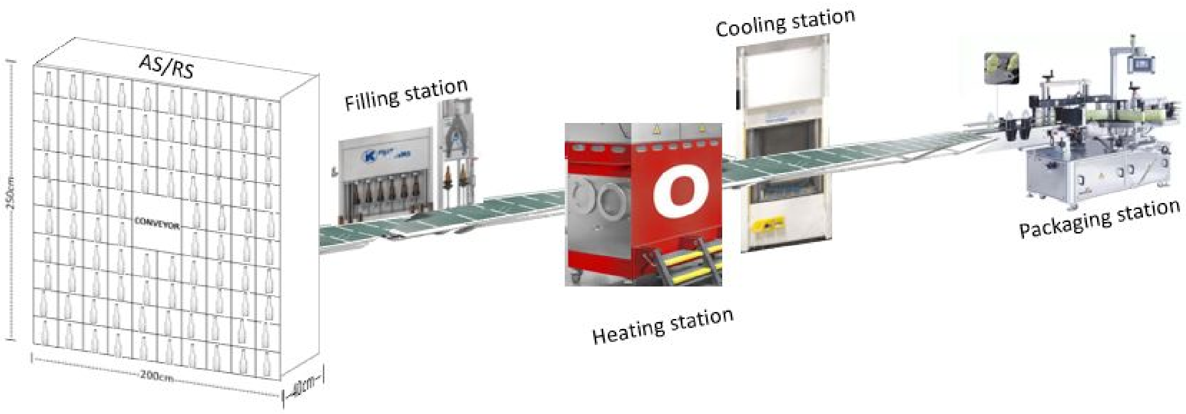
Tecnológico de Monterrey
Campus Guadalajara

Control Engineering

Final Project: LORUVA's Production line



Luis Alfredo Aceves Astengo	A0l229441
Vanya Michelle Medina Garcia	A01228016
Rodrigo Valdéz Hernández	A01229322

INDEX

Introduction

One of the greatest achievements in modern manufacturing is the production or assembly line. Before the industrial Revolution objects were often manufactured from end-to-end by a single person. If a single object required 5 parts and 20 steps , a single individual had to manufacture it all until getting the final product . After the Industrial Revolution manufacturers began to place operators on specialized steps or tasks. Nowadays we have components or stations that are attached successively as the assemblage or product moves along a conveyor. This means that as the unit moves by, each station along the line performs a specific task.

In this project we are working with the company named LORUVA. LORUVA is a mexican beer company that is very popular over the world. They are opening a new factory at the city center and they have chosen us to design their production line. We are a specialized company that designs the transit and end of line conveyors for any packaged or boxed products. The design for the production line will be explained through out the report. Some important ideas will be to analyze each one of the steps and stations to manufacture and to the deliver the final product. There has to be a simplification of all the movement throughout the conveyor with no cross flow, backtracking, or repetitious procedure. Another important point is that all the stations have to be compatible. Having this in mind and the requirements for the beer to be processed, our production line will have the following components: a conveyor, DC motors with three directional movements for the Automated storage and retrieval system(AS/RS), a filling station with the limiting requirements, a heating and cooling station with specific temperatures over time and to finalize a packaging station.

AS/RS

Description and Characterization

The reason why we are using an Automated storage and retrieval system is to minimize the space in the storage and to increase effectiveness and productivity.
Given the specifications we know that the storage has a length of 100cm , height of 250cm and a depth of 40cm. Now the shelf of the storage is divided 10x10, this means that each cell's width measures 10cm and the height is 25cm. In the middle there is a hole in which the displacement arm has to go through in order to place the bottle in the conveyor. The next figure is an example of the storage and conveyor.

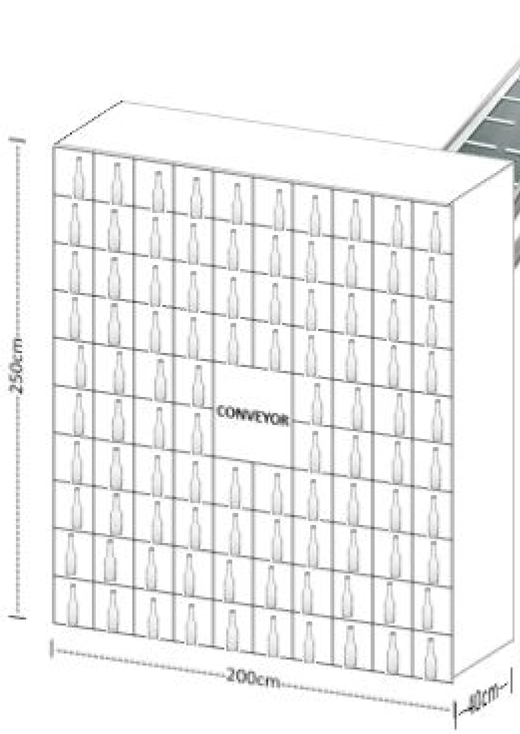


Figure 1.0 Storage

Now by knowing the position of the storage and conveyor we can proceed to design.
The AS/RS system will have three different movements that will be done with DC motors: Horizontal, Vertical and placement. The measurements and number of blocks is the key for the design, we have to know how much the horizontal and vertical movements have to change in order to grab the bottle from the middle. Perse if we want to grab the bottle that is placed on the left down corner and we start on the middle, we have to move 45cm to the left and 112.5 downwards.

In order to make the DC motors more efficient and to reduce error, we have the following requirements:

- At 50% of its capacity, the axel of the motor moves at the speed 2.5rev/s..

- The motors should not work above their 90% capacity

In order to keep a balance each one of the DC motors have a specified speed and noise factor. For the horizontal movement we have that for every revolution the motor moves 1 cm and the noise factor is 0.04. The vertical movement is the fastest from the DC motors, it moves 2 cm each revolution and the noise factor is 0.05. The displacement moves 0.5cm per revolution and has a noise factor of 0.03.

For the modelling of the system, a previous model was provided and this was followed to obtain the current transfer function. By measuring the peak time of the first overshot, the maximum peak value and the value in steady state we used the formulas, that will be shown right below, to obtain the transfer function used for this system.

$$T_P = 1.2\ s$$

$$\omega_d = \frac{\pi}{T_P} = 2.618$$

$$M_P = \frac{Max_{Overshoot} - y_{ss}}{y_{ss}} = \frac{2.8 - 2.5}{2.5} = 0.12$$

$$k = log(M_P)$$

$$\delta = \frac{\pm k}{\sqrt{k^2 + \pi^2}}$$

$$k_s = y_{ss}$$

With the previous formulas, the last step to obtain the final transfer function is to substitute on the formula below:

$$G = \frac{K_s \omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2}$$

Control simulation and analysis

So our AS/RS with the 3 motors can be seen on Figure 2.0:

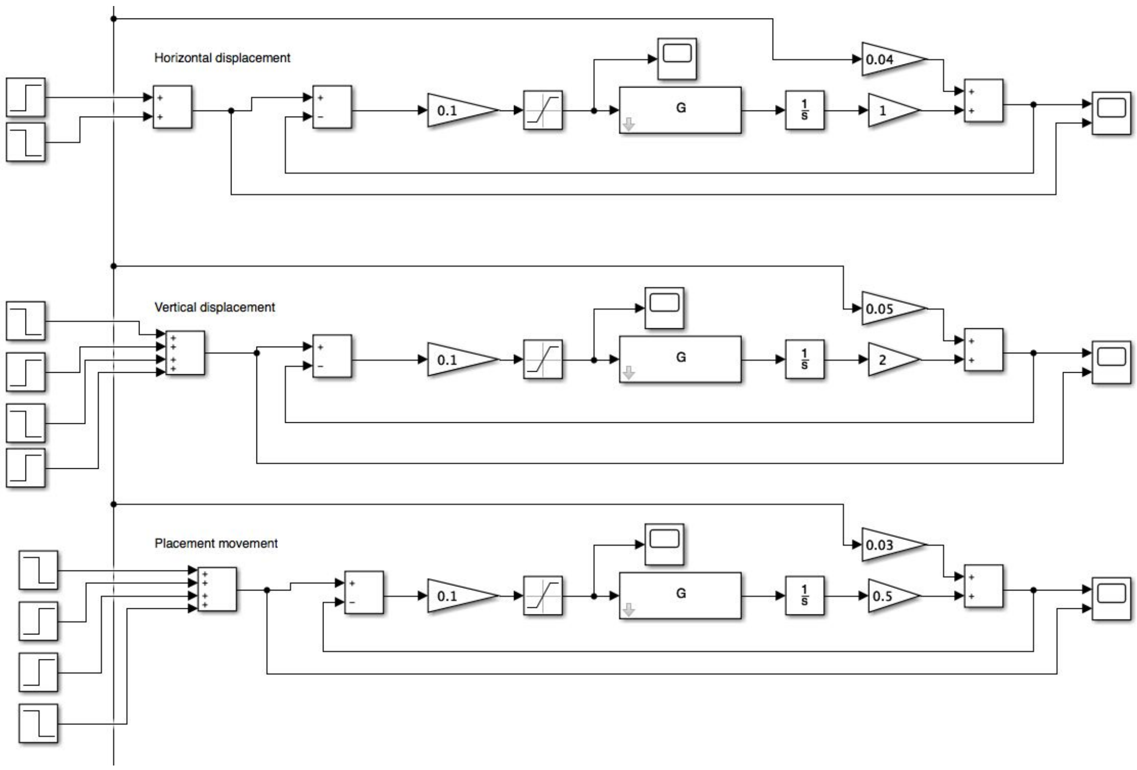


Figure 1.1 AS/RS Motors

We can see the correspondent noise factors. As the transfer function from the motors represents their speed, but we want the output as displacement, we need an integrator after plant. This integrator (pole in 0) guarantees that the output follows the reference. We added saturators before the plant, for the motors not to work above 90% of their capacity. Also, the 0.1 gain before the saturator ensures the following:
This is the input to the plant with this gain equal to 1:

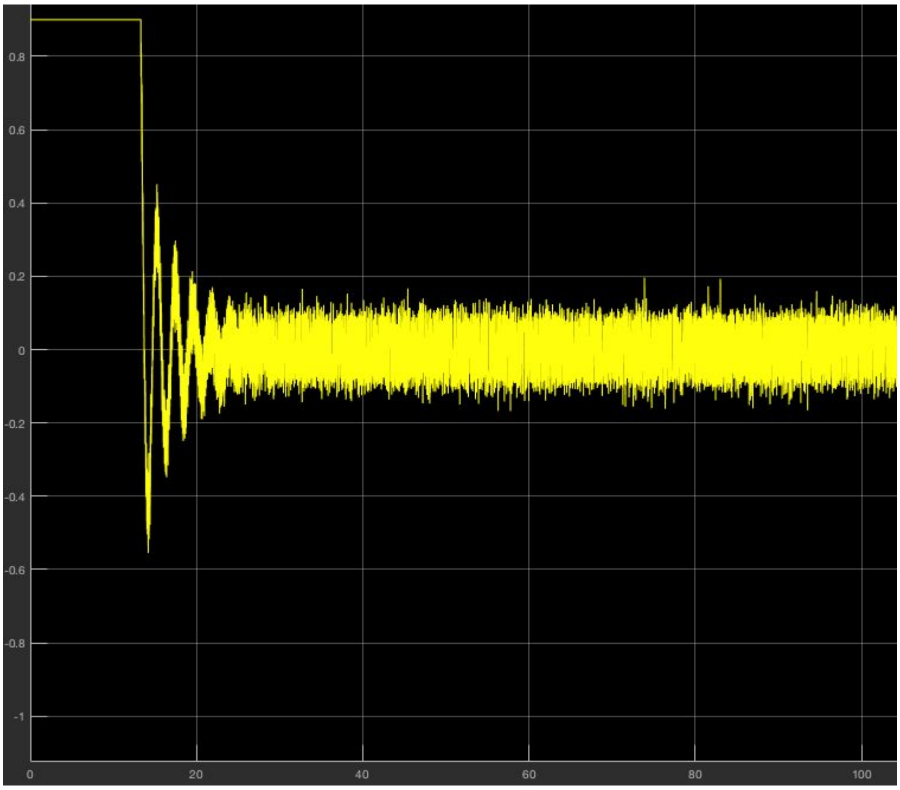


Figure 1.2 Input with gain equal to 1

And this is the input to the plant we the gain of 0.1 as we have it:

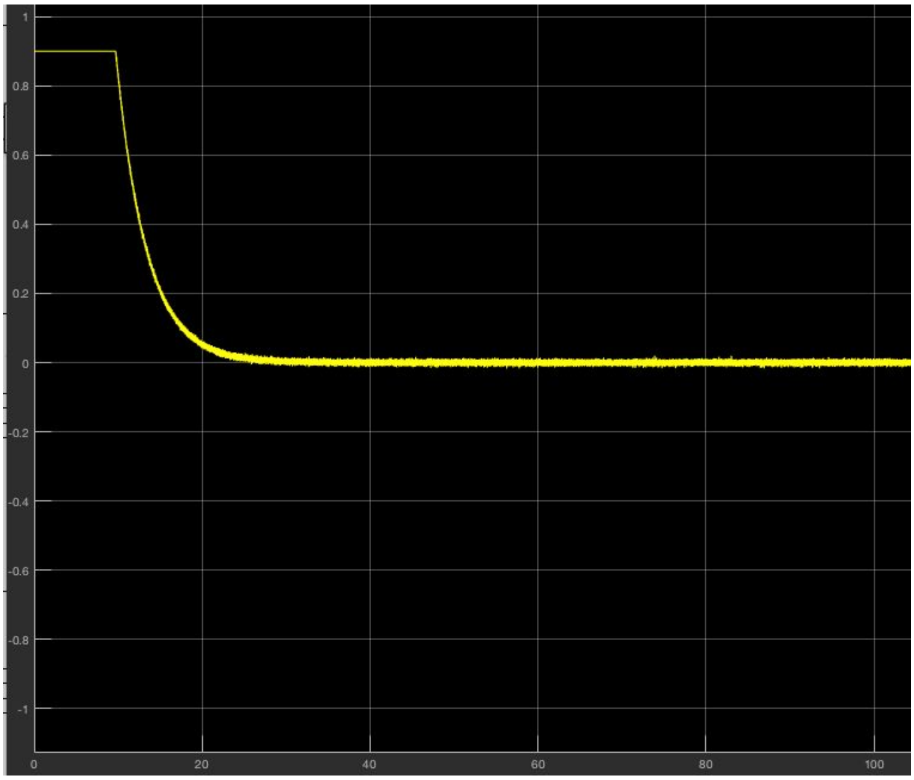


Figure 1.3 Input with gain equal to 0.1

As we can see, it has a more stable input to the motors, so we don't force them.
So to prove the functionality, we have (x,y) variables to indicate which bottle we want to retrieve. The coordinates asume we start in (0,0). So, for example, let's say we want to go for bottle (2,-3). Translating this to displacement, the horizontal motor should move (positive, right) 30cm and the vertical should move -62.5cm (negative, downwards). These calculations are made from the fact that we can move horizontally 100 cm to each side, and vertically 125cm to each side. But also we want to be centered on each cell to retrieve the bottle.
So now, simulating a whole process of retrieving a bottle with the given data, we have the following:

Horizontal movement:

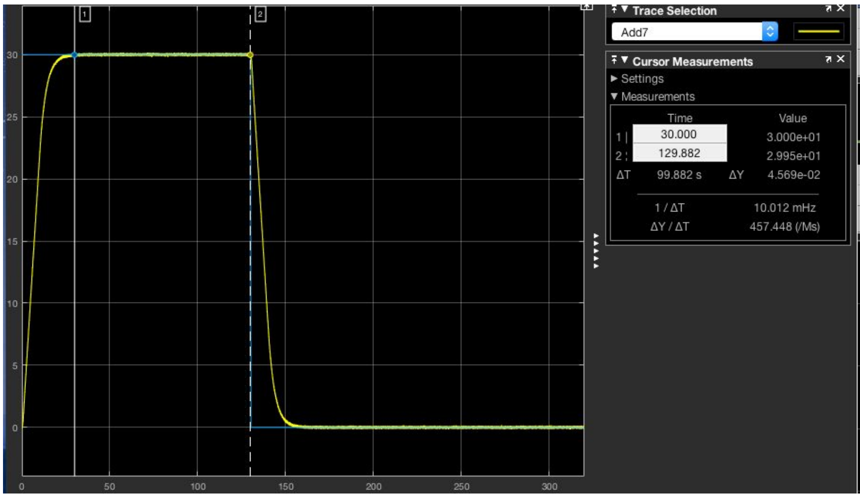


Figure 1.4.1 Horizontal Movement

Vertical movement:

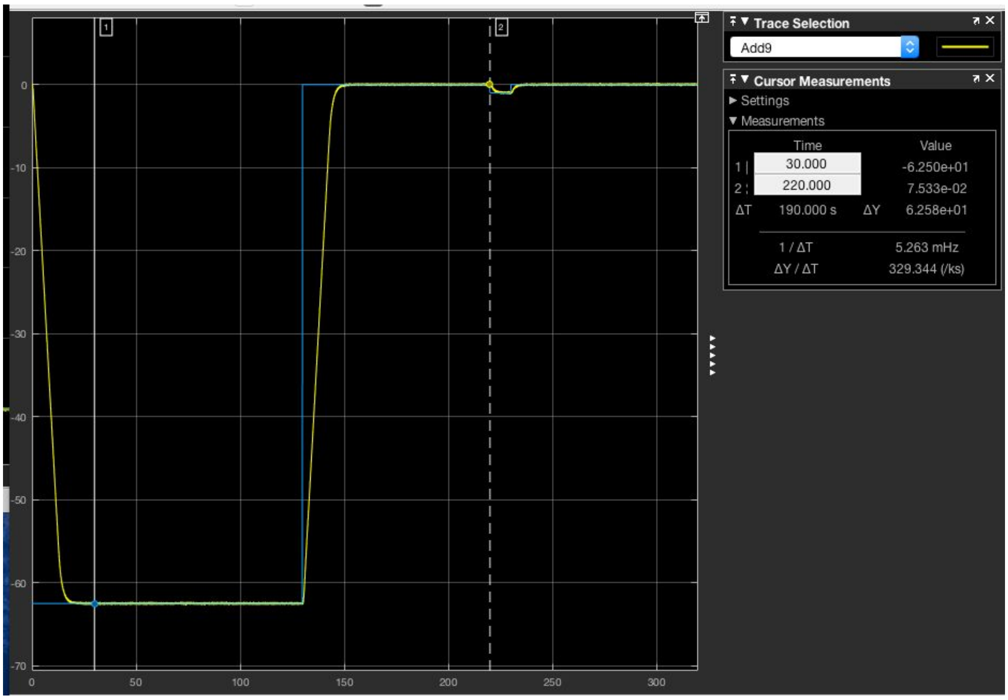


Figure 1.4.2 Vertical Movement

Placement movement:

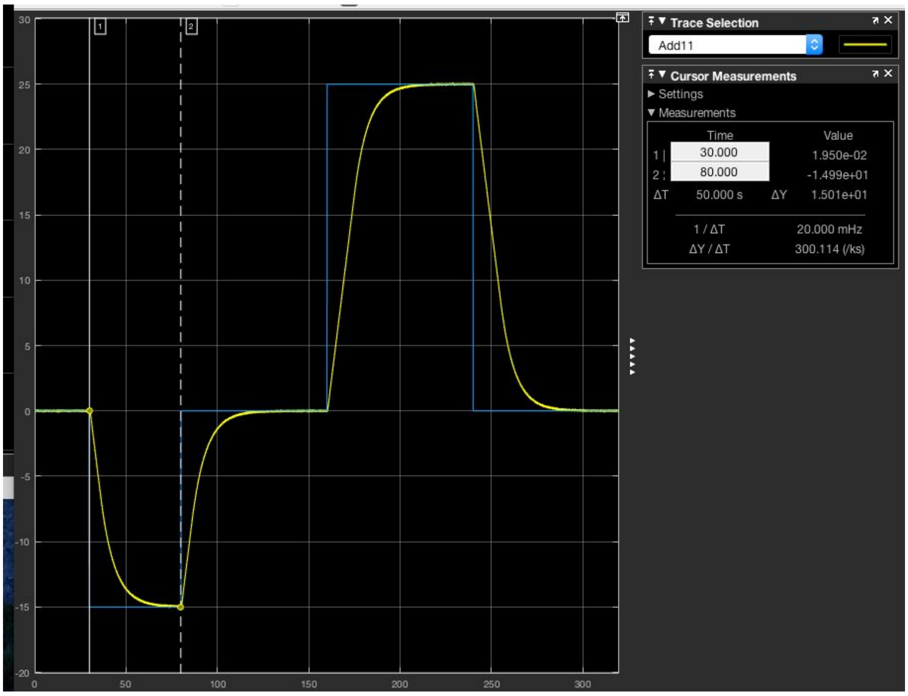


Figure 1.4.3 Placement

So the sequence goes as follows:

At time 0, both horizontal and placement motors start moving towards (30,-62.5). By time 30, both are in place, so the placement motor starts moving towards the cells -15cm (remember this is a 40cm motor and moves +25cm towards the conveyor) and then 15cm back once it has the bottle. This ends by time 130. So now both horizontal and vertical can return to the origin. This is complete about time 160. Now the placement motor can start moving +25cm towards the conveyor. By time 220 this is complete and the vertical motor starts the 1 cm movement downwards to leave the bottle in the conveyor. The second it has to remain in position -1cm is taken into account. Once this is done the placement motor returns -25cm to the initial position.

We would like to mention our system is calibrated, meaning we measured each delay for the possible positions of the horizontal and vertical movement, so we don't have our system in idle using a maximum constant delay. Our code for this looks something like this:

```
%horizontal position
x=2;
if(x>0)
   xdisp=20*x-10;
elseif (x<0)
   xdisp=20*x+10;
else
   xdisp=0;
end

%delay to arrive
if(abs(x)==5)
```

```
    delayx=60; %100 for placement movement
elseif(abs(x)==4)
    delayx=50;
elseif(abs(x)==3)
    delayx=40;
elseif(abs(x)==2)
    delayx=30;
elseif(abs(x)==1)
    delayx=20;
else
    delayx=0;
end

%vertical position
y=-3;
if(y>0)
    ydisp=25*y-12.5;
elseif (y<0)
    ydisp=25*y+12.5;
else
    ydisp=0;
end

if(abs(y)==5)
    delayy=40; %100 for placement movement
elseif(abs(y)==4)
    delayy=30;
elseif(abs(y)==3)
    delayy=25;
elseif(abs(y)==2)
    delayy=20;
elseif(abs(y)==1)
    delayy=10;
else
    delayy=0;
end

if(delayx>delayy)
    delaypos=delayx;
else
    delaypos=delayy;
end
```

The calculations for horizontal and vertical displacement can be seen as well. The placement motor makes always the same movement, so special measurements for each case are not needed.

Here is a another test scenario: retrieve bottle (5,-5) (one of the furthest):
Horizontal movement:

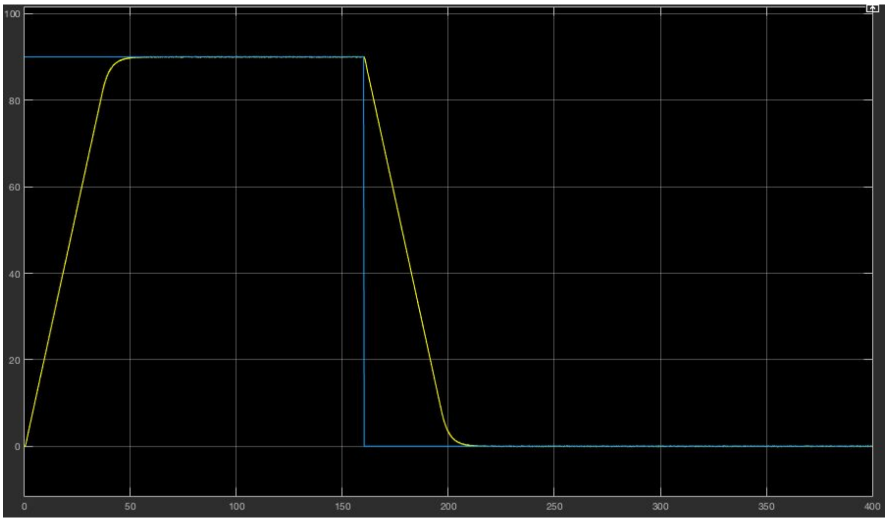


Figure 1.5.1 Horizontal Movement (Test Scenario)

Vertical movement:

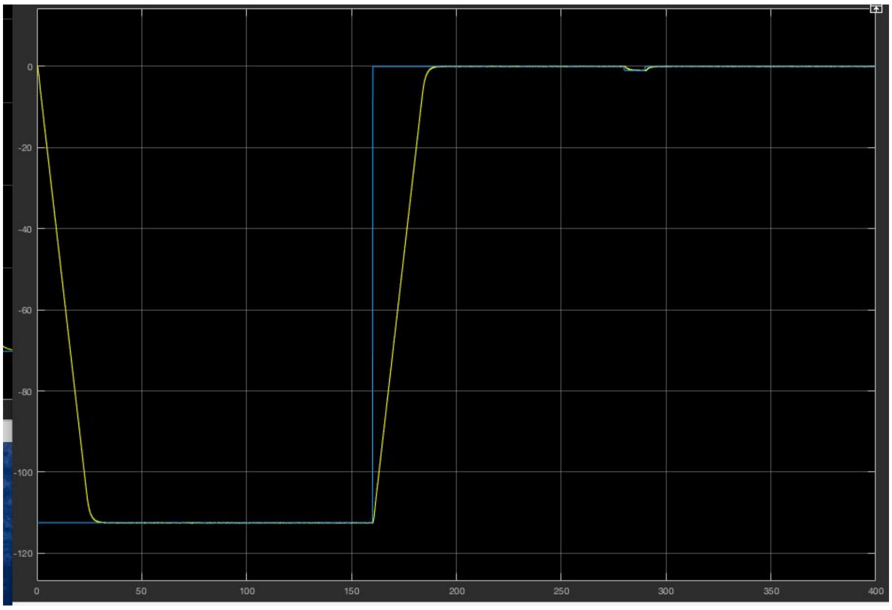


Figure 1.5.2 Vertical Movement (Test Scenario)

Placement movement:

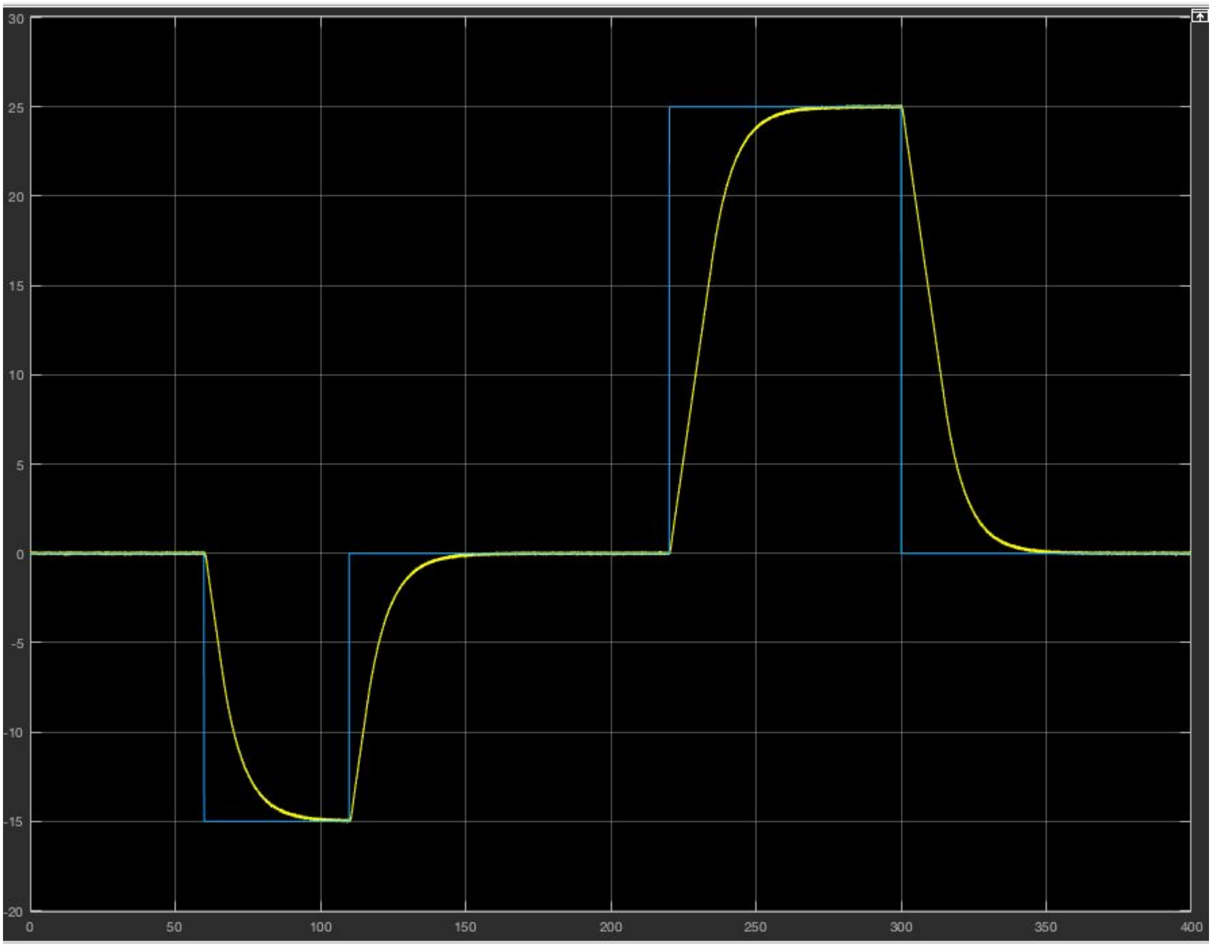


Figure 1.5.3 Placement Movement (Test Scenario)

We can see the process now takes longer.

Filling Station

Description and Characterization

This station controls the filling volume of each one of the bottles with the prepurified beer. This station has its own displacer of type 1. Since this station works with liquids, the velocity of the displacer has to be slower for the liquid not to spill. The functionality of this station is: once the bottle is placed in front of the filling station by the conveying line, the displacer moves toward the bottle, grabs the bottle returns it to the valve and fills it to the correct amount in order to place it back in the conveyor. Many things have to be taken into account in order to fulfill the requirements of the filling of each bottle.
LORUVA gave us the exact parameter and limitations of the volume for each bottle. According to their standards the maximum volume of the bottle is 260mL and the minimum is 245mL. Inside the filling station there is a pump, when it works at its 100% of capacity then it pumps 15 L/min which is equivalent to 250 mL/s. Our goal is to fill the bottle in less than 1.5s.
For the transfer function of the filling station, a basic analysis is needed, first for this system it is required to know the volume, also there is only inflow to the bottle and no outflow so the following function can work for us.

$$Q_i(s) - Q_o(s) = sV(s)$$

Since Qo is equal to zero then the final equation is:

$$V(s) = \frac{Q_i(s)}{s}$$

$$G = \frac{V(s)}{Q_i(s)} = \frac{1}{s}$$

Figure 2.1 Filling station

Control simulation and analysis

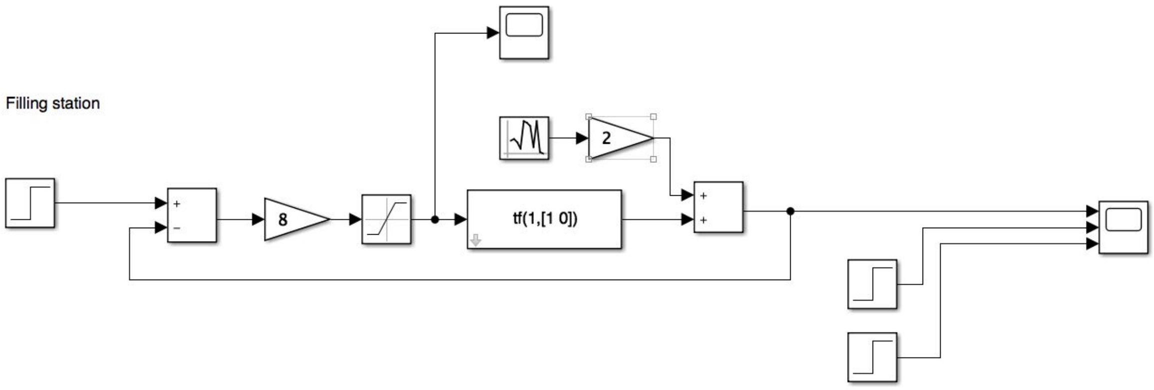


Figure 2.2 Simulink Model (Filling Station)

We added a noise factor of 2 for this plant. Our reference is a step of 250, for it to be just between the desired 245 and 260mL. We have a saturator to avoid exceeding the maximum capacity of 15L/min = 250mL/sec of the pump. Also, we used a proportional controller. 8 seemed like a fair value due to the following:
If we put something high like 40, we fill it faster, but the input to the plant is at maximum capacity for a lot of time:

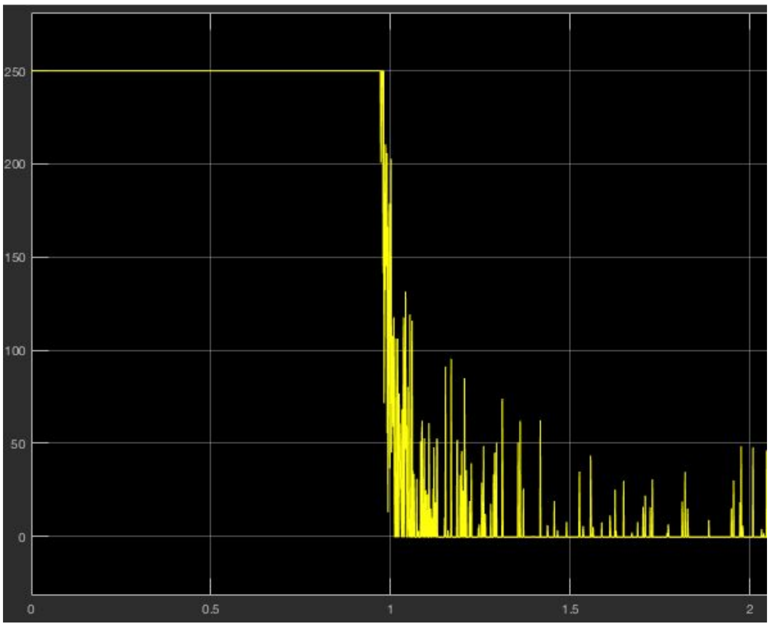


Figure 2.3.1 Input with maximum capacity

And with our gain of 8 we have the following input:

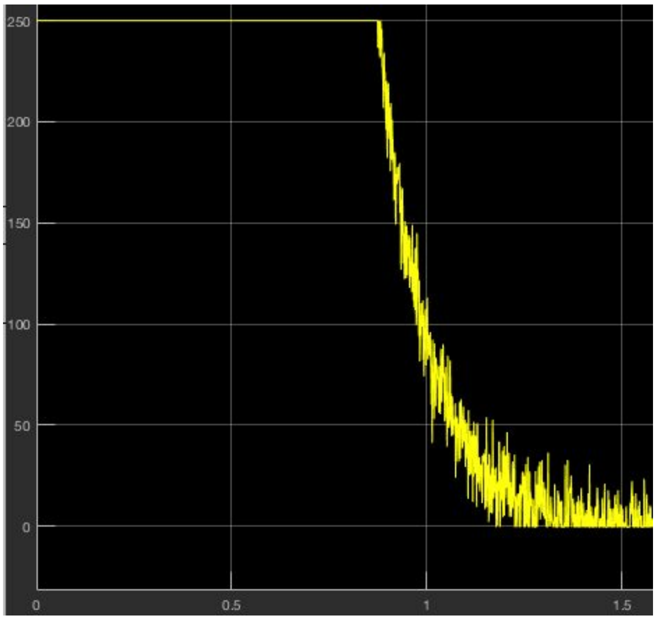


Figure 2.3.2 Input with a gain of 8

We reduced a bit the time at full capacity, but we didn't sacrifice our desired filling time:

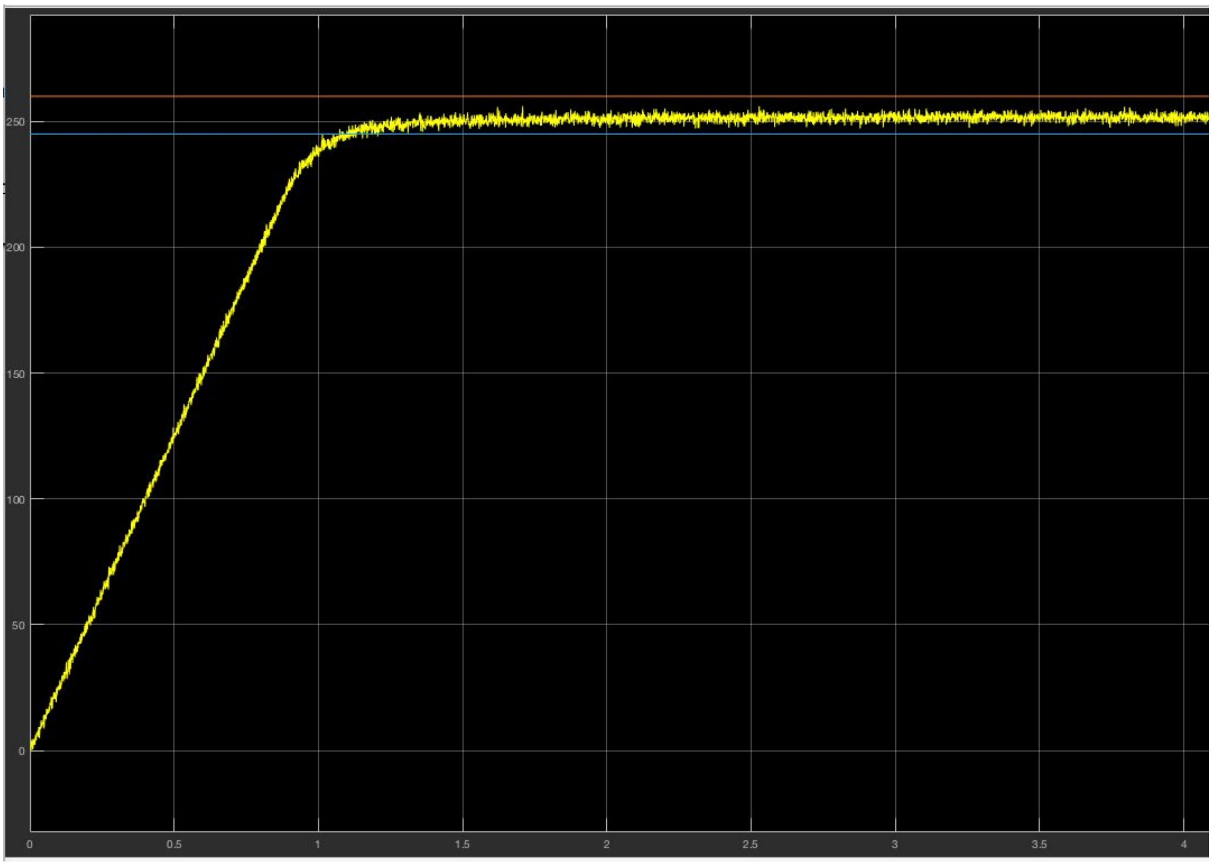


Figure 2.4 Output for the Filling Station

We can conclude that the system works as desired, staying inside the limits listed on the requirements.

Heating Station

Description and Characterization

The process of making beer is complex. When the bottle reaches this station the beer has already passed through other procedures of filtration, extraction, fermentation, brewing, etc. The heating station receives the pre purified beer, we have to elevate the temperature of the liquid in order to kill the last pieces of bacteria that the beer has in order to be ready for consumption. The entire process heat demand of the thermally driven process of wort boiling in this station can be met with a temperature of 200C.
Before going on with our design we first had to understand the processes of breweries and malting plants in order to achieve the cancellation of the last microbacteria. In order to get rid of all the last pieces of micro bacteria we need a very efficient heater that has to reach from 20C up to 150 C in less than 60 seconds at 40 % of its capacity. Since the heater reaches an elevated temperature in less than one minute, it has a noise factor of 1.
In the heating station, the transfer function that is needed can be obtained by using the information that was presented on the requirements. With the requirements of the system, ks at 40% is equal to 130, so what is next is to get it to a 100% with the rule of three, giving us that ks=325. Also the time is given to us and it is equal to 60 seconds. From that the information we just need to use the following:

$$k_s = 325$$

$$t_{ss} = 60\ s$$

$$\tau = \frac{t}{5} = \frac{60}{5} = 12$$

With this it is a matter of substituting on the following formula to get the transfer function:

$$G = \frac{k_s}{\tau s+1} = \frac{325}{12s+1}$$

Figure 3.1 Heating station

Control simulation and analysis

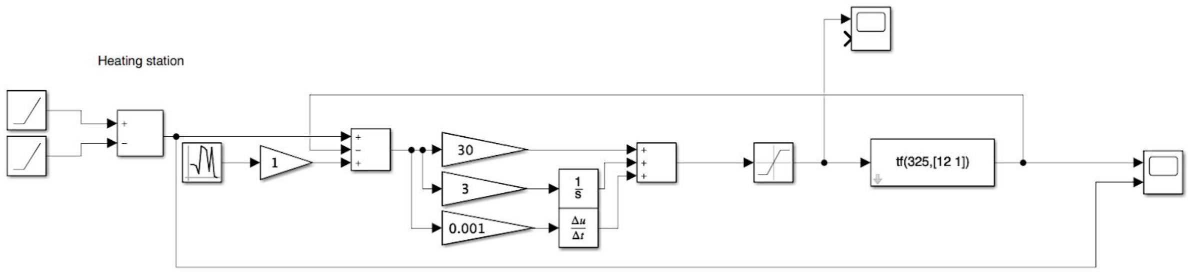


Figure 3.2 Simulink Model (Heating Station)

The noise factor for this station is one. We have a saturator at 1, to avoid the plant to work above 100% percent its capacity. After a bit of trial and error (especially to avoid some derivative errors in matlab) we came up with the following PID:

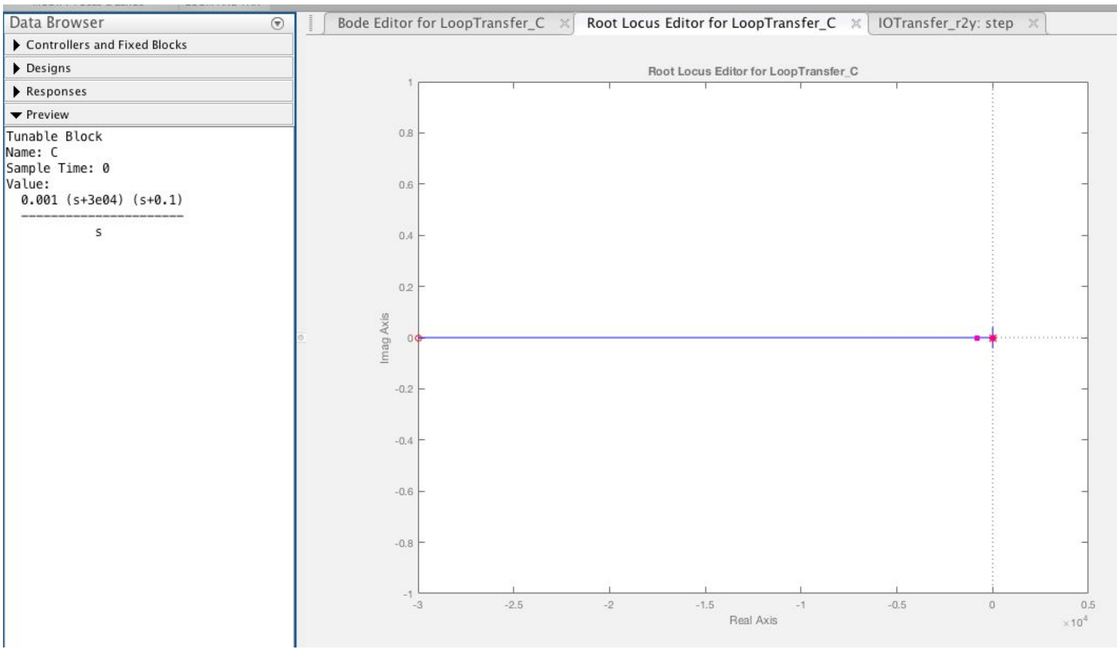


Figure 3.3.1 Sisotool with an integrator and two poles

Zooming in:

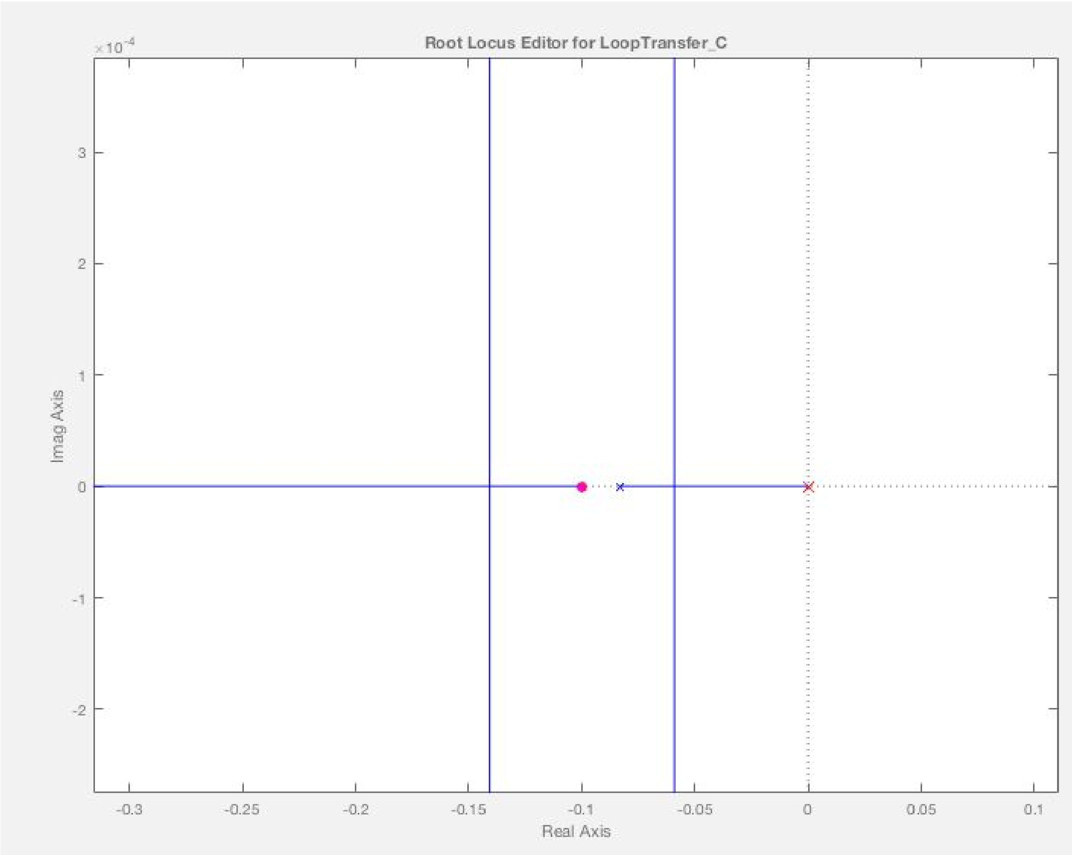


Figure 3.3.2 Zoom In of figure 4.3.1

Now, about the error requirements, as we have a PID, when we have a constant reference, the error is 0. However, we are interested in the ramp.

We know our Kv=3*325 (3 from PID and 325 from transfer function). So our ess=1/Kv=0.001, but our reference slope is 10, so our effective error will be 0.01, which is smaller than the required 2 degrees. We remove the noise from the graph to corroborate this:

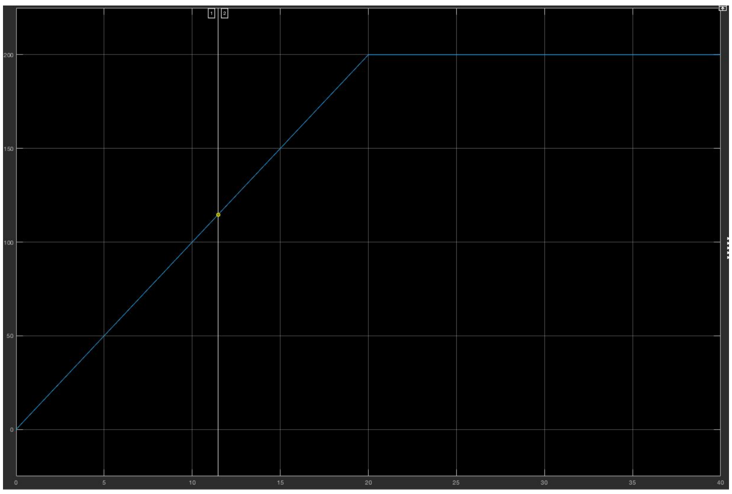


Figure 3.4.1 Output for the Heating System

Zooming in:

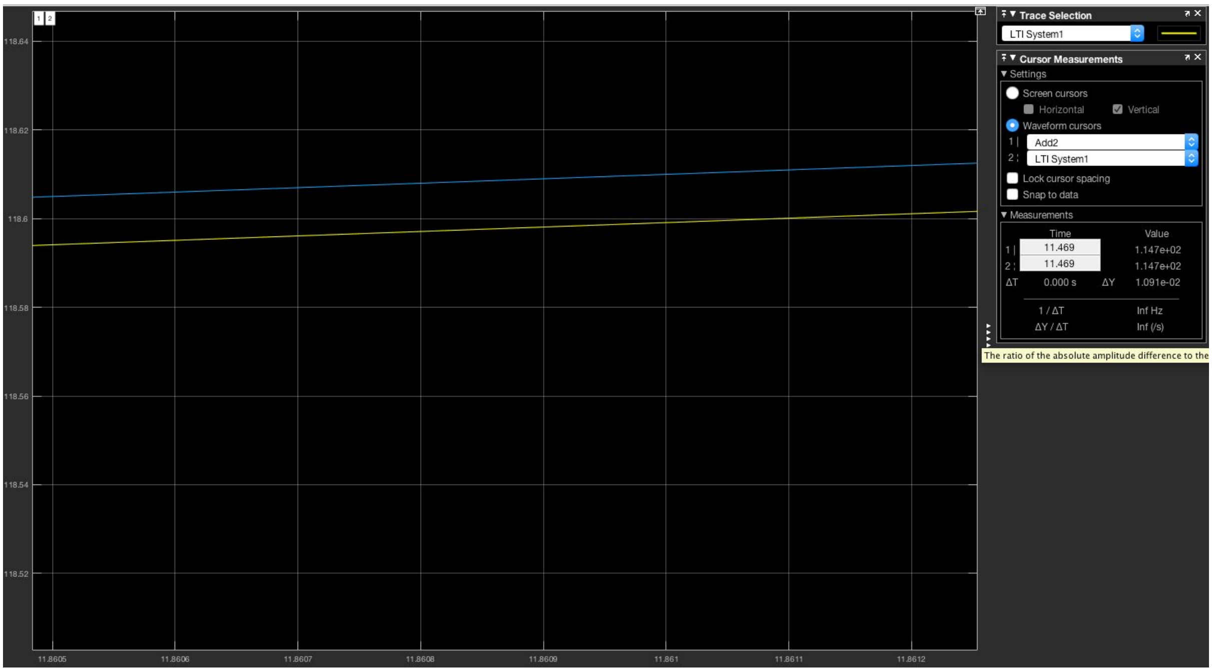


Figure 3.4.2 Zoom In of Figure 4.4.1

Finally adding the noise again we have the desired behavior for the system:

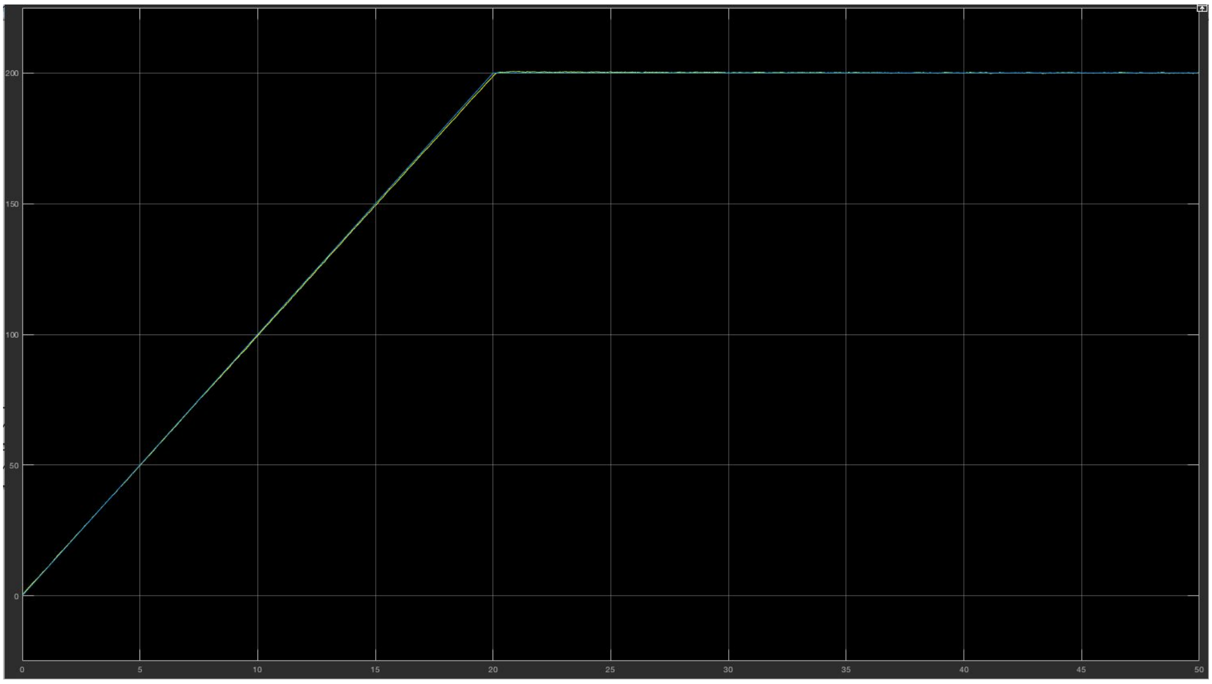


Figure 3.5.1 Output of the heating system with noise

Even adding the noise again we don't exceed the 2 degrees maximum error:

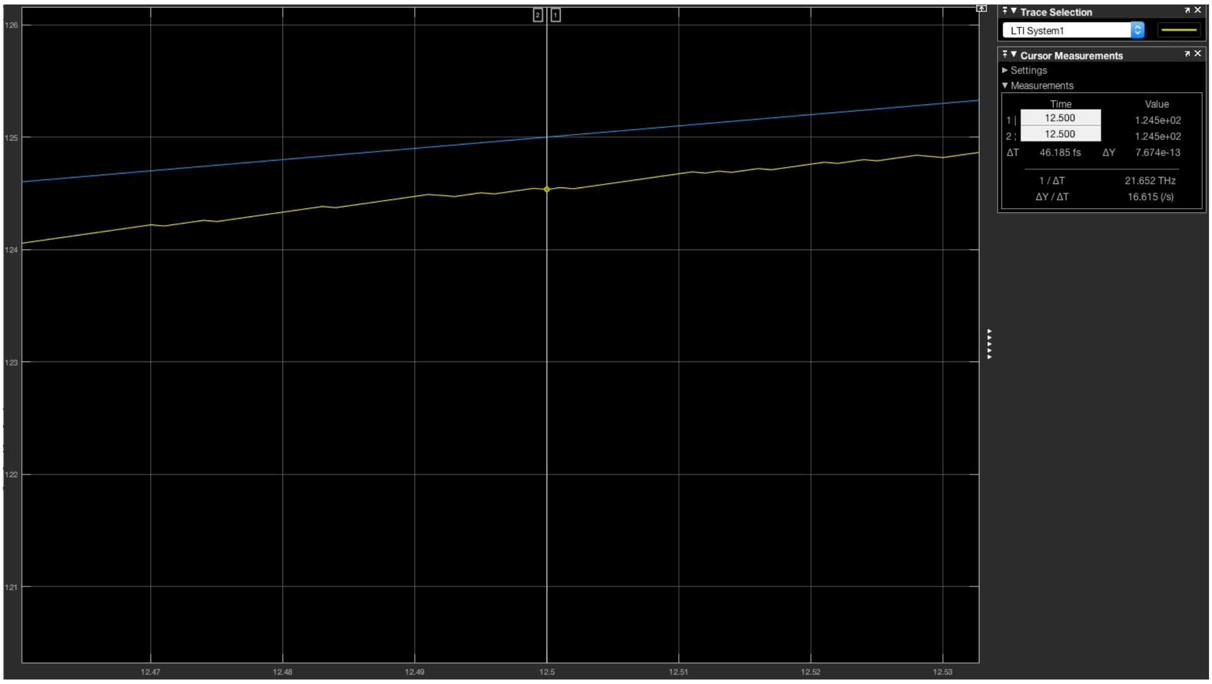


Figure 3.5.2 Zoom In of figure 4.5.1

Cooling Station

Description and Characterization

Like the other stations, the cooling station is crucial for the finishing touches of the processed beer. Now all the microbacteria has been removed and the last step is to cool the wort. There are many things to take into account for this process. For example If the wort is cooled slowly, dimethyl sulfide will continue to be produced in the wort without being boiled off this causes off-flavors in the finished beer. The objective in this station is to rapidly cool the wort 190 C before oxidation or contamination can occur.
For the cooling station the transfer function is obtained using the same process. From the requirements it is mentioned that ks at 25% is 60, from the rule of three ks at 100% is equal to 240, and the time to get to steady state is 40 seconds.

$$k_s = 240$$

$$t_{ss} = 40\ s$$

$$\tau = \frac{40}{5} = 8$$

The next step is to substitute:

$$G = \frac{k_s}{\tau s+1} = \frac{240}{8s+1}$$

Figure 4.1 Cooling station

Control simulation and analysis

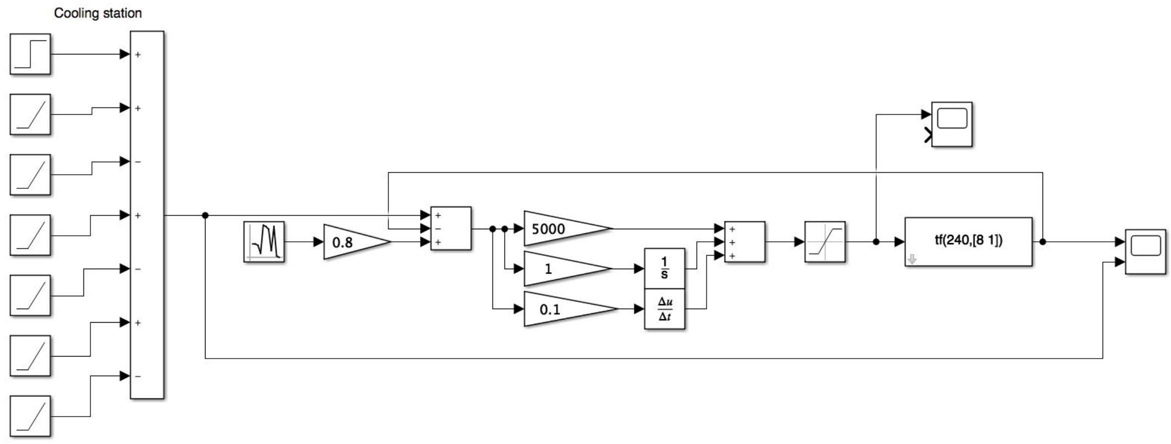


Figure 4.2 Simulink Model (Cooling Station)

The noise factor for this station is 0.8. We have a saturator at -1 (because our reference will be towards negative in this case, to avoid the plant to work above 100% percent its capacity. After a bit of trial and error (especially to avoid some derivative errors in matlab) we came up with the following PID:

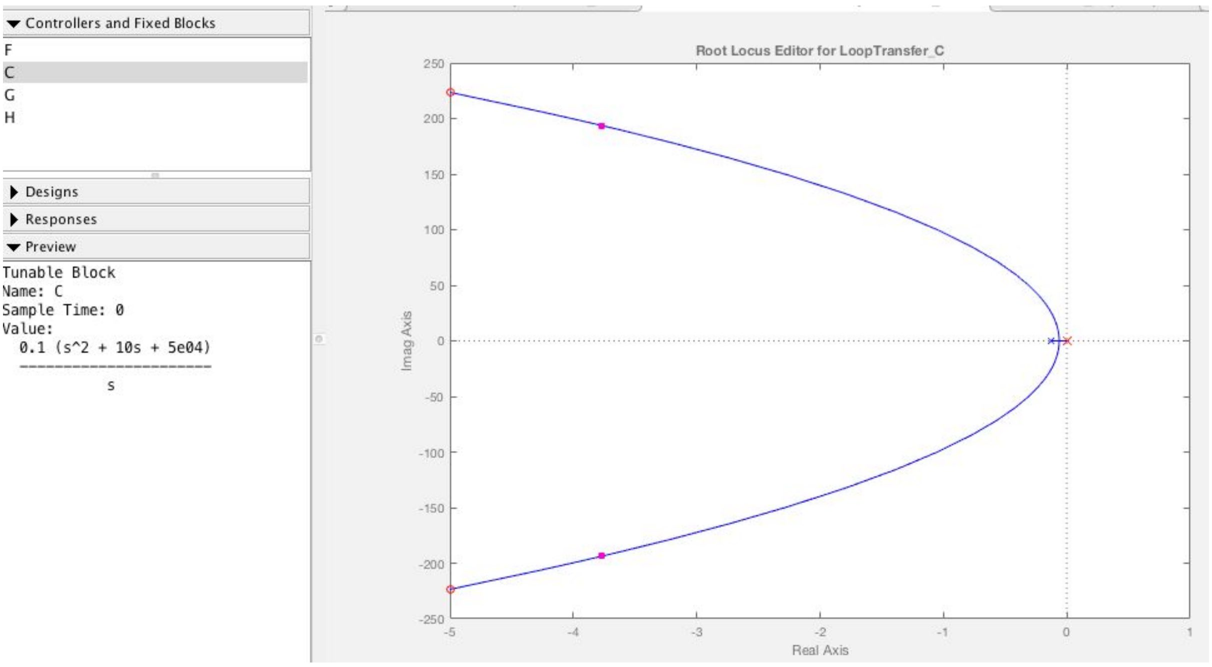


Figure 4.3 Sisotool with an integrator and two poles

To our surprise, it has complex zeros. However, we corroborated and our gain is still in the left side of the imaginary axis, and it works to give the desired behavior.

Now, to have the same reference as the given one by the professor, we have 3 different slopes (none of them exceed the -5 degree per second required).

Now, about the error requirements, as we have a PID, when we have a constant reference, the error is 0. However, we are interested in the ramp. The slides didn't have an error requirement for this, but we will check that the error doesn't exceed 2 degreed on each ramp:

Ramp 1:

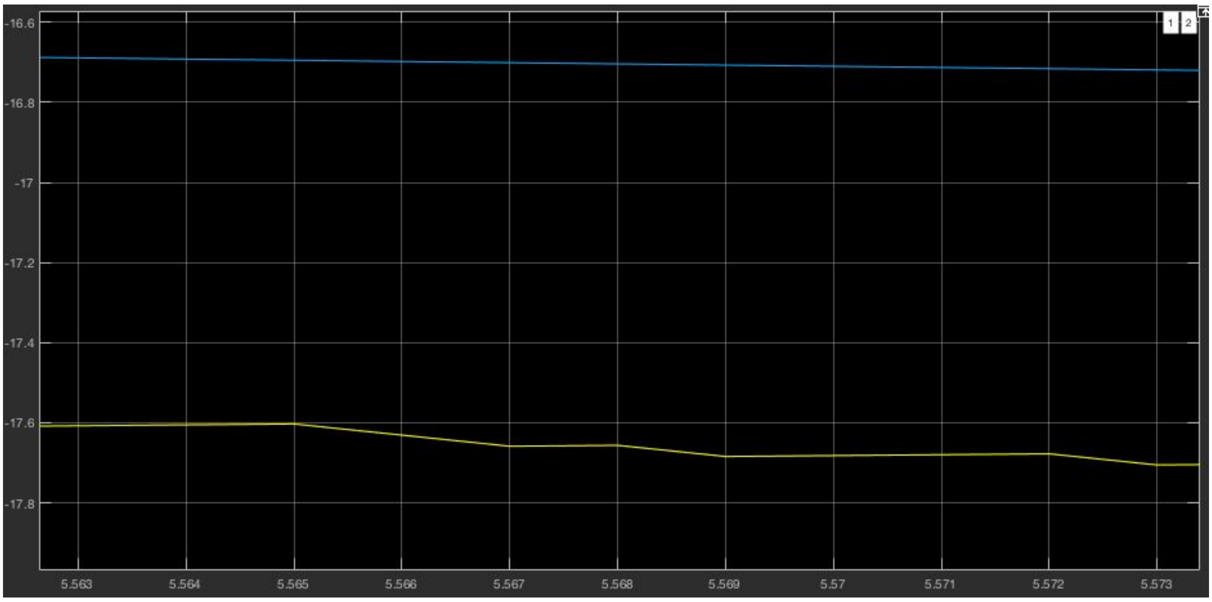


Figure 4.4.1 Error for Ramp 1

Ramp 2:

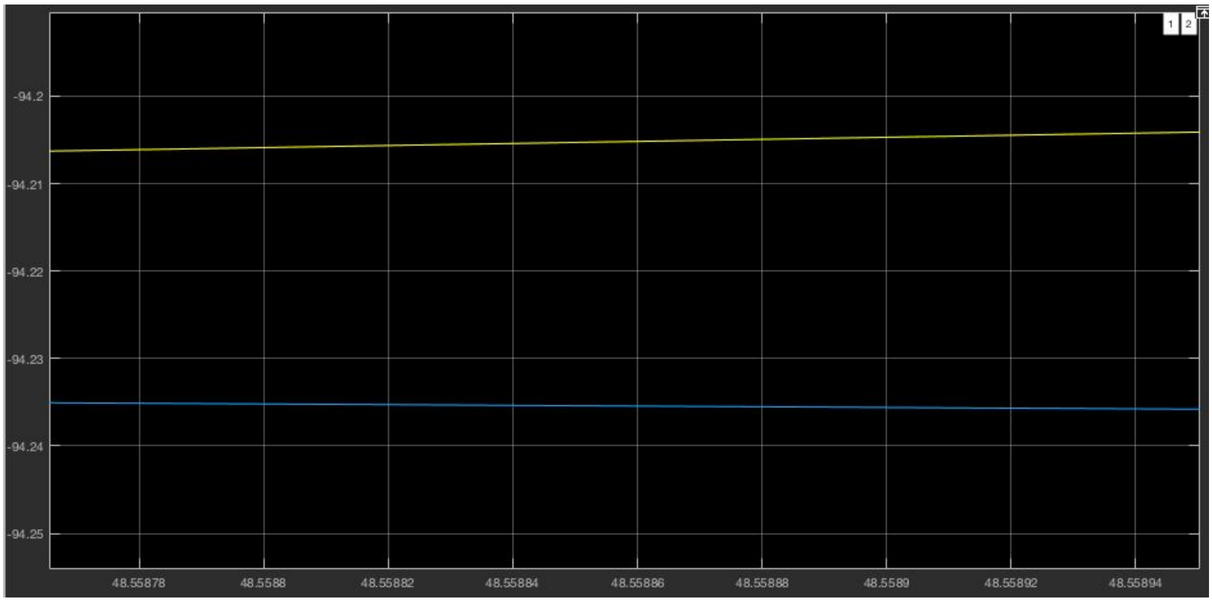


Figure 4.4.2 Error for Ramp 2

Ramp 3:

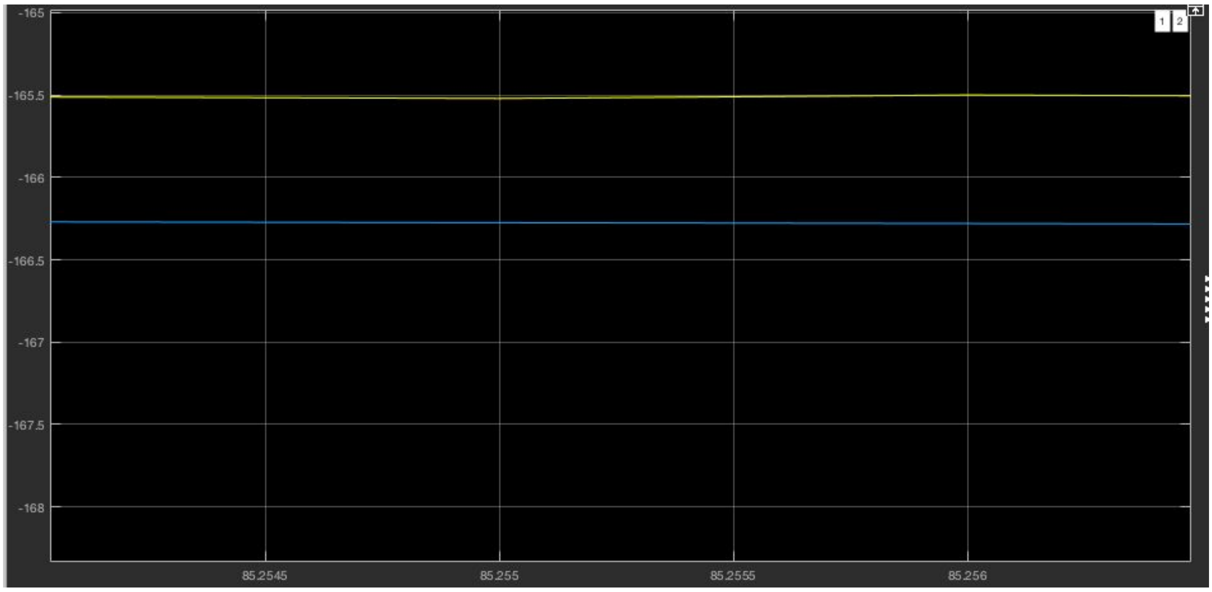


Figure 4.4.3 Error for Ramp 3

And all 3 cases we are on range. So our final behavior:

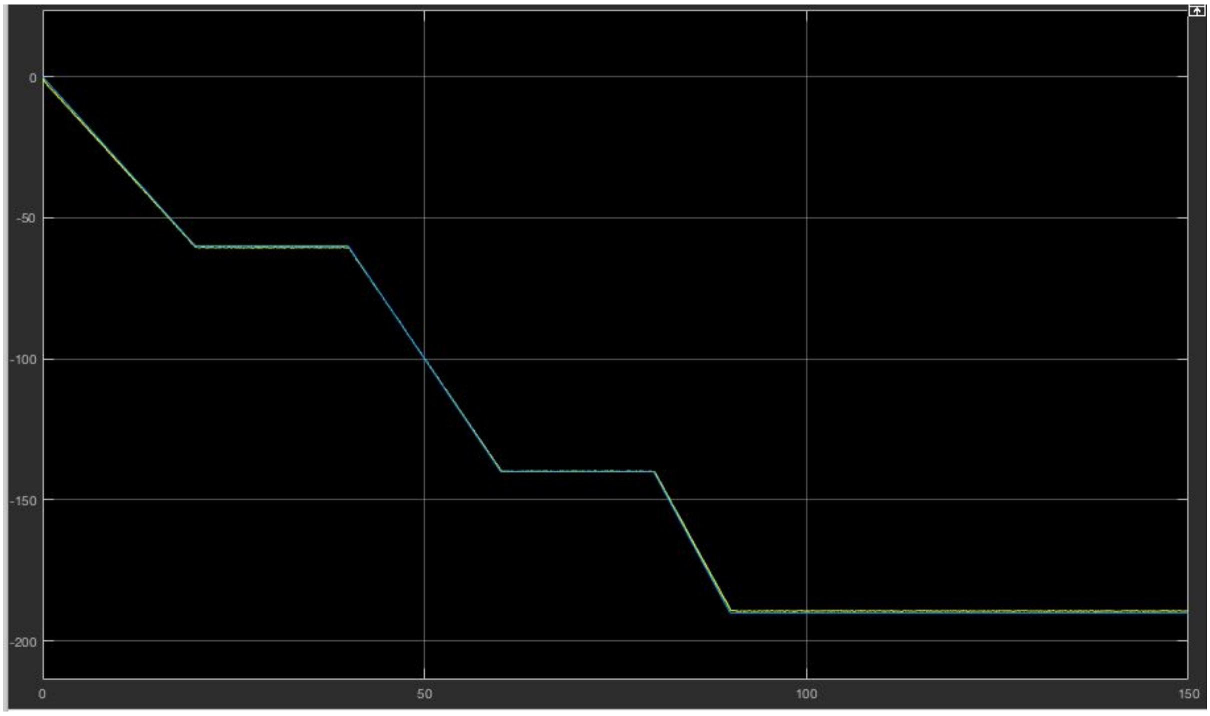


Figure 4.5 Final Behavior

Notice that we start in 0, because we are assuming initial conditions of 0. However, this represents that whichever our input is, it will be reduced by 190 degrees. So our bottle that comes with 200 degrees from heating station, will be reduced to 10 degrees.

Packaging Station

Description and Characterization

The packaging station for this production line includes two factors. One of them is placing the label in the bottle with the logo of the company and the other is to close the bottle with a cap. For this to happen we need 2 DC motors with two different movements. For the label to be placed in the bottle we need a rotational movement. To close the bottle we need a displacer to move in vertical mode. In other words we can make a system with a sine wave including angular position, angular acceleration and angular velocity.
On the packaging station it is much easier than the previous ones, since the project requirements already gives us the matrix A, B, and C in canonical form what is left is to arrange it in matlab as follows:

```
A=[0 1 0;
   0 0 1;
   0 0 0];
B=[0 0 0.2]';
C=[1 0 0];
sys=ss(A,B,C,0);
K=acker(sys.A,sys.B,[-1,-2,-3]);
```

The poles added to k can be chosen randomly, in the end the poles will determine the speed and functioning of the system. With this it is easy to obtain the vertical system, for the rotation system the last thing needed is to choose the L, which is gotten by adding the following to matlab:

```
L=acker(sys.A',sys.C',[-4 -8 -12])';
```

It is important to say that the poles of L need to be at least 3 times bigger than the poles chosen for k.


Figure 5.1 Packaging station station

Control simulation and analysis

Rotation movement:

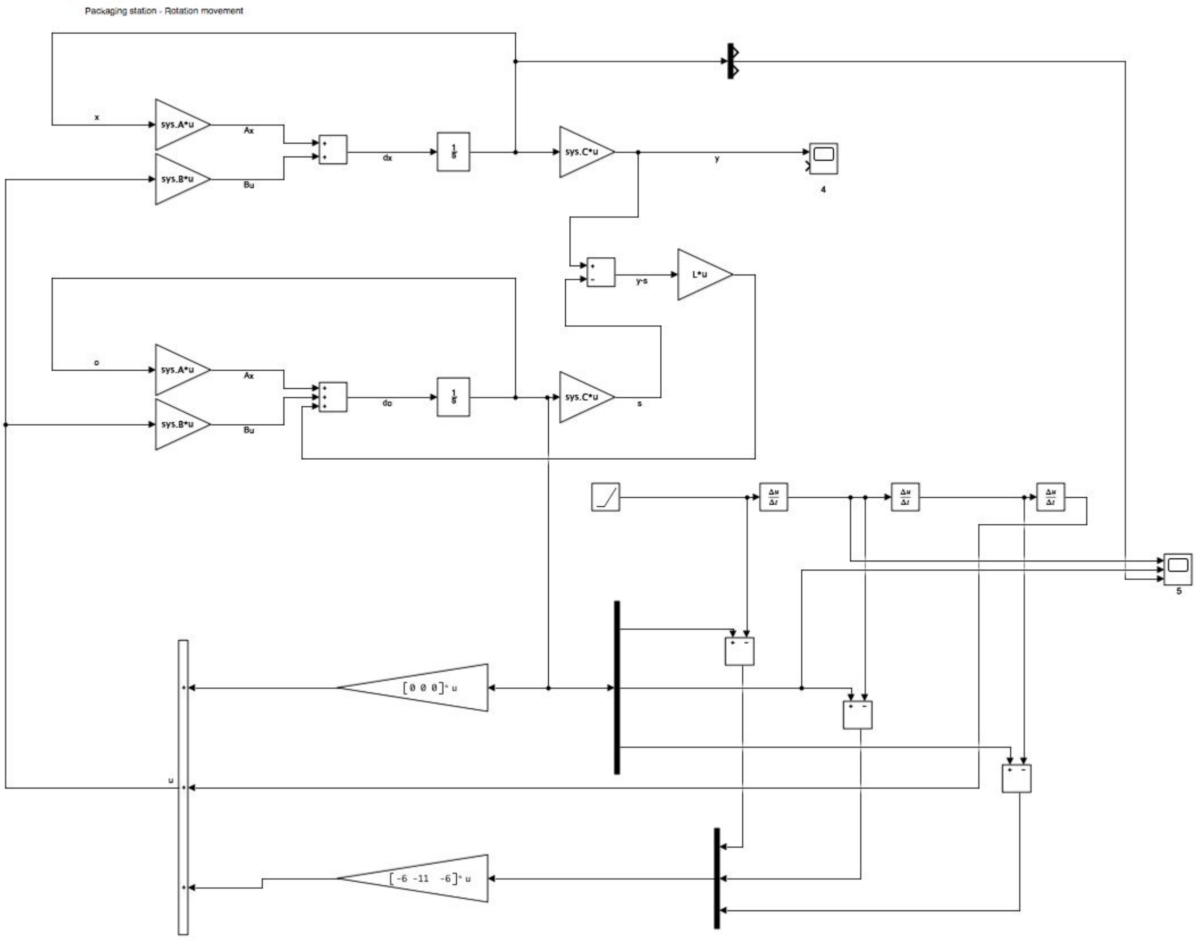


Figure 5.2 Simulink Model (Packaging Station Rotation Movement)

We cannot measure the state variables. So the observer structure is added. We verified the usual requirements for this:

```
%controllable canonical form
A=[0 1 0;
   0 0 1;
   0 0 0];
B=[0 0 0.2]';
C=[1 0 0];
sys=ss(A,B,C,0);

%controlability
Contr=[B A*B A*A*B];
rank(Contr); %3 in this case so it is controllable
%observability
Obs=obsv(sys.A, sys.C);
rank(Obs); %3 so in this case it is controllable
```

As C=[1 0 0], it means our output shows only x1. However, we want the motor at 50rpm which is angular velocity and this is our variable x2. Therefore, we need x1 to follow a ramp of slope 50 and the x2 is a constant 50. That is why we added the known structure for a time varying reference.

We verify the behavior on x2:

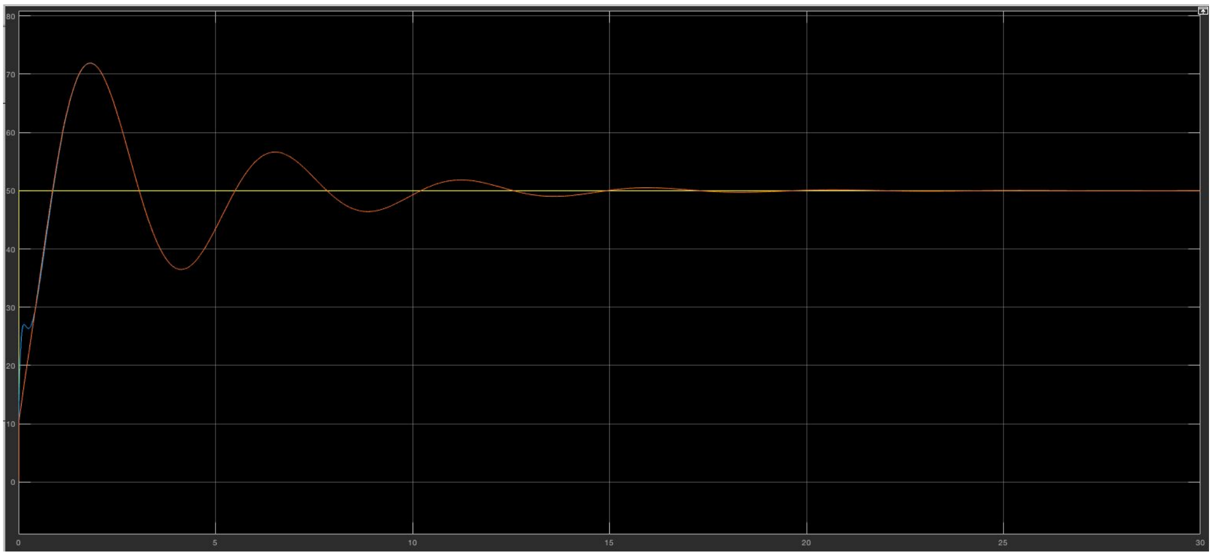


Figure 5.3 Behavior for x2 (RPM)

The yellow signal is the derivative of the ramp we want to follow (50), the blue signal is x2 from our observer and the red signal is x2 from the real system (not supposed to be measure, but just to corroborate results).

Vertical movement:

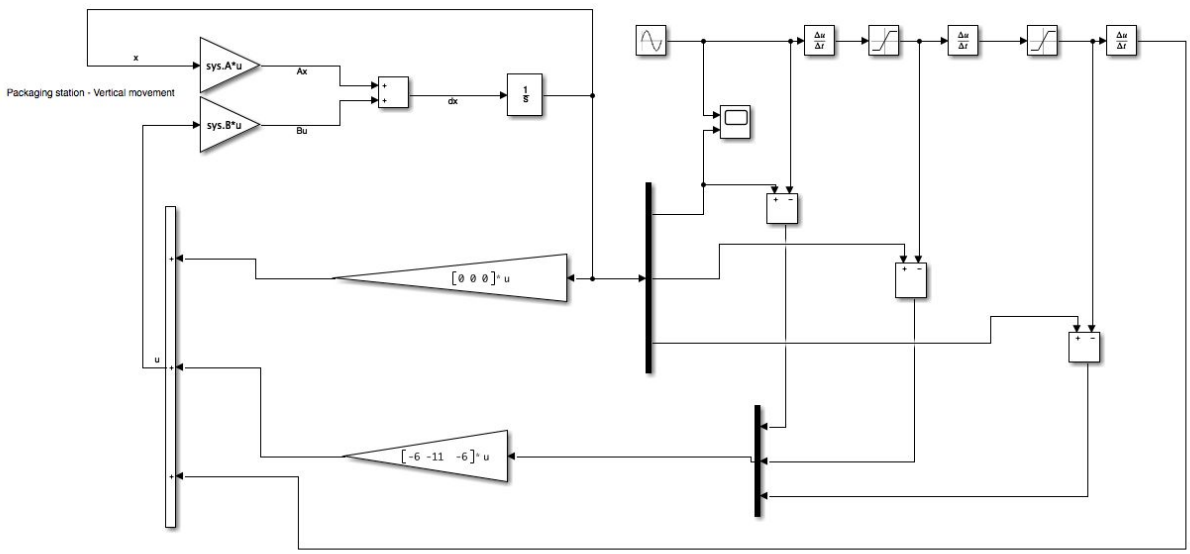


Figure 5.4 Simulink Model (Packaging Station Vertical Movement)

We also needed the structure to follow a time varying reference, but not an observer. We used the same K as the rotation movement motor. Now we verify the behavior (x1 against the reference or the output of the system against it; it is the same because C=[1 0 0]). It should be 5 periods of 60 seconds each and 20cm from max to min means a sine amplitude of 10cm:

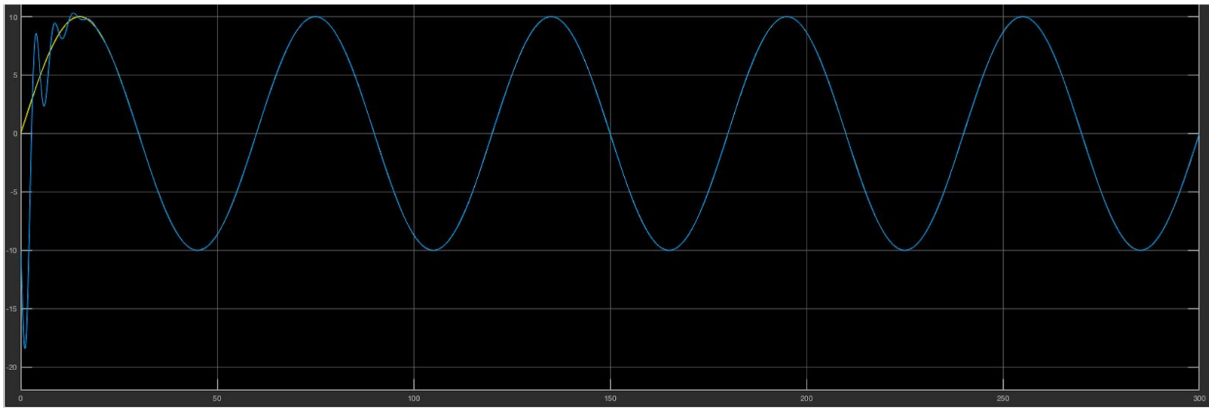


Figure 5.5 Scope with reference and x1

Displacers DC motors

Description and Characterization

The displacers are present in each one of the stations. The job of the displacer is to reach for the bottle in the conveyor and take it toward its station, then waits until the process is finished to return the bottle to the conveyor. We will be using the following FUYU belt drive linear guideway for each of our position movements and displacers.

Figure 6.1 FUYU belt linear guideway

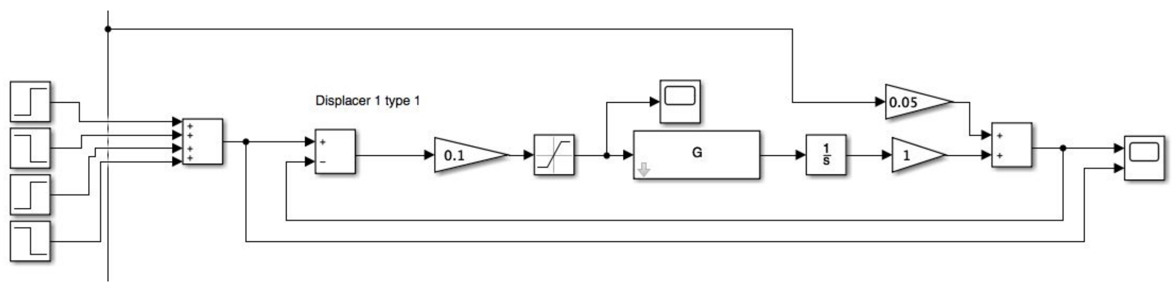
We will be using two types of displacers. Each one of them is specific for each station. After analyzing and knowing the function of each process and station we can proceed to chose the displacers that best fit for them.

Both displacers move up to 40 cm from the conveyor to the station and back to its initial position. Once the process of the station finishes it moves 30cm toward the station waits 2 seconds to grab the bottle and the returns the bottle to the conveyor.
As mentioned before we will have two types of displacer.

Type 1:
This first displacer moves 1 cm per revolution and has a noise factor of 0.05. As mentioned before, the stations that require this first displacer are the filling station and the heating station. We chose the slowest displacer for this stations since we know they are more delicate with their process. For example the filling station since it involves liquid we want to avoid any spilling and for the heating station we know the at the end of the process the bottle is boiling with the liquid and we want to avoid any damages.
Type 2:
This second displacer moves 1.2 cm per revolution and has a noise factor of 0.04. As we can see this displacer has a higher speed than the first displacer. The stations that will be using this displacer are the cooling station and the packaging station. After the cooling station, the now called beer is more stable which means we can increase the speed of the displacers. We can optimize time also in the displacer for the packaging station, so using this displacer is a good idea.

Control simulation and analysis

The motors have the same transfer function as the ones from the AS/RS, only with different noise factors and different relation between revolutions and displacement. We have two types:



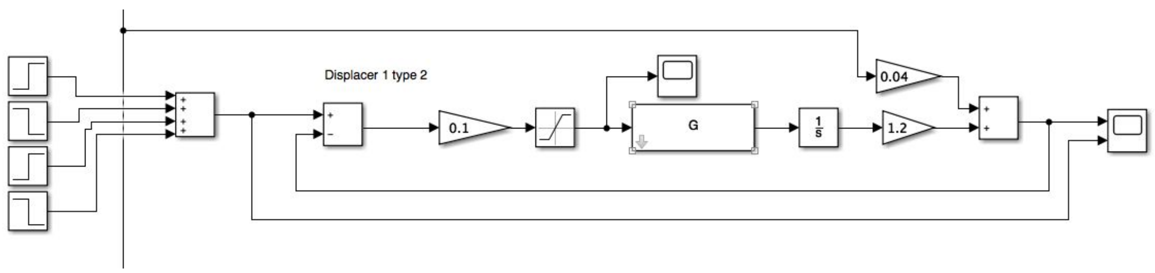


Figure 6.2 Simulink Model (DC motors for displacers)

For each station, we measured the approximate time it takes to complete the task:

```
delaystation1=1.5;
delaystation2=50;
delaystation3=125; %can be 150 like professor's graph
delaystation4=300; %5x60s period
```

So the behavior of these motors on each station is the following:

Filling station:

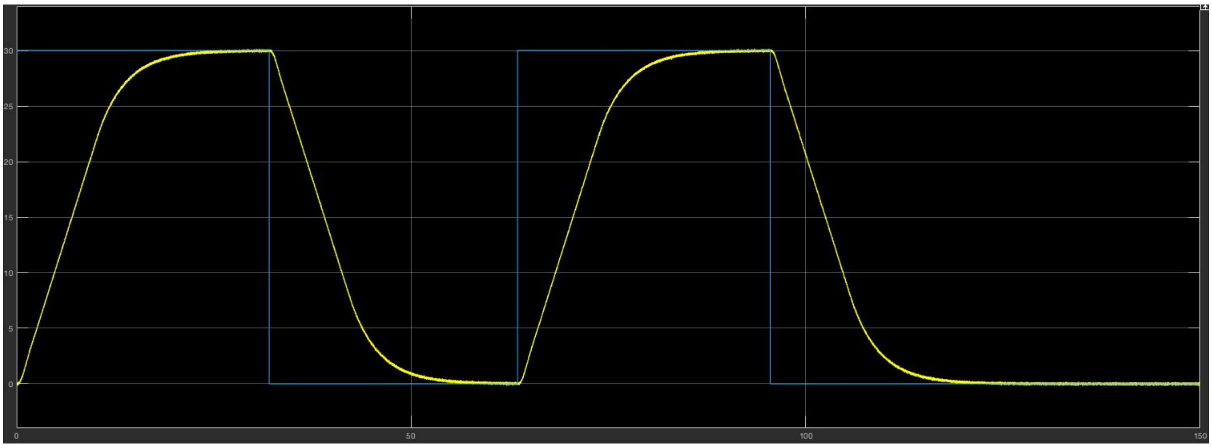


Figure 6.3.1 Displacement for Filling Station

Heating station:

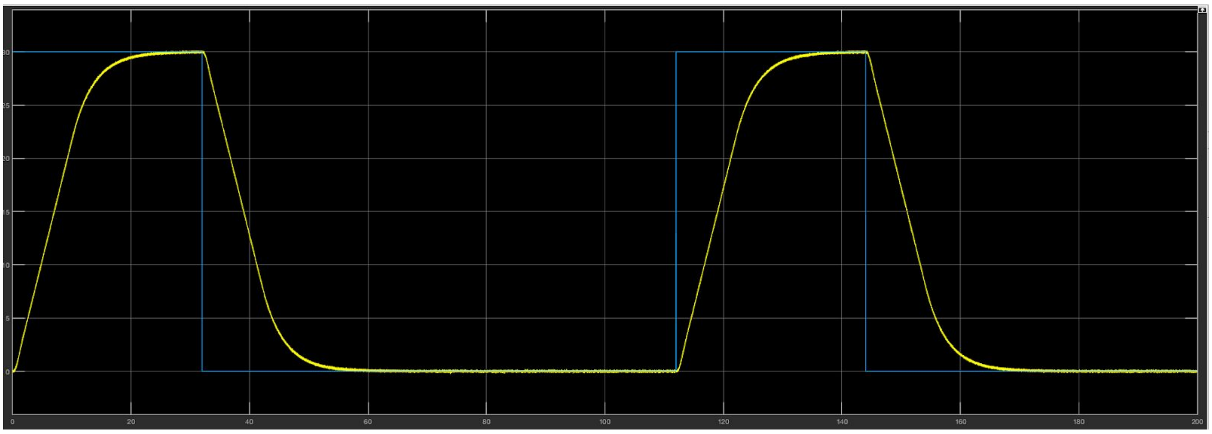


Figure 6.3.2 Displacement For Heating Station

Cooling station:

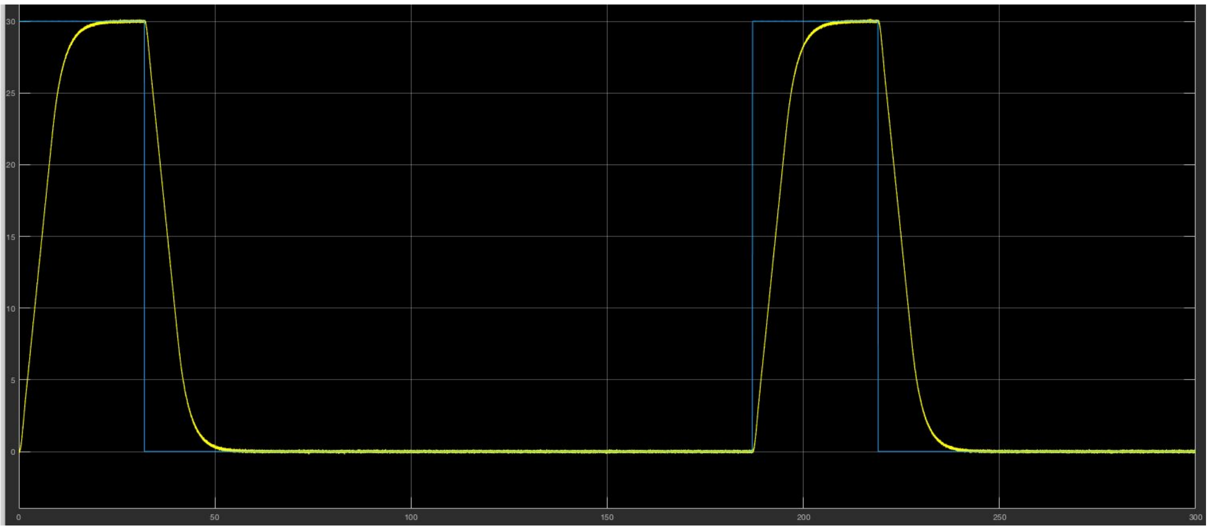


Figure 6.3.3 Displacement for Cooling Station

Packaging station:

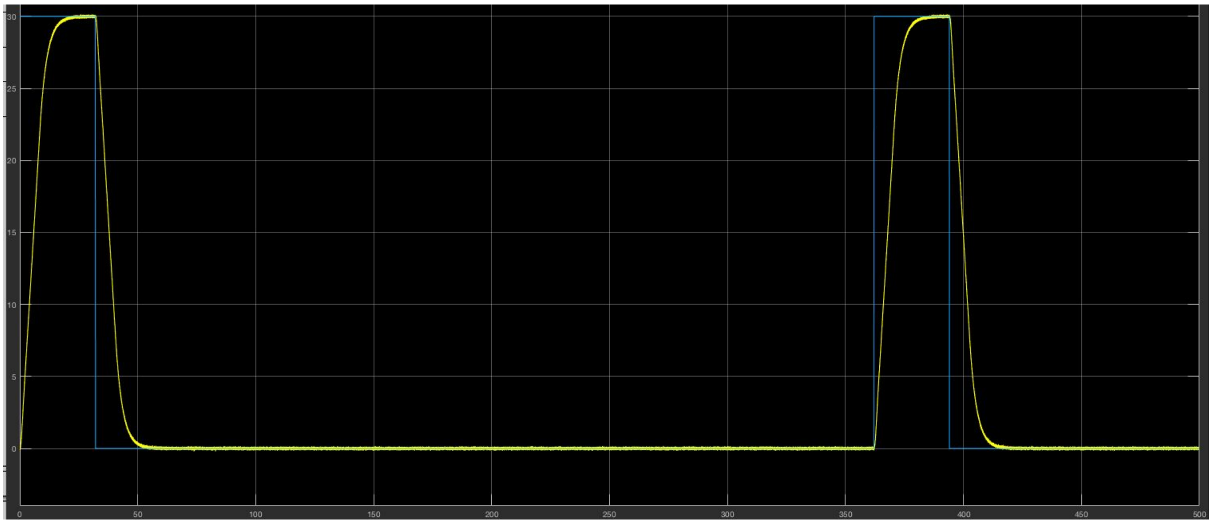


Figure 6.3.4 Displacement for Packaging Station

It takes about 30s for each motor to move the required 30cm, so the sequence we are observing on the graphs is:

- 30 seconds towards station (leave bottle)
- 2 second required delay
- 30 seconds back to conveyor
- Wait delay of station
- 30 seconds towards station (retrieve bottle)
- 2 second required delay
- 30 seconds back to conveyor

Conclusions

Luis Alfredo:
I believe the hardest part of this project was to interpret the requirements; to give them a physical interpretation besides the mathematical analysis. For example it took me quite long to imagine the movement of the 3 AS/RS in 3D space; my teammates spent quite a lot of time making sure I got it. Another thing that seemed complex was that we weren't given any of the transfer functions, but then we remember that from first partial we made algorithms to identify a transfer function given its step-response. Besides this things, the rest of our project went smoothly. It was all remembering and applying the control structures we already knew. It was good that we all did the 4th simulation project, so we easily knew what to do for the packaging station. One interesting part was adding saturators and analyzing the input to the systems. We usually didn't do this, but when applying this to real life it is really important to make sure we don't overload or don't have our plant at full capacity all the time. We discovered the tradeoffs between making the plant finish the process faster or not overloading it.

Once all the models were working, we decided to give them some synchronization, to make our project a bit more realistic. The AS/RS was the model that needed more measurements to be calibrated. We also wanted it to be somewhat automated, so we made some Matlab code to handle the displacements and delay of the motors, according to which cell we want to retrieve. I think this was my favorite part of project.

Vanya:
With this project we put in practice the theoretical background seen in class. It was very curious how we had to interpret each one of the stations and we had to adjust the gains in order to fulfill the requirements. I think this project summarized the past simulation projects all in one, but instead of following just instruction we had to use our logic to understand the behaviour of each subsystem and how we had to unite them all together in order for the timings to have coherence. In order to understand the AS/RS it was very helpful to watch a video of how the system worked in order to implement it correctly. We tried to be careful with all the small details with the timings and measurements, for example to grab the bottle exactly in the middle of each cell and to line up the displacers of each station.
Once we understood what had to be done in each station, it was easier to design them.
I think we all learned understood at a high level how beer is done. I really enjoyed how we put a meaning in each station in order to assure the specifications, for example for the packaging station we used the vertical movement in order to close the bottle and the rotating movement in order to place the label. Overall, I think we started the project on time and we dedicated the thinking and logic in each subsystem correctly.

Rodrigo:
The modelling of a system is something more complex than it may appear, at the beginning our team was a bit confused because we did not know if what we did was correct due to the lack of experience we had. At the end of the project and after a lot of consultancies we managed to fix and get the correct transfer functions in the end. But with this we not only learned to model the system but we also managed to understand a bit more of how gains work and how it can affect the input and output of the system, for example on the DC motors by changing the gain from 1 to 0.1 we could manage to lower the impact on the system, which made the input a lot more stable making sure that the motors don't have to ensure a big change ensuring that this motors can work for a lot longer without the need to replace or fix this motor.

Finally it is important to make sure that all the systems can work together, by this I mean that the times of the system should be aligned all the way through the production line in order to make sure that this works properly which means that in order to make a system completely functional not only do we need to model the system and design its controllers but to align each and every one of the systems together as we did on the AS/RS system and the displacers.

Video

https://drive.google.com/file/d/1abx85HJSU9NQSRniAg8-R_r-WWjo_f_0/view?usp=sharing