Final report

Inés Alejandro García Mosqueda A00834571

Oliver Josel Hernández Rebollar A01641922

Paul Enrique Alonso Ramírez A01634608

Analysis of Control Systems

Group 401

Tecnológico de Monterrey, Campus Guadalajara

October 21, 2023

Github Repository: https://github.com/Ineso1/Linear-Axis-Control/tree/main

Introduction

In this report we shall address the construction of a plant. Such a plant is a linear axis which corresponds to a major component in several industrial machines. Furthermore, it is known that a plant by itself is not useful if we cannot operate it. The main scope of the course is control theory. Classic and Modern. We did not cover modern control theory, thus we use one of the most robust and iconic ways of controlling devices, namely, PID controllers. A PID controller is supposed to modify the system according to a given reference. Naturally, a system has to be able to identify its current state; therefore, it must receive some sort of feedback. This is known as a closed-loop system because it is constantly comparing a current state with a desired state and performing the corresponding modifications to achieve the aforementioned state. The PID controller is composed of three parts: proportional, integral, and derivative. Each of these parts play a crucial role in the actuating action because they compensate for disturbances. For instance, if one exclusively uses a proportional controller, it would be impossible to reach the reference because it is mathematically impossible. Thereby, we add an integral component which allows us to approach the reference, but slows down the steady state time. If the system experiences abrupt disturbances, we must add the derivative part which smooths aggressive changes in the control. The three parts have a special purpose. However, it is noted that all of them come with certain costs. To obtain a desired controller, one must carefully tune the gains based upon a given set of performance characteristics that will be determined by the application or customer.

All of these things will be developed in the remaining sections of the document. The main idea is that the reader, if desired, can replicate what we have done. Also, it should be mentioned that our implementation is not perfect but tolerable. This means that if someone intends to build a plant, it is recommended that he or she looks for ways to improve. We thank the professors of the course for their support during the development of the linear axis

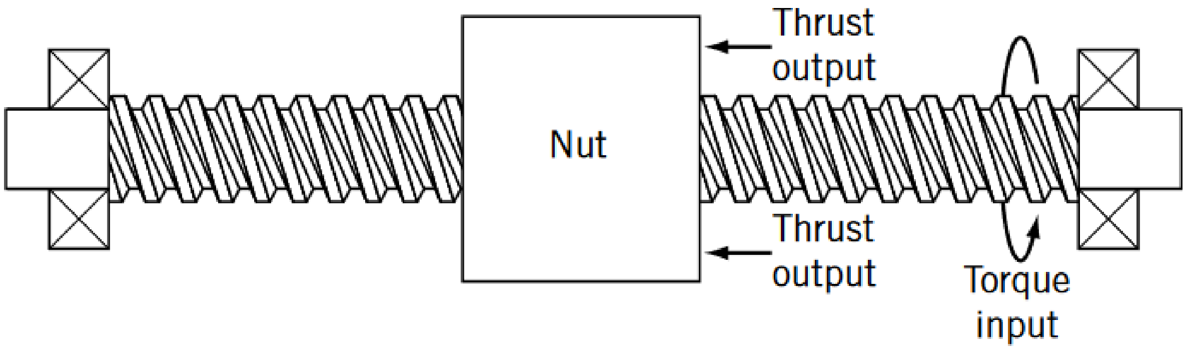
Objective

The objective of this document is to demonstrate one of several applications that the control apparatus has. We have restricted the application to classical control theory on a linear axis which is a component from a CNC or computer numerical control. We shall use a PID controller that was briefly described in previous paragraphs. Also, we apply basic methodologies from other disciplines such as mechanical, electrical and software engineering. Likewise, we used some tools from digital control because it is known that one cannot simply port a controller in continuous time to the microprocessor. Thus, one has to provide an analogous representation for discrete time. Lastly, we expect to extend the control techniques to complex electromechanical systems.

State of The Art

Computer Numerical Control is a way to instruct manufacturing machinery with a computer. It is a method widely used in the operation of industrial robots. Commonly, the instructions for the machinery will come from software known as computer-aided design. One basic type of CNC is point-to-point. Here the tool executes a set of primitive instructions such as carving a path given its initial and final point. Memory is required to keep track of the performed commands (Britannica, 2023).

Linear actuators provide, as the name suggests, motion in a straight line. There are several ways in which one can achieve linear motion. However, we shall devote our attention to the conversion from rotary motion to linear motion. A way to achieve this is by using screw threads to a rotary motion source. The present project utilizes a lead screw. A lead screw has helical threads that are designed for minimum backlash to allow precise positioning. The rotary motion of the lead screw is translated into the nut’s linear motion. The torque required to move the nut will largely depend on the nature of the application (Figliola & Beasley, 2011). The following image better depicts the given description.



Theoretical Framework

As mentioned, lead screws are a way to transmit power in modern machines. They can generate high forces with a small moment. Also, they are known for providing a really good level of motion precision. Thus, we see them extensively in applications that use linear actuators or linear stages. Typically lead screws can operate in two different ways. The first one is by supplying power to the nut and the other one is by rotating the shaft. Thereby, transferring power to the nut. In our project, the second way was used in the project. In other words, the nut's rotational motion is restricted and the screw shaft rotates. As a consequence the nut moves along the screw axis, namely, converting the rotary motion into linear motion. There are a few advantages of lead screws worth noting. Overall, they are cheap and reliable as their construction is simple; they require little to no maintenance; operations are smooth and reliable; load capacity is high (Slid, 2022).

H-bridge is a special circuit configuration that allows control over the current source to a load. This is specially important in motor applications because it is possible to change the behavior of the motor with an H-bridge. One must be careful always guaranteeing the correct H-bridge configurations because it is possible to cause a short circuit and damage the component (Franz, 2023). Operation in an H-Bridge is with switches. When using H-Bridge, one must consider at all times the switches characteristics and the thermal management of the circuit. Ignoring these properties can lead to operation failure. As mentioned, H-Bridges are heavily utilized in motor applications, since they can be mixed with the Pulse Width Modulation, hence, controlling the motor's speed (Electricity - Magnetism, n.d.).

An encoder is a sensor that retrieves physical variables. As with almost every known sensor, encoders convert what they sense into an electrical signal that can later be processed and handled by a piece of software. These devices help to determine the position, speed and direction. This is achieved by a count that the encoder generates and sends it to a controller. There are several technologies that an encoder can use to create a signal; for example, mechanical, magnetic, resistive and optical. Optical is the most common way. Specifically, The light emitted from a source is periodically interrupted by a disk with holes. These interruptions are the ones we use to generate the pulses and then know things like the direction of the encoder, speed, among others (Encoder Products Company, n.d.).

Construction of The Plant

Mechanics

The linear shaft is a long rod-shaped tool used to support linear motion, especially sliding motion. Without it, we wouldn't complete a sliding mechanism with linear power transmission.

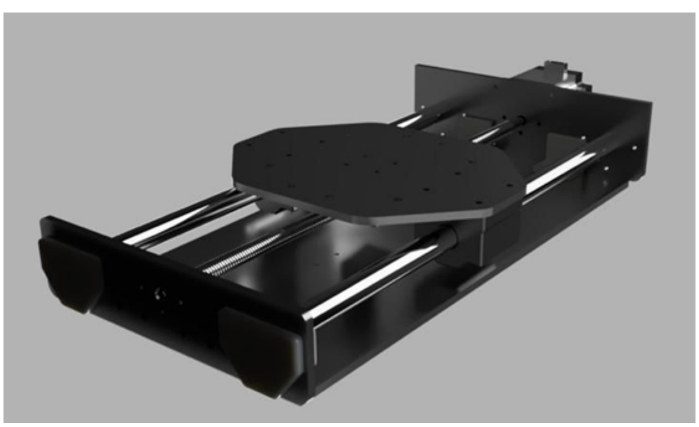
However, manufacturing and selecting the right linear shaft can be complicated. Linear shafts can have different structures and be made from various materials, depending on the required properties. With a linear shaft, motion can be guided and adjusted.

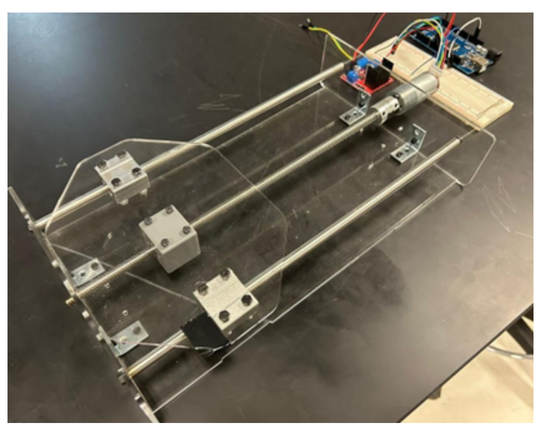
The load and specific requirements determine the size and precision of the linear shaft. In linear power transmission mechanisms, the shaft works alongside an actuator, a support rail, bushings (ball bearings), etc. In linear power transmission, the actuator typically does all the sliding work, while the rail, bushings, and linear shafts provide support.

The screw rotates clockwise, causing the shaft, which is essentially a nut on the screw, to move up and down the screw as the screw rotates. This is what converts the rotational motion of the electric motor into linear motion. Linear motors have been designed to produce high force at low speeds and even when stationary. Their design is not based on power but purely on force.

The direct current motor, also known as the direct current motor, is designed to transform electrical energy into mechanical energy, causing rotary motion. This is made possible thanks to the magnetic field that composes it, which is perhaps the most important part of the whole equipment. However, it requires other components to operate effectively, such as the stator, which serves as mechanical support for the device and holds the machine's poles, which can be wound with copper wire around an iron core or permanent magnets. The rotor is a cylindrical instrument with a winding and a core, which are supplied with direct current through the collector formed by commutator segments. The commutator segments are made of copper and are in contact with the fixed brushes.

There are two ways to mount the linear shaft: continuous support and end support. Continuous support shafts are generally used for heavier loads, while those supported at both ends are used for lighter loads.





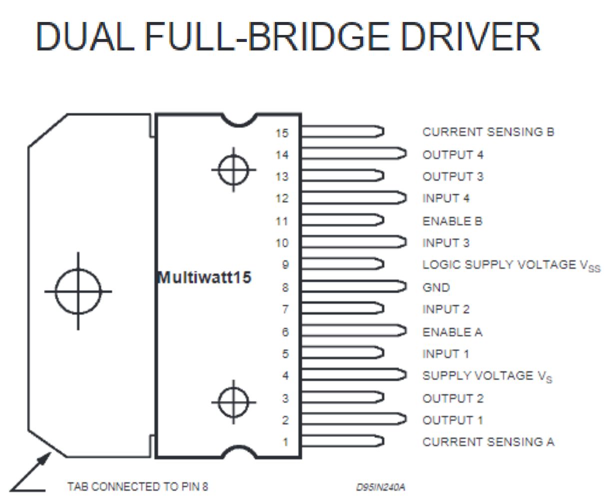
Electronics

In the realm of the Linear Axis, where precision is paramount, our electronic components serve as the central nervous system, orchestrating every facet of movement and control. This electronic symphony relies on a harmonious ensemble of key components, each playing a unique and indispensable role.

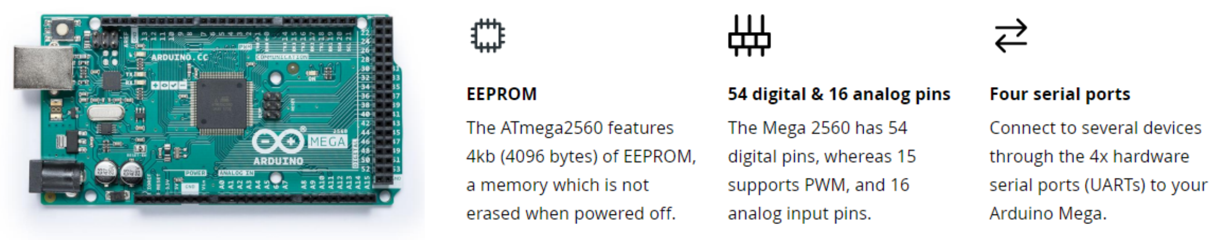
The L298N motor driver plays a critical role in our Linear Axis Control Project, serving as the primary component to translate electronic signals into precise physical motion. This integrated circuit, known for its versatility and reliability, acts as the conductor for the motor's performance.

The main function of the L298N motor driver is essential to our system, serving three main functions:

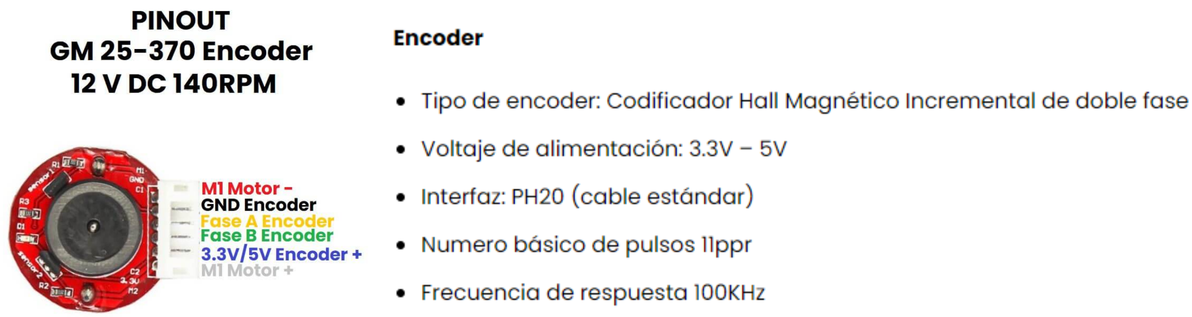
- Direction Control: The motor driver manages the motor's direction by reversing the polarity of the applied voltage to achieve forward or reverse motion.
- Speed Regulation: It allows us to adjust the motor's speed by varying the supplied voltage, providing us with the capability for precise speed control.
- Current Management: It incorporates current-sensing circuits to prevent overcurrent conditions, protecting both the motor and the driver itself.



At the core of our project's electronics is the Arduino Mega, a microcontroller board based on the ATmega2560. This microcontroller serves as the project's central processing unit, executing complex control algorithms and facilitating seamless communication with various sensors and components. The choice of Arduino Mega contributes to the project's user-friendliness and versatility, making it an ideal candidate for orchestrating the multi-disciplinary aspects of our linear axis control system.

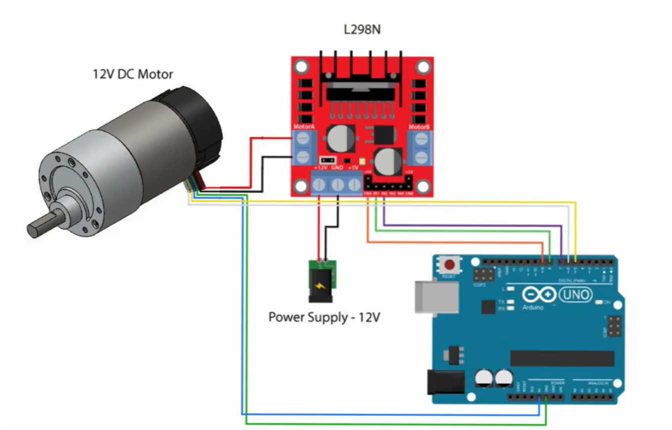


The motor-reducer assembly seamlessly integrates a motor, reduction gearbox, and encoder, a trio of components that harmonize to deliver precise control over the linear motion of our axis. At the core of this assembly is the motor, strategically coupled with a reduction gearbox. Together, they form a symbiotic relationship, with the motor providing the mechanical power required to propel the linear motion. The reduction gearbox, with an approximate reduction ratio of 1:45, transforms the motor's high-speed rotation into the slower, high-torque movement essential for our project's precision. Nestled within this assembly is the encoder, a sentinel of precision. This vital component employs Hall Effect technology and boasts a resolution of approximately 493.9 PPR (Pulses Per Revolution) with a ±10% tolerance. Operating within a voltage range of 3.3V to 5V and offering a response frequency of 100 kHz, the encoder delivers real-time feedback that is instrumental in our pursuit of accuracy. The motor, reduction gearbox, and encoder operate in tandem to orchestrate the linear motion of our axis. As the motor churns, the gearbox tempers its high-speed rotation, allowing for controlled movement. Simultaneously, the encoder diligently records every aspect of this motion—position, speed, and direction. This trove of data serves as the compass by which our system navigates, ensuring that the linear axis moves precisely as intended.



Standard Wiring Configuration for L298N

In the context of connecting and utilizing the L298N motor driver, a typical wiring configuration is employed to ensure optimal functionality. This standard connection scheme serves as the foundation for effectively controlling motors and is widely adopted in various applications.7



Software

The software we developed is quite simple because we were instructed to use an Arduino which is known for being user friendly. We wanted to increase the portability and robustness of our design, so we implemented a class called *PositionMotorControl*. This class has five methods for orchestrating the motor's behavior. The constructor will retrieve all the information for the member functions. Initialization method will send the constructor's data to the Arduino. Control method will be rotating the shaft based upon an error computed using the number of pulses. Set Target Position receives the desired position in the number of pulses. Get Current Position returns the current position attribute and Set Cero establishes what will be taken as the reference, namely, the origin. Attributes declared with the volatile qualifier are used in the interruption called Handle Encoder. This private function member reads the encoder in real time and updates things like current position and the motor's direction. It is declared as static and thus uses a pointer to be called. At the beginning of the main program we create a motor instance, then set the baud rate to communicate with the microcontroller. Lastly we enter the loop and read the Serial Monitor to receive input from the user, if that does not happen, we print the position. Later the control method is called and moves the nut to the desired position.

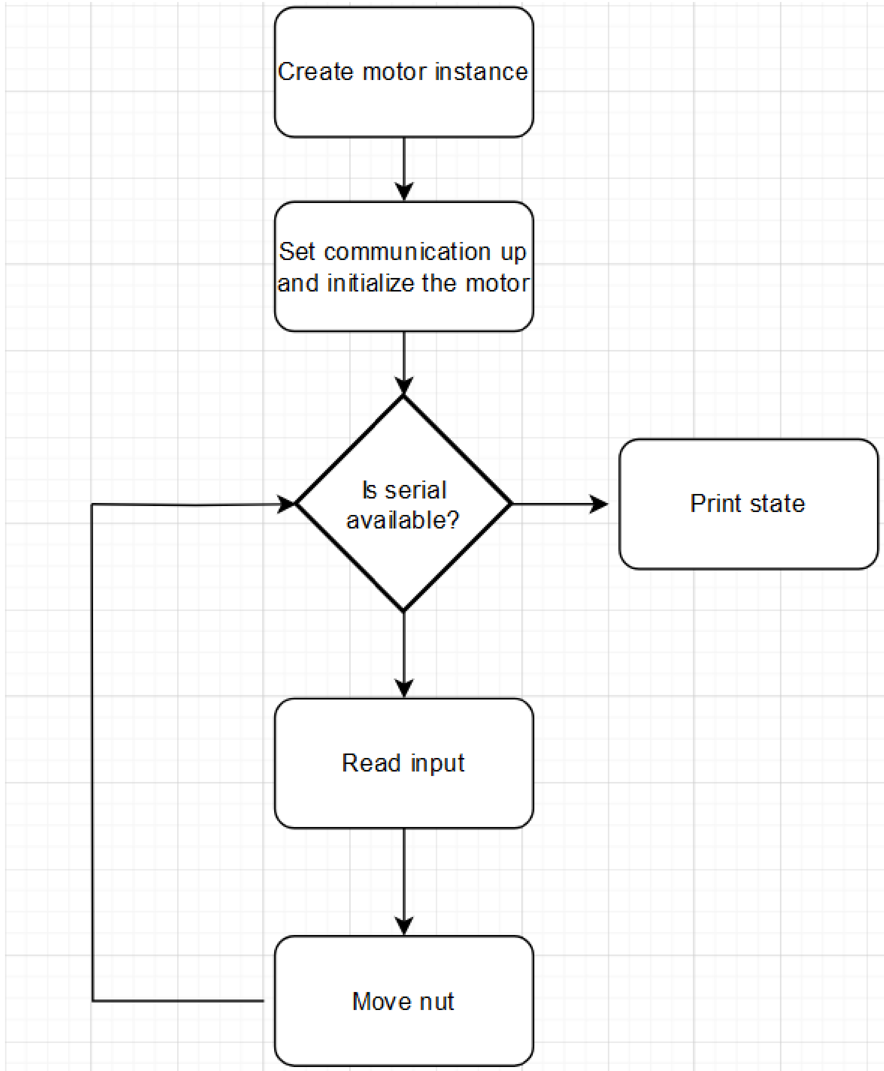
The reason of this implementation is:

- Initialization: One of the key features of the 'PositionMotorControl' class is its initialization process. We utilize the 'init()' method to initialize the motor and encoder pins, thereby configuring the necessary parameters for control. This step is vital in preparing the system for precise motion control.
- Control Methods: The class incorporates various control methods, notably 'control()' and 'controlPID()', which enable us to fine-tune the motor's behavior to meet our specific

needs. These methods play a pivotal role in ensuring the motor's responsiveness and adaptability to different control strategies.

- Position and Velocity: Within our software, we can set the target position, monitor the current position, and adjust the motor's velocity as required. This capability allows us to precisely control the motion of our linear axis, making it responsive to real-world demands.
- Error Handling: Error handling is an integral part of our software. We utilize variables such as 'error' and 'errores' to precisely manage and respond to any deviations from the desired path. These mechanisms contribute to the system's accuracy and robustness.
- Hardware Integration: The class seamlessly integrates with the hardware by working with motor pins, supporting encoder feedback, and incorporating mechanical constants. This integration ensures accurate and controlled motion. Mechanical constants such as gear ratios and encoder resolutions are considered to convert between pulses and real-world positions.
- Interrupt Handling: Our software employs interrupt-driven functions to manage encoder changes efficiently. By doing so, we enhance the overall efficiency of our system, enabling rapid response to position changes and contributing to the system's stability.
- Custom Libraries: To streamline and encapsulate the functionality, we have developed custom libraries, 'PositionMotorControl.h' and 'PositionMotorControl.cpp'. These libraries simplify the integration of motor control into our project and promote code modularity.

The following is a high-level description of the main program. The control method is responsible for rotating the shaft based on an error computed using the number of pulses.



Implementation of The System

Identification of The Plant

Identifying the plant means creating a mathematical or empirical model that describes the behavior of the system. This model helps us understand how the system responds to different inputs and disturbances. Without a proper model, it's challenging to predict or explain system behavior accurately. Knowing the plant is fundamental for designing effective control systems.

Engineers need to understand the dynamics of the system to design controllers that can regulate it optimally. Control algorithms often rely on an accurate model of the plant to determine control actions. This helps us to optimize the performance of the system. By understanding the plant's response characteristics, we can fine-tune control strategies to achieve desired performance metrics, such as stability, speed, accuracy, and robustness. Plant identification allows us to optimize the performance of the system. By understanding the plant's response characteristics, we can fine-tune control strategies to achieve desired performance metrics, such as stability, speed, accuracy, and robustness.

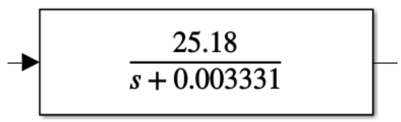
We implemented a systematic and data-driven process, leveraging carefully designed code to uncover the dynamic behavior of our motor.

Data Acquisition

The initial phase of our plant identification process entailed the collection of extensive data. We aimed to create a comprehensive dataset that allowed us to explore how the motor system responded to varying input signals. To accomplish this, we employed a controlled experimentation approach. We applied a range of PWM (Pulse Width Modulation) signals to the motor while simultaneously measuring and recording the corresponding encoder output. This process was carried out at consistent time intervals to ensure that we obtained a precise and reliable dataset. The data acquisition step was pivotal in capturing the dynamic response of our motor system under a variety of operating conditions. It allowed us to observe the system's behavior as it adapted to different control inputs, providing valuable insights into its characteristics.

Regression Analysis

With our dataset in hand, we turned to the tools offered by MATLAB's System Identification Toolbox. This software environment provided us with the means to perform advanced data analysis, model fitting, and system identification. Through the application of regression analysis techniques, we embarked on the task of approximating the plant function that governs the behavior of our motor system. This function represents the mathematical relationship between the input (PWM signals) and the output (encoder responses) of the system. Our objective was to develop a mathematical model that could accurately describe and predict the motor's behavior under various conditions. The regression analysis process involved the exploration of different model structures, identification of model parameters, and the validation of the resulting model against the acquired data. By iteratively refining the model, we sought to achieve a high level of accuracy in representing the dynamic behavior of the motor system.



Access to the Code Repository for characterization of the plant

For a comprehensive understanding of our project's code and implementation, you can access the complete source code and related resources in our GitHub repository. The code is available at the following link:

GitHub Repository - Code for Characterization of Angular Velocity:

https://github.com/Ineso1/Linear-Axis-Control/tree/main/ControlProgram/CaracterizacionVelocidadAngular

Syntonization of The Control

In the realm of control systems, achieving optimal performance often necessitates the careful adjustment and tuning of control parameters. This process, known as control tuning or syntonization, is a pivotal step in ensuring that the control system responds accurately and effectively to the desired setpoints and disturbances.

Importance of Control Syntonization

- Enhanced Performance: Accurate control parameter tuning can lead to improved system performance, enabling faster response times, reduced settling times, and more precise control of the system's behavior.
- Stability and Robustness: Proper syntonization ensures that the control system remains stable under various operating conditions and disturbances. It prevents issues such as overshoot, oscillations, and instability.
- Minimized Energy Consumption: Well-tuned control systems can operate more efficiently, consuming less energy and reducing operational costs.
- Adaptation to Changing Conditions: As real-world conditions change, the control system needs to adapt. Control syntonization allows the system to adjust to new parameters and maintain optimal performance.
- Improved Safety: In safety-critical systems, accurate syntonization is imperative to prevent dangerous and unexpected behavior.
- Control parameter tuning typically involves adjusting variables like proportional gain (Kp), integral gain (Ki), and derivative gain (Kd) in a PID (Proportional-Integral-Derivative) controller. The specific methodology and approach to syntonization depend on the control algorithm and the characteristics of the plant.

Methods of Control Syntonization

There are various methods and techniques for control syntonization, including:

- Manual Tuning: In this method, we adjust control parameters manually based on their knowledge and experience. It's a trial-and-error approach that may require multiple iterations to achieve optimal results.

- Ziegler-Nichols Method: This method provides systematic guidelines for tuning PID controllers. It involves identifying the ultimate gain and ultimate period from the system's response to a step input.
- Frequency Response Analysis: Using frequency response techniques, we can analyze the system's behavior to determine the optimal control parameters.
- Optimization Algorithms: Advanced techniques, such as gradient descent or genetic algorithms, can be used to automate the syntonization process. These methods aim to find optimal parameter values efficiently.

Our approach

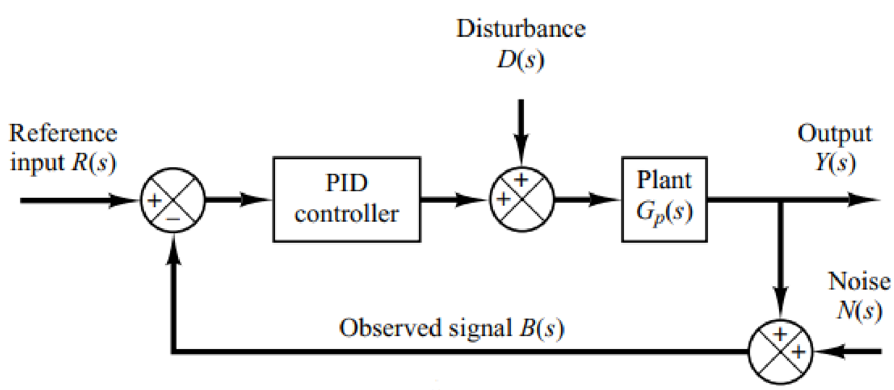
In our project, we employed an analytical approach for the syntonization of the PID (Proportional-Integral-Derivative) controller. The goal was to fine-tune the control parameters to achieve the desired system response. Our approach primarily centered around using the concepts of maximum impulse and steady-state time, and ultimately, partial fractions decomposition.

To begin the syntonization process, we identified the system's ultimate gain. The ultimate gain is the magnitude of the control signal that, when applied as a step input, induces sustained oscillations in the system's response. The maximum impulse is a crucial parameter for the syntonization of the controller. Concurrently, we determined the steady-state time, which represents the period of one complete oscillation cycle under the influence of the maximum impulse parameter. The steady-state time provides insight into the system's oscillatory behavior.

Syntonization Through Partial Fractions Decomposition

The ultimate gain and steady-state time values served as foundational information for the syntonization process. We leveraged these parameters to derive a transfer function that accurately represented the dynamics of the system. This transfer function was critical for tuning the PID controller. The syntonization process involved expressing the system's response as a sum of first-order and second-order transfer functions, represented through partial fractions decomposition. By decomposing the system into these simpler components, we could better understand and control its response. Using the derived transfer function, we adjusted the control parameters; Proportional Gain (Kp), Integral Time (Ti), and Derivative Time (Td to achieve the desired system response. The insights gained from the decomposition guided us in selecting appropriate values for these parameters to optimize the system's performance.

$$U(s) = K_p\left(1 + \frac{1}{T_i s}\right)R(s) - K_p\left(1 + \frac{1}{T_i s} + T_d s\right)B(s)$$



Advantages of Analytical Syntonization

- Theoretical Foundation: This analytical approach to syntonization is firmly grounded in control theory and system dynamics, providing a strong theoretical basis for tuning the controller.

- Precise Control: By using Ku and Tu to derive the transfer function and employing partial fractions decomposition, we were able to fine-tune the control parameters with a high degree of precision.
- Efficiency: The analytical approach is efficient and eliminates the need for extensive trial-and-error tuning, resulting in a more streamlined and systematic syntonization process.

In the subsequent section, we will detail the specific steps taken and results achieved through this analytical syntonization approach. This will provide a comprehensive understanding of how we optimized the control system to meet our project's requirements.

Certainly, you can include the second approach you used for PID tuning with MATLAB Simulink in your report:

Alternative Approach: MATLAB Simulink PID Tuning

In addition to our analytical approach, we employed an alternative method for synchronizing the PID (Proportional-Integral-Derivative) controller. This alternative approach involved leveraging the capabilities of MATLAB Simulink, a powerful tool for system modeling and control design.

Using MATLAB Simulink for PID Tuning

Model Representation: In this approach, we first created a dynamic model of our system within MATLAB Simulink. This model encapsulated the plant's behavior and its interaction with the PID controller. In this implementation we added a PID controller block to our Simulink model. This block allowed us to easily adjust the controller's parameters, including the Proportional Gain (Kp), Integral Time (Ti), and Derivative Time (Td). MATLAB Simulink provides a suite of automated tuning tools that simplify the process of optimizing the control parameters. These tools can be especially valuable when dealing with complex systems or when a quick and efficient tuning process is required. With the PID controller block and automated tuning tools in place, we conducted simulations to observe the system's response under various conditions. MATLAB Simulink allowed us to visualize and analyze the system's behavior, helping us identify the optimal control parameters. The flexibility of MATLAB Simulink allowed us to iteratively refine the control parameters until the system's response aligned with our project's requirements. We could easily adjust the gains and time constants in real-time during the simulation.

Advantages of the MATLAB Simulink Approach

- User-Friendly Interface: MATLAB Simulink offers an intuitive and user-friendly interface for control system design and tuning, making it accessible even to those with limited control engineering expertise.
- Efficiency: The automated tuning tools within MATLAB Simulink can expedite the tuning process, saving time and effort, especially in cases where a complex system requires quick syntonization.
- Visual Feedback: The ability to visualize and analyze the system's response in real-time is a significant advantage. It allows for a more intuitive understanding of the system's behavior and control performance.
- Robustness: The automated tools in MATLAB Simulink are equipped to handle various system responses, including overshoot, oscillations, and settling times. This ensures robust control parameter tuning.

For an in-depth exploration of our control analysis, including detailed models, simulations, and results, we invite you to review the complete set of resources available in our GitHub repository. These resources provide comprehensive insights into our control strategies and methodologies, offering a deeper understanding of our project's control system.

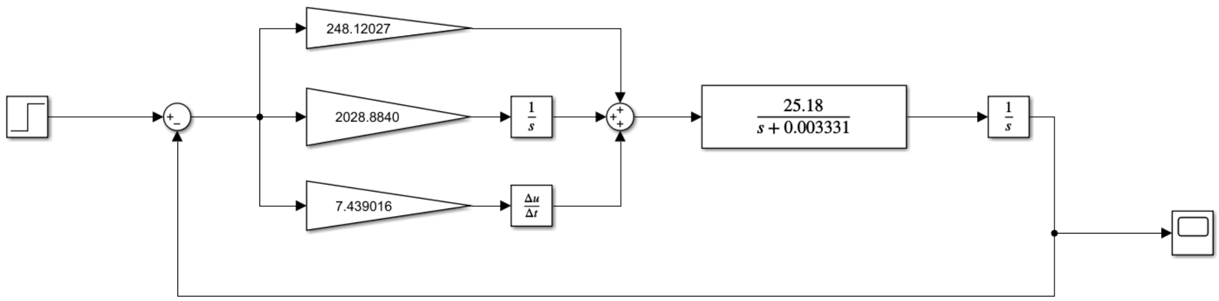
Access to the Code Repository for PID Analysis and Tuning

GitHub Repository - Analysis of Control:

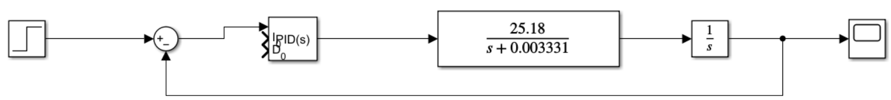
https://github.com/Ineso1/Linear-Axis-Control/tree/main/Analisis%20de%20Control

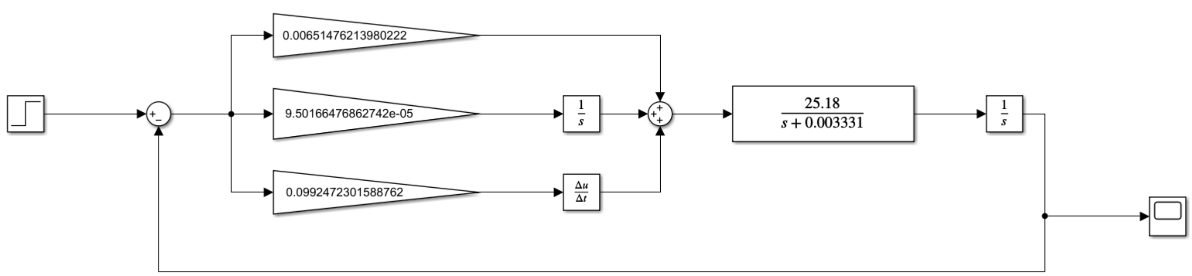
Our analytical approach MATLAB code

```
Unset
#Caracteristicas del control en velocidad angular
Mp = 0.15;
psib = sqrt(((log(Mp))^2 / (pi^2))/(1 - ((log(Mp))^2 / (pi^2))))
ts = 5;
omeganb = 3/(psib*ts)
Q = 2*psib*omeganb;
T = omeganb;
(-Q + sqrt(Q^2 - 4*T))/2;
p1 = 10*Q;
A = Q + p1;
B = T + p1*Q;
C = p1*T;

#Funcio de planta y coeficientes PID

omegan = 0.01;
psi = 3;
2*psi*omegan;

kd = ((A/omegan) - 2*psi)/omegan
kp = B/(omegan^2) - 1
ki = C/(omegan^2)

integ = ki/kp
deriv = kd/kp
```



Out MATLAB toolbox approach





Derivation of The Difference Equation

To derive the difference equation we need to start by considering the PID controller in the time domain. That is,

$$u(t) = K_p e(t) + K_I \int e(t)\,dt + K_d \frac{de(t)}{dt}$$

We then transform it with the Laplace transform. Also, because of the discrete nature of the system. It is necessary to add a holder. Thus the independent transfer functions of both holder and PID controller are as follows.

$$\frac{1-e^{-T_s s}}{s}\ \frac{K_D s^2 + K_P s + K_I}{s}$$

By using some algebra of diagrams. The output of a block is the product of the transfer function and the input signal. Therefore, the product of both holder and PID controller is the resulting equation.

$$\left[\frac{1-e^{T_s s}}{1}\left(\frac{K_D s^2 + K_P s + K_I}{s}\right)\right]$$

At this point it is convenient to use the z transform because we are dealing with a discrete implementation, so it looks like this

$$\mathcal{Z}\left[\frac{1-e^{T_s s}}{1}\left(\frac{K_D s^2 + K_P s + K_I}{s}\right)\right]$$

After the transform and some algebraic manipulation we arrive to the PID expression in z-domain

$$\frac{CV(z)}{E(z)} = K_P + K_I T_s \frac{1}{z-1} + \frac{K_D}{T_s}\frac{z-1}{z}$$

By getting rid of the fractions and separating CV and E on different sides of the equation we arrive to the following expression

$$CV(z) - CV(z)z^{-1} = \left(K_P + \frac{K_D}{T_s}\right)E(z) + \left(-K_P + K_I T_s - 2\frac{K_D}{T_s}\right)E(z)z^{-1} + \frac{K_D}{T_s}E(z)z^{-2}$$

Then, by using known formulas we perform the inverse transform and arrive to the known difference equation form

$$cv(n) = cv(n-1) + \left(K_p + \frac{K_d}{T_s}\right)e(n) + \left(-K_p + K_i T_s - 2\frac{K_d}{T_s}\right)e(n-1) + \frac{K_d}{T_s}e(n-2)$$

Implementation of The Control

Our linear axis control system incorporates a PID (Proportional-Integral-Derivative) controller to precisely regulate the motion of the linear axis. The PID controller plays a crucial role in achieving the desired position while maintaining stability and responsiveness.

```
C/C++
void PositionMotorControl::controlPID(){
 error = targetPosition - (currentPosition * 2 * 3.1416 / 493.9);
 errores[1] = errores[0];
 errores[2] = errores[1];
 errores[0] = error;
 prevPwm = motorSpeed;

 double kp = 248;
 double ki = 2300;
 double kd = 7.44;
 double a = (kp + kd/0.02) * errores[0];
 double b = (-kp + (ki*0.02) - (2*kd/0.02)) * errores[1];
 double c = (kd/0.02) * errores[2];

 int pwmSpeed = prevPwm + a + b + c;
 motorSpeed = pwmSpeed;

 // Comparison between direction and velocity
 if (error <= -1) {
  motorDirection = 1; // Move forward
 }
 else if (error >= 1) {
  motorDirection = -1; // Move backward
 }
 else {
  motorDirection = 0; // Stop
  pwmSpeed = 0;
 }

  // Motor's control
 analogWrite(motorPin1, motorSpeed);
 digitalWrite(motorPin2, motorDirection == -1 ? HIGH : LOW);
 digitalWrite(motorPin3, motorDirection == 1 ? HIGH : LOW);

}
```

PID Parameters:

- kp (Proportional Gain): 248
- ki (Integral Gain): 2300
- kd (Derivative Gain): 7.44

PID Components:

- a: Proportional component
- b: Integral component
- c: Derivative component

This PID controller computes a new PWM speed for the motor based on the current error and the stored error values. It adjusts the motor direction to move towards the target position while considering the error's magnitude.

The PID implementation in our code ensures precise and stable control of the linear axis, enabling it to accurately follow the desired trajectory. The fine-tuned PID parameters are a result of our syntonization process, as described in previous sections. The flexibility to adjust these parameters allows us to tailor the control system to meet specific performance requirements.

Access to the Code Repository for The full Linear Axis Control Code

For a comprehensive view of the project's implementation details, including the codebase, models, and system setup, please explore the complete set of resources available in our GitHub repository:

GitHub Repository - Implementation Section:

https://github.com/Ineso1/Linear-Axis-Control/tree/main/ControlProgram/LinearAxisControl_v2/ControlEjeLineal_v2

Results of the Control

Our PID control approach for the linear axis has yielded notable outcomes, each with distinct characteristics. These results highlight the impact of control syntonization on the performance of our system. We Analyze the achievements of both approaches, emphasizing the advantages and disadvantages of quick and slow response times.

Quick Response Approach (Maximum Impulse: 15%, Steady-State Response: 0.4 seconds)

Slower Response Approach (Maximum Impulse: 1%, Steady-State Response: 2 seconds)

Advantages of a Quick Response:

- Precision: The quick response approach offers precise control, allowing the linear axis to closely follow the desired trajectory with minimal deviation.
- Reduced Settling Time: The rapid steady-state response, achieved within 0.4 seconds, results in a shorter settling time. This is particularly advantageous when the system needs to reach and maintain a specific position swiftly.
- Enhanced Responsiveness: Quick response times make the system highly responsive to changes in the input or disturbances, enhancing its ability to adapt to dynamic conditions.

Disadvantages of a Quick Response:

- Risk of Overshoot: A rapid response may lead to overshooting, where the system briefly exceeds the desired position before stabilizing. This can introduce oscillations or instability.
- Increased Energy Consumption: Achieving a quick response may require higher control efforts, potentially leading to increased energy consumption.

- Potential for Reduced System Longevity: High-speed movements can contribute to wear and tear on mechanical components, which may reduce the system's overall longevity.

Advantages of a Slower Response:

- Stability: The slower response approach prioritizes stability, minimizing the risk of overshoot and oscillations. This can be advantageous in systems where stability is critical.
- Reduced Energy Consumption: Slower responses typically require less energy, making them energy-efficient and cost-effective in the long run.
- Enhanced System Durability: Reduced stress on mechanical components can prolong the system's lifespan, contributing to its long-term reliability.

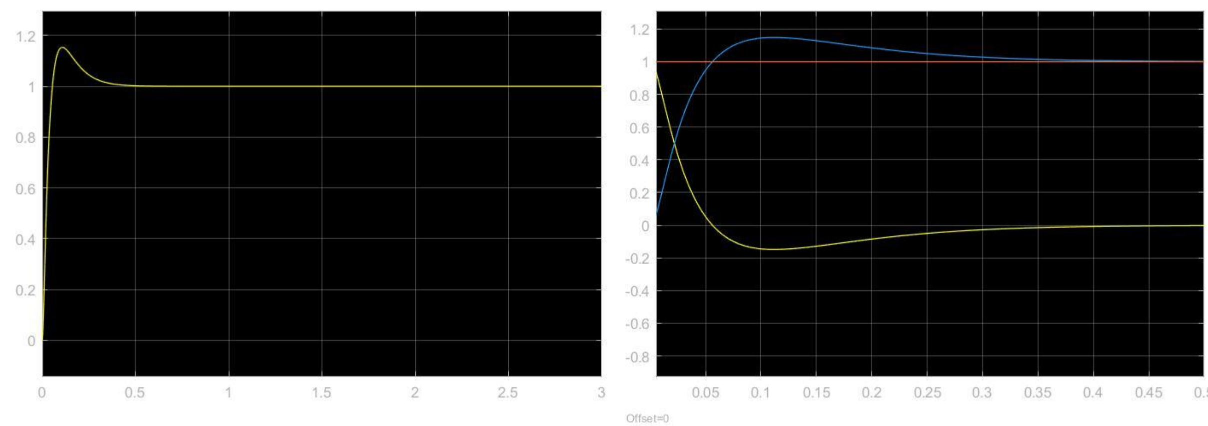
Disadvantages of a Slower Response:

- Reduced Precision: Slower responses may result in less precise control, particularly in applications where rapid, precise movements are required.
- Longer Settling Time: A slower steady-state response with a 2 second settling time may not be suitable for applications that demand rapid positioning or immediate responses to disturbances.
- Limited Responsiveness: Slower responses make the system less adaptable to dynamic changes, potentially affecting its ability to respond swiftly to new conditions.

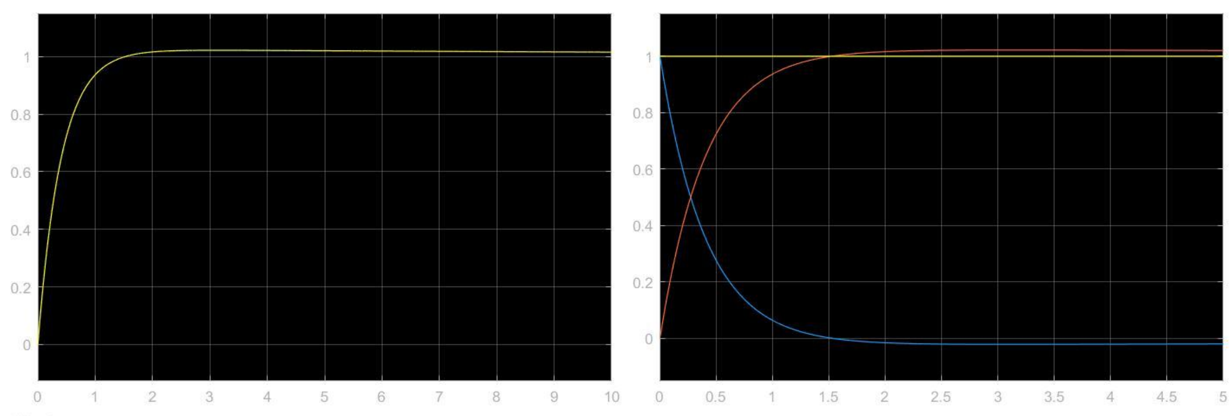
Balancing Act

The choice between a quick response and a slower response is a delicate balancing act. The optimal approach depends on the specific requirements and constraints of the application. In some cases, precision and rapid response are paramount, while in others, stability and energy efficiency take precedence. Therefore, the selection of the control approach should be guided by a thorough understanding of the system's operational context and its performance objectives.

In our project, we have demonstrated the flexibility to adapt our control strategy to different scenarios, showcasing the versatility and effectiveness of our control system design. The ability to choose between quick and slow responses allows us to tailor the system to meet a wide range of practical demands.



Quick Response Control MI = 15%, ST = 0.4s



Slower Response Control MI = 1%, ST = 2s

Conclusion

General

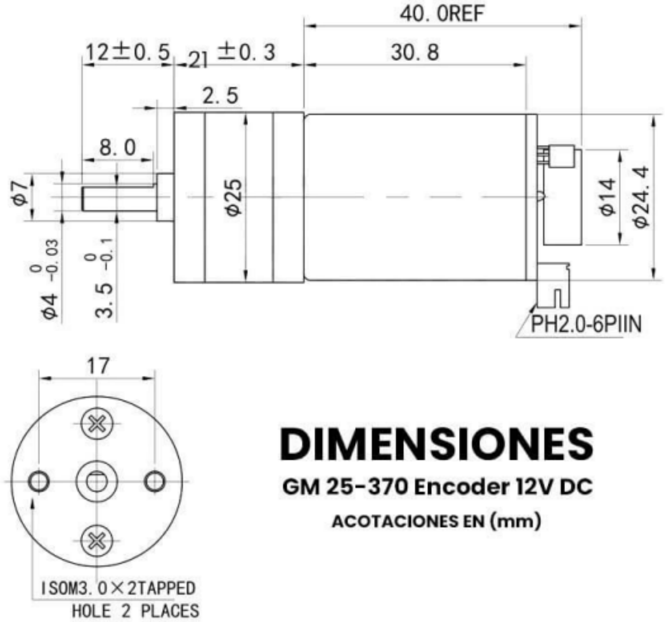
The control, mechanics and electronics of a linear axis operate to achieve precision, speed and reliability in numerous applications ranging from industrial machinery to advanced robotics. Mechanics provide the structural foundation and movement capabilities, driven by electronics ensuring precise energy delivery and timing. The control system receives input from the user or system, converts them into commands for the electronics and mechanics to execute. In conclusion, the perfect integration of these three components is essential. Its collaborative operation ensures efficient and accurate performance of a linear axis, which is vital to maintaining the high standards required in today's technologically advanced environment.

Individual

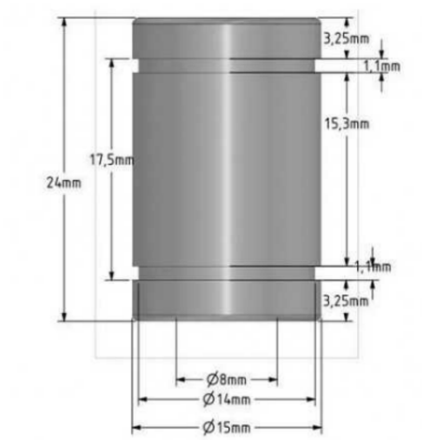
Oliver In my understanding of the control, mechanics, and electronics of a linear axis, they work together to deliver precision, speed, and reliability in a broad spectrum of applications, from industrial machines to cutting-edge robotics. While the mechanics give the structural backbone and movement capabilities, the electronics are responsible for ensuring precise power transmission and synchronization. As for the control system, I see it as the brain that takes in user or system inputs and turns them into actionable instructions for the mechanics and electronics. To sum it up, I believe that the flawless integration of these three elements is critical. Their collective function ensures the efficient and precise operation of a linear axis, which I consider essential in the high-standard, tech-driven environments of today.

Inés I've acquired a profound understanding of control theory and its practical applications in digital systems, particularly in the discrete-time domain. The journey involved bridging the gap between theory and real world implementation, a valuable skill in the realm of engineering. From PID controllers to plant identification and syntonization, I've learned how to harness mathematical principles to achieve precise control. I now appreciate the beauty of control theory in creating efficient, stable, and responsive systems. This experience has provided me with a strong foundation in translating complex mathematical concepts into code, enabling me to bring control theory to life in my projects. As I move forward, I feel well-equipped to tackle diverse engineering challenges with a solid grasp of mathematics, the theory, and the practical application of control systems in digital systems and discrete-time environments.

Paul I have learned several things over the course of 10 weeks. Most importantly, I have realized all the things that can be done with control theory. I believe that a linear axis is just one of the many applications of control. I am excited to see what other things can be done, especially, in the realm of robotics which is an important and emerging field. As far as I am concerned there are several subfields in control theory given the nature of several dynamical systems. There will be times when it is not possible to linearize a system, so we will be forced to look for generalized ways of modeling a system. Nevertheless, I presume this was a fairly good introduction to control theory and some of its applications. I look forward to learning more on the subject, and luckily enough with a more theoretical approach, as I strongly believe it is essential to comprehend the area.



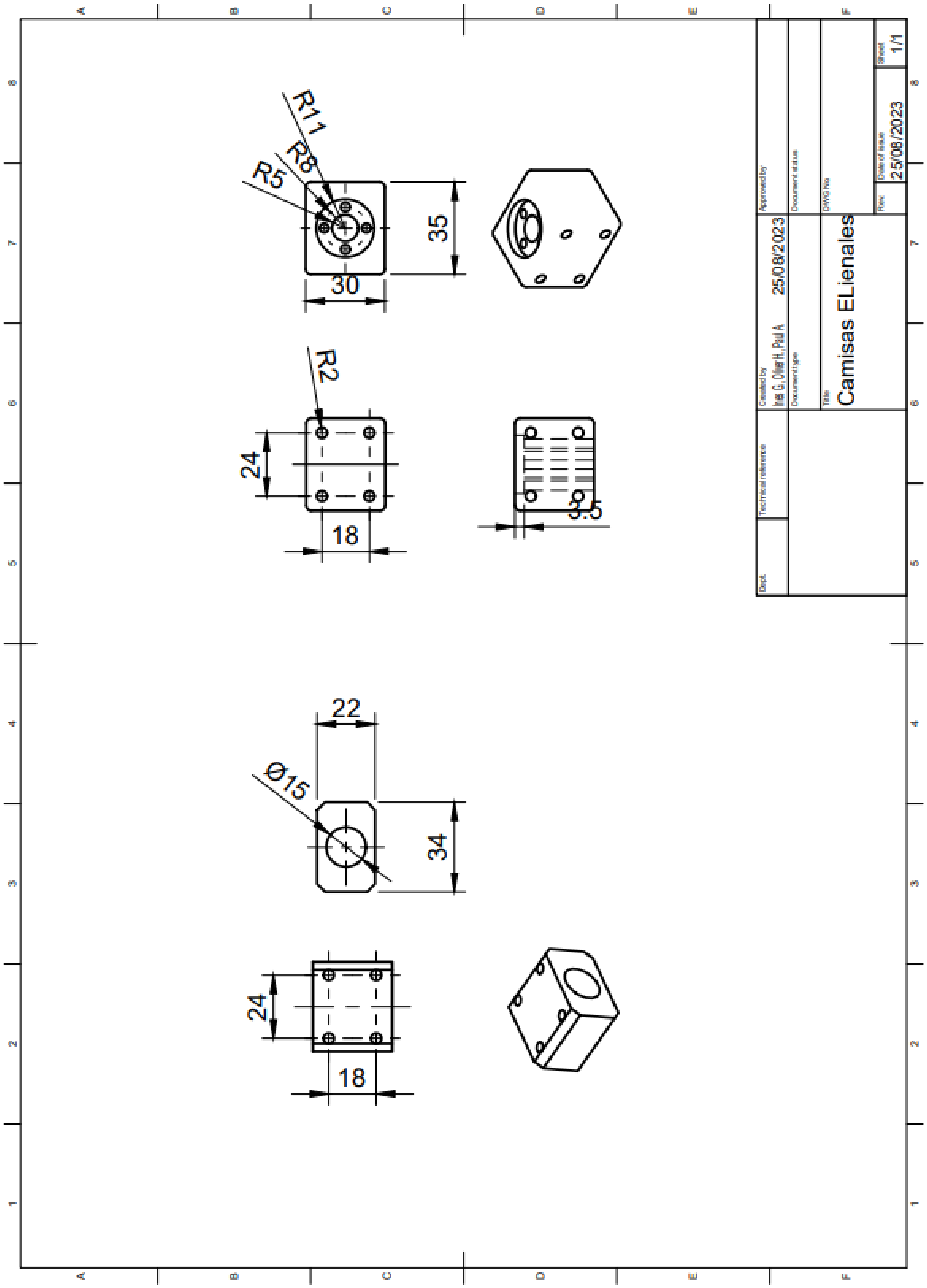
https://uelectronics.com/producto/gm-25-370-motor-con-encoder-12v-dc-140rpm-330rpm/



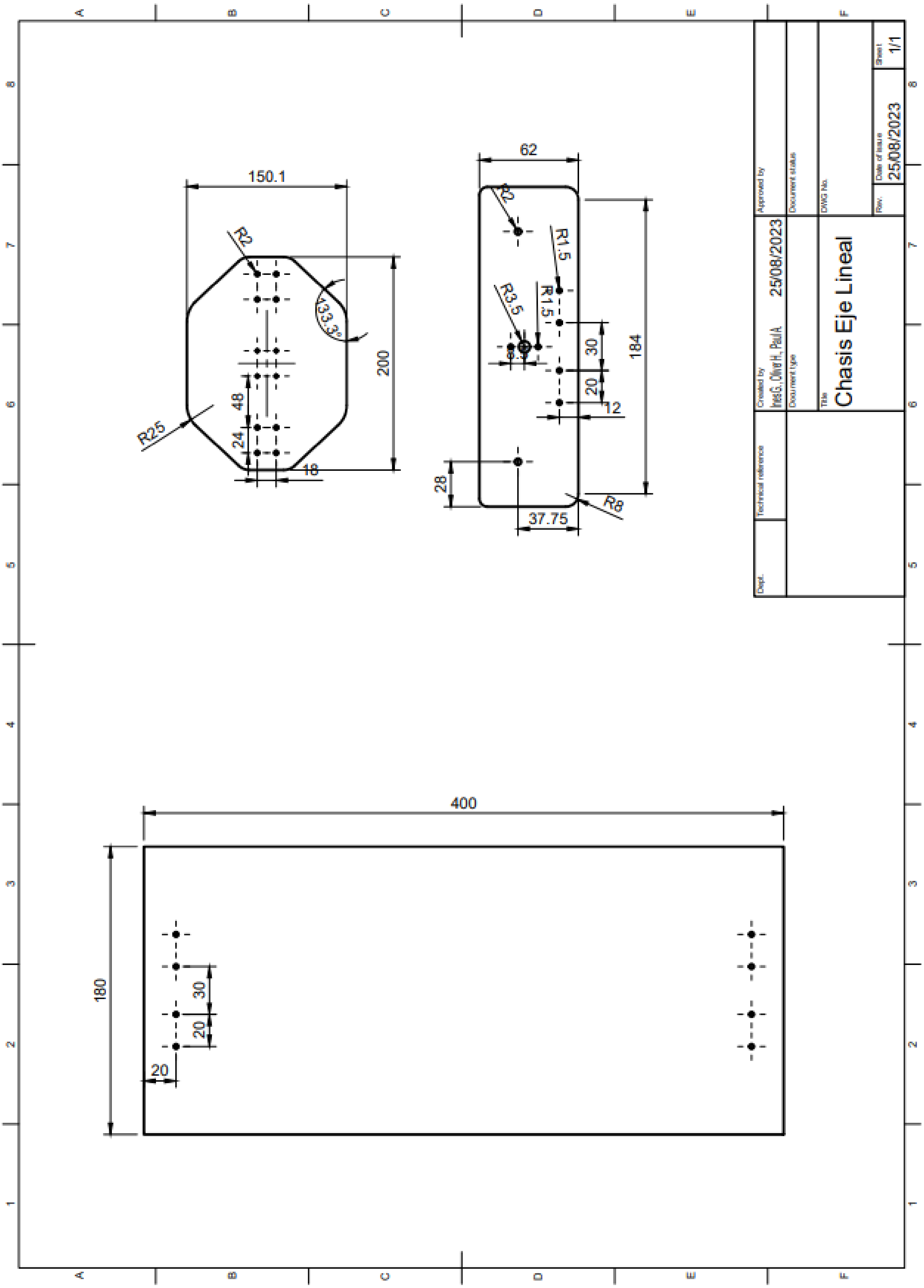
https://uelectronics.com/producto/balero-rodamiento-lineal-lm8uu-de-8mm-3d-cnc/

Etiquetas: Acoplador Nema, cople, Cople D18L25, cople flexible, Cople Flexible 5x5mm, Cople Flexible 5x8mm, Cople Flexible 6.35x8mm, Cople Flexible 8x10mm, Cople Flexible 8x8mm, Cople Flexible D18L25, D18L25, Piezas y Accesorios 3D CNC
https://uelectronics.com/producto/cople-flexible-d18l25-nema-acoplador/

Varilla Lisa 8mm 60cm para Impresora 3D/CNC
https://uelectronics.com/producto/varilla-lisa-8mm-para-impresora-3d-cnc/



R11
R8
R5
35
30
R2
24
18
3.5
22
Ø15
34
24
18
Technical reference
Created by
Ines G, Oliver H., Paul A.
25/08/2023
Approved by
Document type
Document status
Title
Camisas ELienales
DWG No.
Rev.
Date of issue
25/08/2023
Sheet
1/1
Dept.


150.1
R2
133.3°
R25
48
24
18
200
62
R2
R1.5
R3.5
R1.5
30
20
12
184
28
37.75
R8
400
180
30
20
20
Approved by
Document status
DWG No.
Created by
25/08/2023
Document type
Title
Chasis Eje Lineal
Rev.
Date of issue
25/08/2023
Sheet
1/1
Technical reference
Dept.

References

Britannica. (2023, May 28). Computer Numerical Control. https://www.britannica.com/technology/numerical-control

Electricity - Magnetism. (n.d.). H-Bridges. https://www.electricity-magnetism.org/h-bridges/

Encoder Products Company. (n.d.). What is an encoder?. https://www.encoder.com/article-what-is-an-encoder

Figliola, R., & Beasley, D. (2011). Theory and Design for Mechanical Measurements. John Wiley & Sons, Inc.

Franz, K. (2023, April 18). What Is an H-Bridge?. https://digilent.com/blog/what-is-an-h-bridge/

Slid Siim. (2022, January 24). Lead Screws Explained. https://fractory.com/lead-screws/

Mega, A. (s/f). Arduino® MEGA 2560 Rev3. Arduino.cc. Recuperado el 10 de septiembre de 2023, de https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf

Mega, A. (s/f). Arduino® MEGA 2560 Rev3. Arduino.cc. Recuperado el 10 de septiembre de 2023, de https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf

Mega, A. (s/f). Arduino® MEGA 2560 Rev3. Arduino.cc. Recuperado el 10 de septiembre de 2023, de https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf

Appendix

Code

https://github.com/Ineso1/Linear-Axis-Control/tree/main