedback on the velocity of the motor. It utilizes optical technology to generate incremental pulses corresponding to the rotational movement of the motor. By measuring the rate of these pulses, the system can accurately determine the motor's velocity.
- Infrared Distance Sensor: This sensor is incorporated to determine the position of a ring or object within its range. It employs infrared technology to measure the distance between the sensor and the object (in this case, the ring). The system interprets the sensor's output to ascertain the precise position of the ring relative to the sensor, enabling accurate tracking and monitoring.

Scheme 1.2 illustrates the placement of the infrared distance sensor, which is equipped with an output characteristic detailing the correlation between the output voltage and the position of the ring along the shaft. The graphic visually represents this relationship. Additionally, it's essential to note that the sensor is positioned at a fixed distance of 3 cm from the rotational axis. This factor should be taken into account during the analysis of the results.

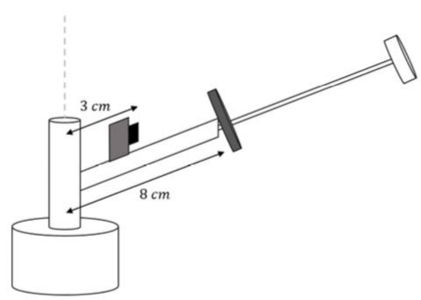

Figure 1.2: Scheme of the centrifugal ring positioner

The system's actuator is the current directed to the DC motor. By regulating this current, which flows through the DC motor, it becomes feasible to speed up or slow down the system, and thus altering the position of the ring.

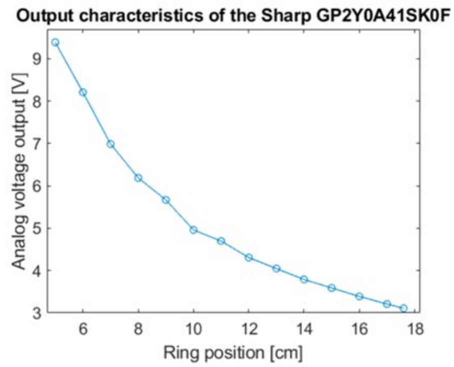

Figure 1.3: Infrared distance sensor characteristics

1.3 Modelling approach

To identify the order of a system two common approaches are used in this project: Gray box modeling and White box modeling. In scenarios involving the white box approach, the system's order is initially known, facilitating the direct determination of the transfer function. Moreover, all relevant mechanical and electrical constants are well-established, allowing for straightforward system identification. In contrast, the gray box approach relies on a combination of physical insights into the system and mathematical tools. Through an understanding of the system's physical reality, it becomes feasible to compute the transfer function, thereby revealing the system's order. However, in this approach, the electrical and mechanical constants are unspecified and remain unknown.

2 DC motor analysis

The primary phase in regulating the speed of the DC motor involves characterizing its behavior, including the identification of non-operational regions and deriving model equations based on these observations. Following this, the initial model is subjected to testing and validation against the real plant to confirm its precision and applicability.

2.1 Governing equations

The motor requires current application to initiate rotation, but this doesn't consistently occur. Specifically, for certain current values, the motor fails to start due to friction. Overcoming this friction is necessary for initiating rotation. Consequently, there exists a dead zone, determined through dichotomous research, where applying a current within the range of -0.9 to 0.3 does not induce motor rotation. On the other hand, the rotational speed reaches also a maximum point where it saturates on both directions -9V and 8.9V.

As explained in subsection 1.3, the mathematical description of the DC motor can be expressed. We can use the grey box approach to identify the system order of the DC motor. Figure 2.1 illustrates the electrical circuit of the DC motor

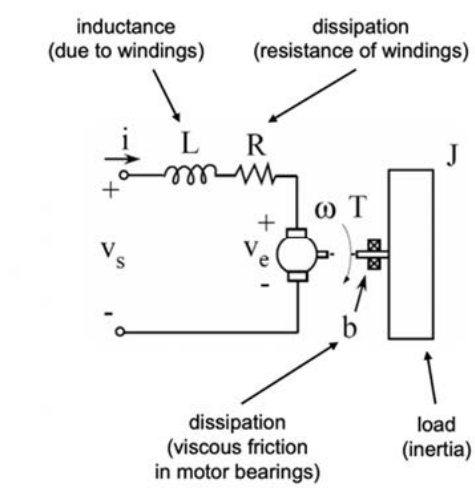

Figure 2.1: Schematic of the electric DC motor circuit

The electrical equation of motion of the DC motor is expressed thanks to the Kirchhoff's law:

$$v_s - v_l - v_R - v_e = 0 \tag{2.1}$$

We have $v_l = L\frac{di_a}{dt}, v_R = Ri_a, v_e = K_v w_m$, and then the electrical equation of motion becomes:

$$v_s - L\frac{di_a}{dt} - Ri_a - K_v w_m = 0 \tag{2.2}$$

The rotor has some characteristics as its viscous friction coefficient F and its inertia J. A current is supplied to the motor and he exerts a torque on the rotor. The mechanical equation of motion of the system is then:

$$J_t \frac{dw_m}{dt} + F w_m = K_i i_a \tag{2.3}$$

Both equations can be computed in the frequency domain by using Laplace transformation. This transformation is done considering initially, the system at rest: i(0) = 0.
After the Laplace transformation, the equations become:

$$V_s - L s I_a(s) - K_v w_m(s) = 0 \tag{2.4}$$

$$J_t s w_m(s) + F w_m(s) = K_i I_a(s) \tag{2.5}$$

By isolating the current in equation 2.5 and substituting it into equation 2.4, we derive an initial transfer function. This function establishes the relationship between the input voltage and the system's rotational speed as the output in the frequency domain. The resulting transfer function is of second order :

$$\frac{w_m(s)}{V(s)} = \frac{K_i}{(sJ + F)(sL + R) + K_v K_i} \tag{2.6}$$

One way to evaluate the order of the system is by considering the timing of actions. In this context, the mechanical constant associated with the motor significantly outweighs the electrical constant linked to the motor inductance, primarily due to the motion of the mechanical rod. Consequently, it is possible to simplify the transfer function to a first-order system.

2.2 System identification

The preceding section illustrates that the system exhibits characteristics consistent with a first-order system. To validate this assumption, the real plant is tested using the MATLAB script "SampleCodeCSD." During this experiment, various speeds, obtained from the velocity sensor for a specific step current, are input into the computer. The resulting step response graph aligns well with the characteristics of a first-order system. The identification code produces the transfer function that accurately represents the actual behavior of the system, outlined as follows:

$$G(s) = \frac{3.48}{3.1s + 1} \tag{2.7}$$

The initial Simulink model, Scheme 2.2, demonstrates a clear similarity in behavior between the model and the actual plant for a step response of 0.88 V, illustrated in figure 2.3 . However, it is noteworthy that the response of the real plant exhibits certain fluctuations. The model should include the Zero-Order Hold (ZOH), responsible for converting the discrete-time signal into a continuous-time signal.

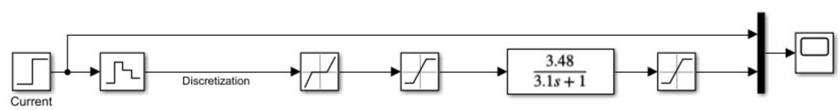

Figure 2.2: Simulink model of the DC motor

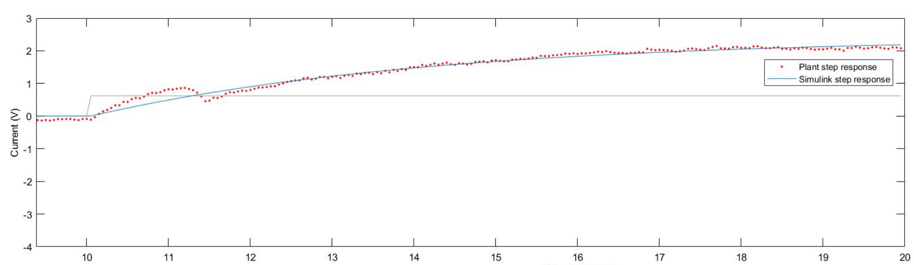

Figure 2.3: Simulink scope of the transfer function vs. plant response to a step of 0.88 V

3 DC motor control

Cascade control will be implemented for system control, as it allows independent control of each subsystem. Moreover, the secondary system has the capability to effectively counteract any disturbances that may impact the overall plant performance.

3.1 Proportional controller

This section focuses on regulating the rotational speed of the DC motor. There is no specific criterion for the static error of the speed, allowing the control to be carried out using a proportional gain, denoted as K_p. To effectively control the system, it is essential to assess the stability of the actual plant. The transfer function is characterized as a first-order function.

$$G(s) = \frac{3.48}{3.1s + 1} \tag{3.1}$$

The stability requirement is met because the single pole exists in the negative region, visible in figure 3.1. This placement of the pole is enough to ensure stability. Based on the phase and gain margin calculations, the gain K_p is about 2.78.

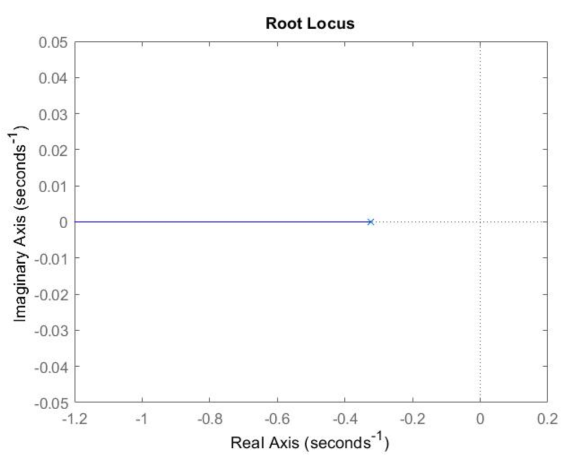

Figure 3.1: Root locus of the DC motor

3.2 Velocity control

Figure 3.2 illustrates The velocity response of the plant versus the input velocity for an input velocity of 3.2 V. The motor control is evidently effective, exhibiting a minor steady-state error, as previously mentioned. However, with the implementation of the outer loop control, this error will be mitigated.

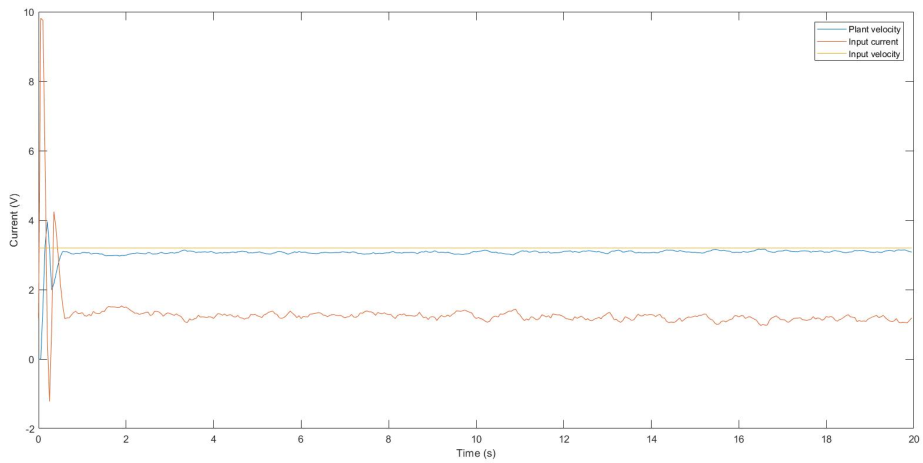

Figure 3.2: Velocity response

4 Ring position analysis

4.1 Governing equations

As mentioned earlier, we have knowledge of the system's physical characteristics, prompting us to adopt a white-box approach. First, loads entering to the system have to be specified.

The examination of forces depicted in Figure 4.1 focuses on the displacement of the ring along the x1 axis. The forces under scrutiny include the centrifugal force propelling the ring's motion, kinetic friction exerting an opposing force to the ring's movement, and the x_1 component of gravitational force. The radius in the axis x_1 is $R = x \cos \alpha$. It is noteworthy that forces along the y_1 axis are not considered, as the ring exclusively moves along the x_1 axis.

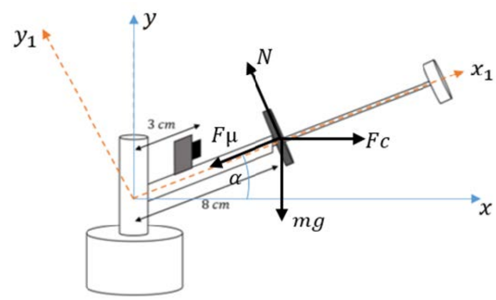

Figure 4.1: Free bofy diagram of the centrifiugal ring positioner

The equation for the centrifugal force is :

$$Fc_{x_1} = mw^2Rcos\alpha = mw^2xcos^2\alpha \tag{4.1}$$

The equation for the gravitational force is :

$$Fg_{x_1} = mgsin\alpha \tag{4.2}$$

The equation for the friction force is :

$$F\mu_{x_1} = \mu\frac{dx}{dt} \tag{4.3}$$

Following the derivation of the equations for forces, the next step involves the application of Newton's second law:

$$\sum F = m\frac{d^2x}{dt^2} \tag{4.4}$$

$$Fc_{x_1} - F\mu_{x_1} - Fg_{x_1} = m\frac{d^2x}{dt^2} \tag{4.5}$$

$$mw^2xcos^2\alpha - \mu\frac{dx}{dt} - mgsin\alpha = m\frac{d^2x}{dt^2} \tag{4.6}$$

$$\frac{d^2x}{dt^2} = w^2xcos^2\alpha - \frac{\mu}{m}\frac{dx}{dt} - gsin\alpha \tag{4.7}$$

Where

- m mass [kg]
- α tilt angle [°]
- x position of the ring along the axis x_1[cm]
- w angular speed [rad/s]
- μ Ring friction coefficient [kg/s]

This equation needs to be linearized since it is not linear.

4.2 Linearization of the differential equation

The system's behavior is described by a second-order differential equation, necessitating the linearization of the equation 4.7 around an equilibrium point. It is possible to determine that the variables which control the system are :

- x: Position of the ring
- $\frac{dx}{dt}$: Velocity of the ring in the shaft
- w: Angular speed of the motor.

The DC motor dictates the angular speed for moving the ring along the shaft, serving as the input, while the ring's position acts as the output. To enhance the representation, let's express the system in a state space format.

$$x_1 = x$$
$$\dot{x_1} = x_2$$
$$x_2 = \frac{dx}{dt}$$
$$\dot{x_2} = w^2x_1cos^2\alpha - \frac{\mu}{m}x_2 - gsin\alpha$$

We use the stability theorem, then it becomes:

$$\begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = w^2 x_1 cos^2\alpha - \frac{\mu}{m} x_2 - gsin\alpha = 0 \end{cases} \tag{4.8}$$

So the equilibrium point is:

$$w_e = \sqrt{\frac{gsin\alpha}{x_e cos^2\alpha}} \tag{4.9}$$

To linearize the Ordinary Differential Equation (ODE), we consider small fluctuations around the equilibrium point. This leads us to express the system's input and output as $w = \widetilde{w} + w_e$ and $x = \widetilde{x} + x_e$. Consequently, we can apply the Taylor expansion theorem to evaluate the following expressions:

$$\frac{d\widetilde{x_2}}{dt} = f(x_e, w_e) + \left.\frac{\partial f}{\partial x_2}\right|_{x_e, w_e} \widetilde{x_2} + \left.\frac{\partial f}{\partial x_1}\right|_{x_e, w_e} \widetilde{x_1} + \left.\frac{\partial f}{\partial w}\right|_{x_e, w_e} \widetilde{w} \tag{4.10}$$

$$\frac{d\widetilde{x_2}}{dt} = w_e^2 x_e cos^2\alpha - gsin\alpha - \frac{\mu}{m}\widetilde{x_2} + w_e^2 cos^2\alpha \widetilde{x_1} + 2w_e x_e cos^2\alpha \widetilde{w} \tag{4.11}$$

Then we have an analysis around the equilibrium point where $w_e^2 x_e cos^2\alpha = gsin\alpha$ could be written as:

$$\frac{d\widetilde{x_2}}{dt} = \frac{\mu}{m}\widetilde{x_2} + w_e^2 cos^2\alpha \widetilde{x_1} + 2w_e x_e cos^2\alpha \widetilde{w} \tag{4.12}$$

Small variations are considered, with $\widetilde{x} = \Delta x$. Then the Laplace transform is applied on the previous equation:

$$s^2 \triangle x_1(s) = -\frac{\mu}{m} s \Delta x_1(s) + w_e^2 \cos^2\alpha \Delta x_1(s) + 2w_e x_e cos^2\alpha \Delta w(s) \tag{4.13}$$

Therefore, the second order transfer function becomes:

$$G(s) = \frac{\Delta x_1(s)}{\Delta w(s)} = \frac{2w_e x_e cos^2\alpha}{s^2 + \frac{\mu}{m}s - w_e^2 \cos^2\alpha} \tag{4.14}$$

4.3 Transfer function of the ring position

using equation 4.14 and having the equilibrium point $(x_e, w_e) = (0.08[m], 6.89[rad/s])$, The calculated transfer function using the white box approach will become:

$$G(s) = \frac{0.97}{s^2 + 12.55s - 41.92} \tag{4.15}$$

It is crucial to emphasize that in the context of the ring positioner system, the input is expected in rad/s, while the output from the DC motor is measured in volts [V]. Therefore, it becomes necessary to establish the relationship between volts and rad/s. This correlation is derived by counting the number of rotations of the plant within a specific time, resulting in the conversion factor: 1V = 1.84 rad/s.

5 Ring positionner control

5.1 PI controller

The ring's position is unstable because when the motor runs, the ring automatically moves to the top. That's why it's necessary to constantly adjust the motor speed to maintain a consistent position for the ring. Thanks to the transfer function and the root locus function which allows us to observe the poles, we notice that the ring's position is unstable because there is a positive pole. Choosing a PI controller in this situation is because the proportional part really helps reduce the error that remains steady. This controller includes an integral part that keeps working over time to improve stability and accuracy by fixing any remaining errors. So, using both the proportional and integral components in the PI controller not only minimizes steady errors but also deals well with changes and uncertainties in the system, making it work better overall. For getting the values of the parameters of the PI controller we use ControlSystemDesigner function in Matlab and tune the controller, we get the PI controller :

$$C_2 = 1.1951 + \frac{0.4}{s} \tag{5.1}$$

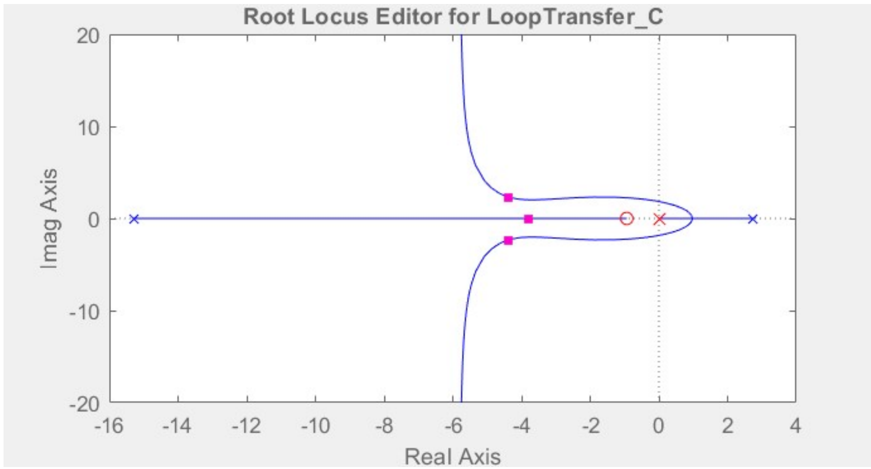

Figure 5.1: Root-Locus of the controlled ring positionner

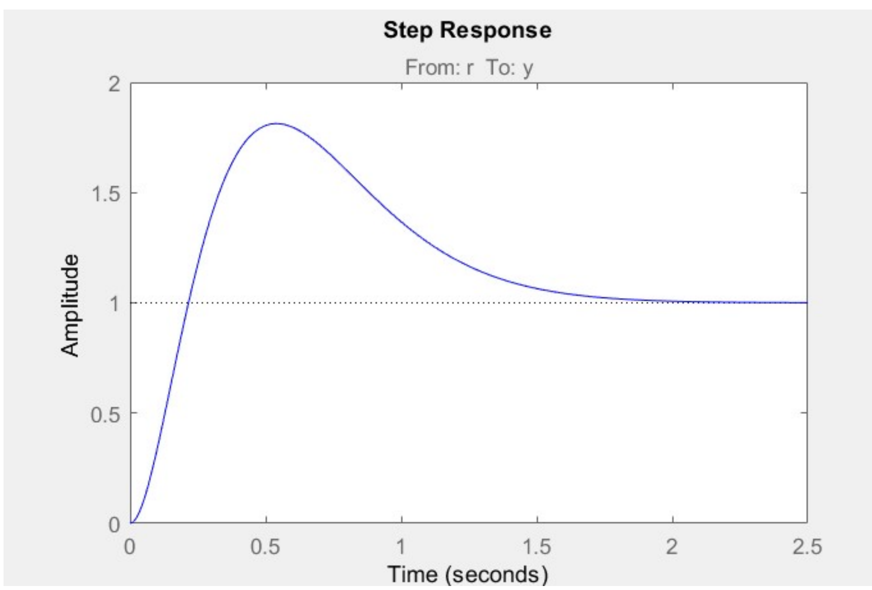

Figure 5.2: Step response of the controlled ring positionner

5.2 Discretization of the controller

To use the PI controller, we can't directly implement it on Matlab, which is why we need to discretize the controller to make it implementable. There are several possible discretization methods, and the one we opt for is the Tustin method.

We have the Laplace equation :

$$U(s) = (Kp + \frac{Ki}{s})E(s) \tag{5.2}$$

U(s) represents the the reference speed and E(s) represents the error between the reference position of the ring and the real position.

With the Tustin method we have

$$s = \frac{2(z-1)}{Ts(z+1)}$$

Using the Tustin method we obtain :

$$U(z) = (Kp + \frac{Ts(z+1)Ki}{2(z-1)})E(z) \tag{5.3}$$

Then when the equation is developped and with using the inverse transform we obtain :

$$2U(k+1) - 2U(k) = 2KpE(k+1) - 2KpE(k) + TsKiE(k+1) + TsKiE(k) \tag{5.4}$$

After this, when we rearrange the equation and introduce a new variable

$$U_0 = U(k-1) + (TsKi - 2Kp)E(k-1)$$

we obtain the following function:

$$U(k) = U_0 + (2Kp + TsKi)E(k) \tag{5.5}$$

5.3 Ring position control for a step response

Finally, when the discretization is implemented in Matlab, we need to send a reference position of the ring position. Two tests were done, one for 10 cm and one for 11 cm. We can see that the real position of the ring follows the reference position that is given to him.

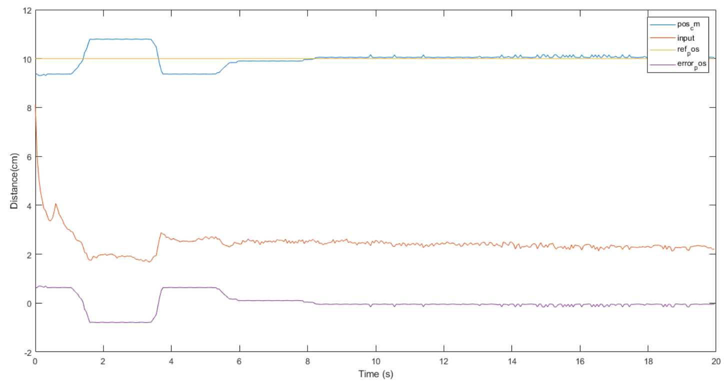

Figure 5.3: Controlled ring position 10 cm distance for a step response

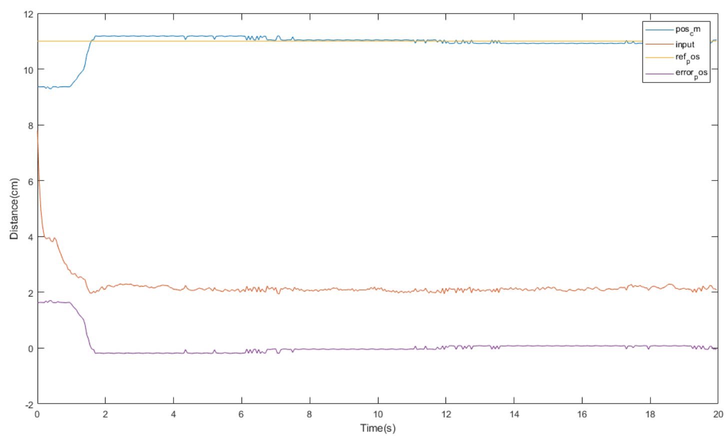

Figure 5.4: Controlled ring position 11 cm distance for a step response

5.4 Ring position control for a sinusoidal response

Tests were also conducted using a sinusoidal waveform as a position reference to observe if the system accurately tracked this more complex reference compared to a step response. The obtained results are shown below; they are relatively good, but there still exists a significant amount of error. It can be observed that the actual position doesn't precisely follow the reference position. When the sinusoidal wave reaches its extremes, we observe an odd behavior in the actual position of the ring, which is due to friction between the ring and the shaft.

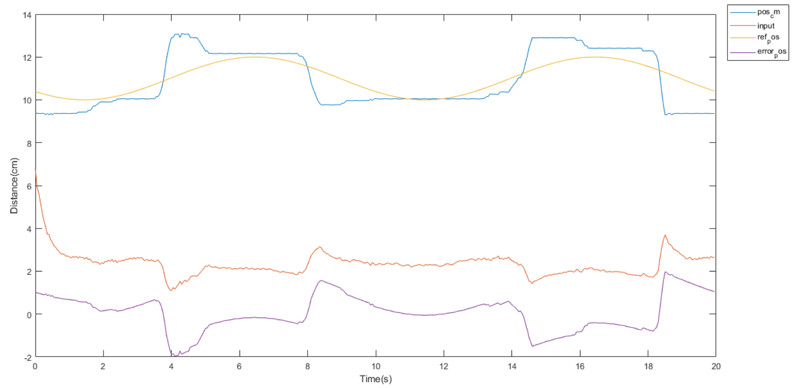

Figure 5.5: Controlled ring position 11 cm distance with sinuosidal of 2 cm amplitude max

6 Conclusion

In conclusion we needed to control a dynamic system, the ring positioner. We used system identification with two methods: Gray box for the DC motor and White box for the centrifugal ring. To avoid any harm to the actual system, we built numerous Simulink models to test different controllers. We made the choice of a cascade controller that allows individual subsystem control, empowering the secondary one to respond faster to disturbances than the primary system. We applied simulation results to the real setup to validate transfer functions and controllers. Comparisons were drawn between the model's step responses and those of the real system, including responses to sinusoidal inputs. After all of this, we had to create the different controllers and determine their values, either mathematically or by tuning in Matlab, to obtain the correct parameter values. We chose a P controller for the motor and a PI controller for the ring position. Finally, we discretized our controller to implement it in Matlab, then sent a reference position for the ring to verify its ability to stabilize at that position.

7 Appendices

7.1 Final Code Matlab

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Setup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
openinout; %Open the ports of the analog computer.
Ts=1/20;%Set the sampling time.
lengthExp=20; %Set the length of the experiment (in seconds).
N0=lengthExp/Ts; %Compute the number of points to save the datas.
Data=zeros(N0,4); %Vector saving the datas. If there are several datas to save, change "1" to
DataCommands=ones(N0,3); %Vector storing the input sent to the plant.
cond=1; %Set the condition variable to 1.
i=1; %Set the counter to 1.
tic %Begins the first strike of the clock.
time=0:Ts:(N0-1)*Ts; % Vector saving the time steps.
ref_position_1 = 8; % Initial position in cm
ref_position_2 = 13; % Final position in cm
xvolt =[3.1,3.2,3.4,3.6,3.8,4,4.3,4.6,4.9,5.6,6.2,7,8.3,9.4];
xcm = [20.6,20,19,18,17,16,15,14,13,12,11,10,9,8];
reference_speed = 3.2; % Speed reference in voltage
prev_error=0;
ref_speed_til=6.86;
```

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Kp1 = 0.65 ;
Kp2 = 1.1951;
Ki = 0.4 ;
w=2*pi/10;
    for j=1:N0
    t=toc;
    w=2*pi/10;
    input = 0.88;
   for the ring position subsystem
    if t<10
        ref_position=11+sin(w*(j-2*(N0/10))*Ts)%11; %Input of the system
    else
         ref_position=11+sin(w*(i-2*(N0/10))*Ts);%11;
    end
    end
```

```
while cond==1

%   %  for motor velocity subsystem
%     if t<10
%         input=0; %Input of the system
%     else
%         input=1.4;
%     end

    ref_position=12+sin(w*(i-2*(N0/10))*Ts);

    [in1,in2,in3,in4,in5,in6,in7,in8]=anain; %Acquisition of the measurements.
    if in2>9.4
        position_cm=8;
    elseif in2<3.1
        position_cm=20.6;
    else
    position_cm=interp1(xvolt,xcm,in2,'linear','extrap');
    end
    error = ref_position-position_cm ;
    reference_speed = ref_speed_til + (Kp2 + Ki*Ts)*error-Kp2*prev_error ;
%     reference_speed = 0.3*97.087*(error-  0.9648*prev_error)+ref_speed_til+5 ;
    if reference_speed>8.8
        reference_speed=8.8;
    elseif reference_speed<0
         reference_speed=0;
    end
```

```
    input = Kp1*(reference_speed-in1*1.84) + 0.88; %for motor velocity subsystem
    if (input>0)
        input=input+0.3;
    elseif (input<0)
        input=input-0.9;
    end
    Data(i,1) = in1 ; %Save one of the measurements (in1).
    Data(i,2) = in2 ;
    Data(i,3) = position_cm;
    Data(i,4) = error;
    anaout(input,0); %Command to send the input to the analog computer.
    DataCommands(i,1)=input;
    DataCommands(i,2)=ref_position;
    DataCommands(i,3)=reference_speed;
    t = toc; %Second strike of the clock.
    prev_error=error;
    ref_speed_til=reference_speed;
    if t>i*Ts
        disp('Sampling time too small');%Test if the sampling time is too small.
    else
        while toc<=i*Ts %Does nothing until the second strike of the clock reaches the sampling time set.
        end
    end
    if i==N0 %Stop condition.
        cond=0;
    end
    i=i+1;
```

```
end


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plots
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

closeinout %Close the ports.
figure
plot(time,Data(:,2))
hold on
plot(time,Data(:,1))
hold on
plot(time,DataCommands(:,1))
% plot(time,Data(:,1),time,DataCommands(:),time, Data(i,1)*ones(400,1)); %Plot the experiment (input and output).

figure
plot(time,Data(:,3),time,DataCommands(:,1),time,DataCommands(:,2),time,Data(:,4))
legend('pos_cm','input','ref_pos','error_pos')

% figure
% plot(time,DataCommands(:,3),time,Data(:,1))

```