ARM WRESTLING TRAINING ROBOT

ME 429 – Mechanical and Thermal Design

Final Report

Samed BAYER

Elif GÜNDOĞAN

Nilay Meltem YILDIRIM

ADVISOR: Çetin YILMAZ

ABSTRACT

Arm Wrestling Training Robot is a project that aims to offer players the opportunity to improve themselves while having fun and competing in different modes. It measures the torque applied by the opponent and gives feedback at the end of the game. In order to achieve these goals, a robot arm that can move in 2 degrees of freedom using 2 DC servo motors is designed. In this way, it is aimed that the robot moves in a similar manner to real arm wrestling while keeping the system as simple as possible. To obtain low speed and high torque, the motors are supported with the gear reducer and chain-sprocket system. While the technical drawing of the system is made with the help of Solidworks, necessary strength and durability analyzes are made with Adams software. Bearing and chain selections are made according to the dynamic loads. A simulation-based algorithm is developed to use the robot in different modes with random numbers and this algorithm is encoded with the Phyton software language. Position control of the motors are planned to be handled with a PID controller. The spesifics of this control mechanism will be determined after the installation of the whole system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Arm wrestling, also known as "arm fighting", is a type of sport played with two people and also a popular game played around the world. It is known that the first organized arm wrestling tournaments started in the 1950s, but arm wrestling history is estimated to date back to ancient Egyptian history, based on the many murals in Egyptian caves. On the other hand, it is known that the modern arm wrestling sport is based on a Native American game [1].

In addition to playing professionally, arm wrestling has turned into a game that people often play for entertainment in their daily lives, starting from childhood. Two people place their elbows on the table, wrap each other's hand and try to bring the other person's hand to the table. Arm strength is obviously important to win, but there are other factors that make the game more interesting. It is possible to slightly move your elbow, pull the opponent's hand to yourself and try to bend their wrist to maximize your impact. According to professional players, even the handgrip can affect how well you apply your strength. Stamina, both physical and psychological, is also an important factor.

Figure 1. Arm wrestling robot systems [2, 3].

There are many arm-wrestling devices in arcades that work on a simple principle. They apply a certain torque, and if the opponent can overcome that torque, they can beat the machine, which makes them deterministic. These machines have one degree of freedom (DOF) and only move in the win-lose axis. An example of such a system can be seen in Figure 1. The study by Kang et al. aims to improve this system by incorporating a force feedback control and trying to make the game competitive, where the robot's response changes according to the applied force [4]. However, the human arm has 3 joints disregarding the hand and 6 DOF, which determine the position of the wrist and the orientation of the palm [5]. This shows that a 1 DOF system is a major reduction of the human arm. Examples of more complex arm-wrestling robots can be seen in the literature. Gao et al. develop a 2 DOF arm wrestling robot with integrated joint torque sensors [6]. Yamada and Watanabe develop a 5 DOF arm wrestling robot system that changes its expressions based on the game. As seen in Figure 1, their system has a whole arm instead of only the upper arm as well as a face, which makes it more human-like. It is important to note that with increasing degrees of freedom, the system becomes more complex and the control effort is increased.

Professional arm wrestlers work out to strengthen their arms and upper body. They use standard gym equipment and some specialized tools. They also train with other arm wrestlers which is better because they can get feedback. Even then, they do not have a quantitative measure of how much they have improved over time. The purpose of this project is to design a robotic arm that measures the torque applied by the user and varies its actuation torque accordingly, offering a fair arm wrestling match. The data acquired during the match, torque, its duration, and position, is given as feedback to the user. The safety of the user is very important because arm wrestling can cause serious injuries.

OVERVIEW OF POSSIBLE SOLUTIONS

In this project, providing motion is one of the basic requirements and this process is carried out by means of actuators. Actuators are mechanical or electro-mechanical devices that provide controlled movements or positioning and can be operated electrically, manually or with various fluids [7]. There are two basic movements, linear and rotary. Linear actuators typically have a push and pull capability while converting energy into straight-line motions for positioning applications [7]. On the other hand, a rotary actuator is an actuator type that produces a rotational motion or torque. Typical use to exemplify this movement is to control various valves such as ball valves or butterfly valves.

Figure 2. Pneumatic linear (a), hydrolic linear (b) and electric rotary actuators [9, 32, 35].

While working on this project, 3 types of actuators were considered: electric, pneumatic and hydraulic actuators. Pneumatic actuators use compressed air and are capable of handling high torque loads, but since the power supply is a compressible gas, they cannot give successful results against adhesion/slip reaction, time constraints and inconsistencies in efficiency [8]. In addition, extra equipment is required for their work. This equipment is the compressor that will produce the compressed air that keeps the actuator running. Compressors make a lot of noise when running, which can be quite problematic for some uses. On the other hand, hydraulic actuators are tools that use hydraulic power to provide mechanical work [8]. Mechanical motion output is obtained in the form of linear, rotary, or oscillating motion. A hydraulic actuator can exert enormous force as liquids are incompressible. Because of that, these types of actuators are used in construction machines for very heavy loads [7]. The preferred fluid in their use is oil and therefore they don’t provide a very clean operation. Our latest type of actuator, the electric actuator, is a mechanical device used to convert electricity into kinetic energy with a single linear or rotary motion. Electric actuators provide the highest precision-controlled positioning [9]. They also offer full motion control and are scalable for any purpose or power requirement. Low noise, cleanliness and accessibility are other arresting features that make this actuator type important and useful. Besides, it has a low response time [9]. All of these reasons make electric actuators the most commonly used type of actuator in robotic applications.

Before choosing an actuator and designing the arm-wrestling robot, it is necessary to find how much torque an average arm wrestler generates. A simple experiment is conducted to obtain this information. A scale is placed on the table. A person places their elbow on the table and applies force in a position similar to arm wrestling, as seen in Figure 3. By multiplying this force with the length of that person’s arm, a torque value is obtained. The torque values are given for three people in Table 1. The results are compared with Kang et al., who performed a similar experiment where the scale was vertical. As seen from Table 1, the average of the current experiment is slightly higher with 45.6 Nm.

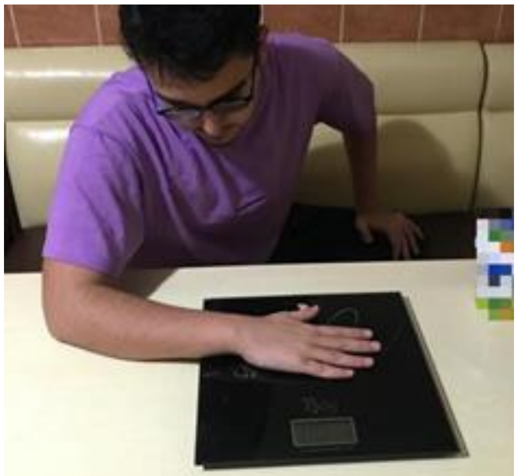


Figure 3. Experimental setup.

Table 1. Experimental results.

Person 1	29 cm x 19.9 kg = 56.6 Nm
Person 2	27.5 cm x 16 kg = 43.1 Nm
Person 3	27 cm x 14 kg = 37.1 Nm
Average	45.6 Nm
Average (Kang et al.)	32.7 Nm

A design torque of 50 Nm is chosen for this project. When a safety factor of about 1.5 is considered, which is common in robotic applications [10], the torque we need to achieve goes up to 75 Nm. When we examine the options we have in detail, we see that the selection of electric actuators is the most reasonable solution for us because they are very accessible and clean to use. Possible electric rotary actuators are AC and DC motors. Because DC motors have higher starting torque and offer a wide range above and below the rated speed, we continued our project by choosing a DC motor. However, 75 Nm is high for a lot of commercial DC motors. Because of that, several options for torque multiplication are considered.

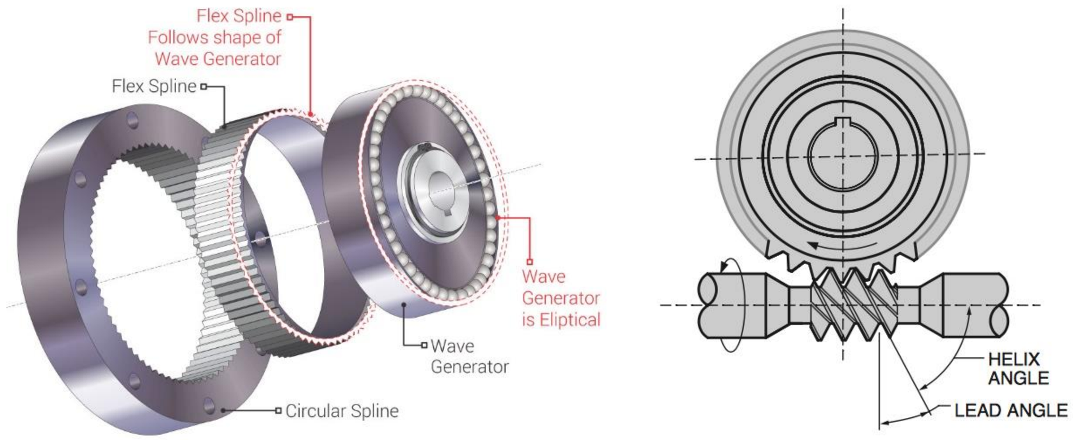


Figure 4. Strain wave gear reducer (a) and worm gear reducer (b) [11, 12].

The most common method for torque multiplication in electric motors is using a gearbox. It reduces the speed and increases the torque. Some different types of gearboxes with high reduction ratios can be seen in Figure 4. Harmonic drive or strain wave gear consists of an elliptical wave generator, a flex spline that takes the shape of the wave generator and a circular spline. It has high reliability, however it is quite expensive and not back-drivable [13]. Back driving is being able to work in reverse. In other words, if the gearbox can receive input from its output shaft, it is back-drivable. This is an important distinction to make in relation to the arm wrestling robot as it determines whether the opponent can change the position of the arm by applying torque. Worm gear reducer is another self-locking reducer. It changes the direction of the rotation by 90 degrees because of the worm gear.

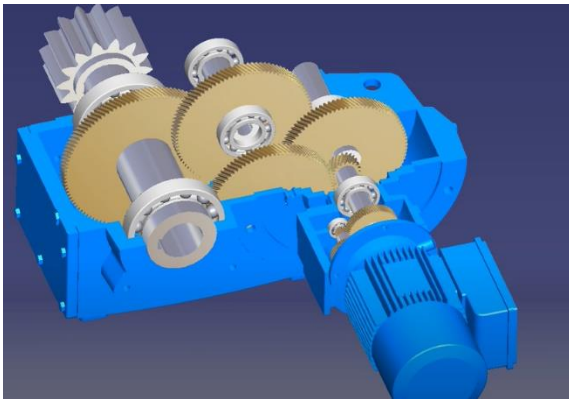


Figure 5. Spur gearbox reducer [13].

Both back-drivable and self-locking mechanisms can be used, however the design of the algorithm needs to be changed accordingly. Initially, a back-drivable system was considered, because it is intuitively more like arm wrestling. Spur gearbox reducers, as seen in Figure 5, are back-drivable up to high speed reduction ratios. They are compact and commonly used in motion-control applications.

In general, when we searched for commercial motors with gearboxes, the ones that can provide 50-100 Nm torque were either too expensive or too bulky. Big and heavy motors would be hard to place on the arm without hindering its movement and/or causing extra strain on the arm. For this, we turned to solutions such as adding a second gear reducer, increasing the torque with a chain or belt system. Considering an appropriate motor we found that could provide 12 Nm, we need a torque multiplication of approximately 1:6. First, we focused on the use of pulley systems. A pulley mechanism with 2 mm steel wire was considered. But the diameter of pulleys in the system became too large to achieve the reduction ratio because the smallest pulley radius is 4 cm for 2 mm wire, which when multiplied by 6 causes the large pulley to take up a lot of space. Then we designed two-belt systems like shown in the figure given below:

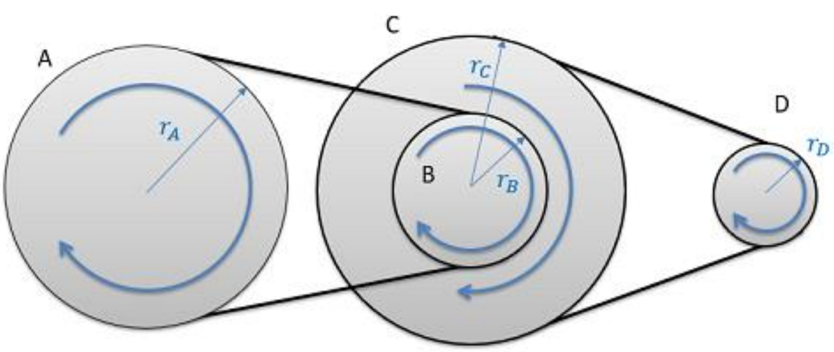


Figure 6. Belt driven system example [14].

Even if we put two systems in a row like in the example figure, we could not prevent it from taking up too much space. In addition, two belts would need two sets of bearings which would be costly. That's why we decided to look for alternative solutions. Another way is to add a 2nd gearbox to the system. But after getting a ready-made motor-gearbox system, fixing and correctly aligning a second gearbox is harder. Therefore, we tried to find an alternative solution, although we did not completely reject this one.

Figure 7. Chain / sprocket system [34].

The sprocket system is the last alternative method. Unlike belt drives, chain technology has no slippage. They are also generally suitable for small center distances. A torque increase of up to 1:8 can be achieved and they operate with an efficiency of up to 98% [15]. For chains, as for bearings, it is necessary to consult manufacturers' catalogs. Chains and sprockets are manufactured according to standards and appropriate ones that meet the desing criteria needs to be chosen from the catalog. Sprocket diameter can be lowered down to 2.5 cm, which solves the size problem encountered with belts. In addition, the chain system is more compact and easy to install compared to a belt drive.

The purpose of our robotic arm is to create scenarios according to the torque applied by the other person and to give feedback to that person so that they can learn their development. For this, the applied torque must be measured. We focused on 3 different torque measurement methods. The first is to measure torque using the current drawn by the motor.

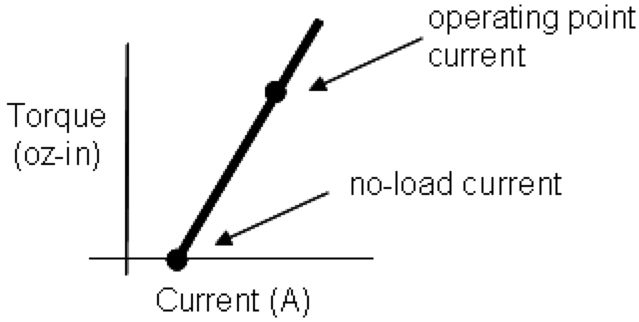


Figure 8. Torque-current curve for a DC motor [16].

The figure shows us the linear relationship between current and torque for a DC motor. What draws attention in this graph is that the torque-current curve for a DC motor is that the no-load (stall) current is greater than zero. Due to the fact that some current is needed to overcome the internal friction, which exists inside the motor itself.

Another method is to measure the torque with “Strain Gage Torquemeter”.

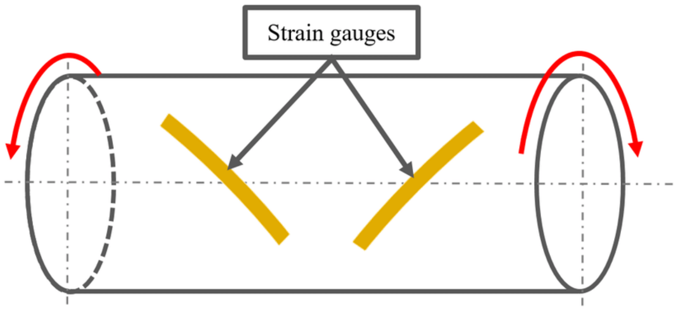


Figure 9. Strain gage torquemeter [17].

A strain gauge works by converting strain into an electrical signal [18]. The sensor is mounted on a rotating shaft that deforms when a torque is applied. Generally, four sensors are placed on the rotating shaft in the form of a Wheatstone bridge circuit [18]. There is a great risk of positional errors in this method. In addition, it is difficult to establish a connection between material and gage.

Because of the addition of sprockets, an alternative method to measure the torque is by measuring the chain tension. One method to do that is by using a magnetic pick-up. A magnetic pick-up is an inductive sensor that consists of a coil wrapped around one or multiple permanent magnets, as seen in Figure 10. It is frequenly used in electric guitars. The permanent magnet in the pick-up creates a dipole in the string, because strings are made from ferromagnetic materials. When the string vibrates, its magnetic field fluctuates, which causes an electric signal in the coil [19].

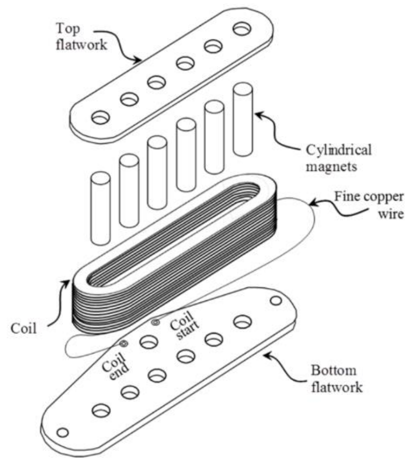


Figure 10. Parts of a magnetic pick-up [19].

A similar magnetic sensor is used in bikes to measure the chain tension from the vibration as the chain goes over the sensor. By combining it with a speed sensor, it is possible to calculate the power generated by the biker. According to the sensor manual, the tension is reasonably accurate up to a 30mm gap between the sensor and the chain [20].

DETAILED DESIGN AND ANALYSIS

The arm wrestling robot consists of a shoulder and an arm, as labeled in the figure below. The names are chosen to be convenient and does not imply that the movements are identical with the human arm. The actuation is carried out with DC servo motors and a sprocket system that multiplies the torque. It has two degrees of freedom. The first axis is the win-lose axis and controlled with the motor positioned on the shoulder, labeled Motor 1. Motor 1 is fixed to the shoulder with a bracket and M3 screws. The arm is connected to the shoulder with a pin joint, which is actualised with bearings 3 and 4. The second axis is controlled by Motor 2, which is positioned on the table. The shoulder is fixed to the table with two metal legs. Bearing 1 and 2 are fixed inside the legs. The shoulder is able to rotate freely with respect to the table by a shaft that goes through the bearings.

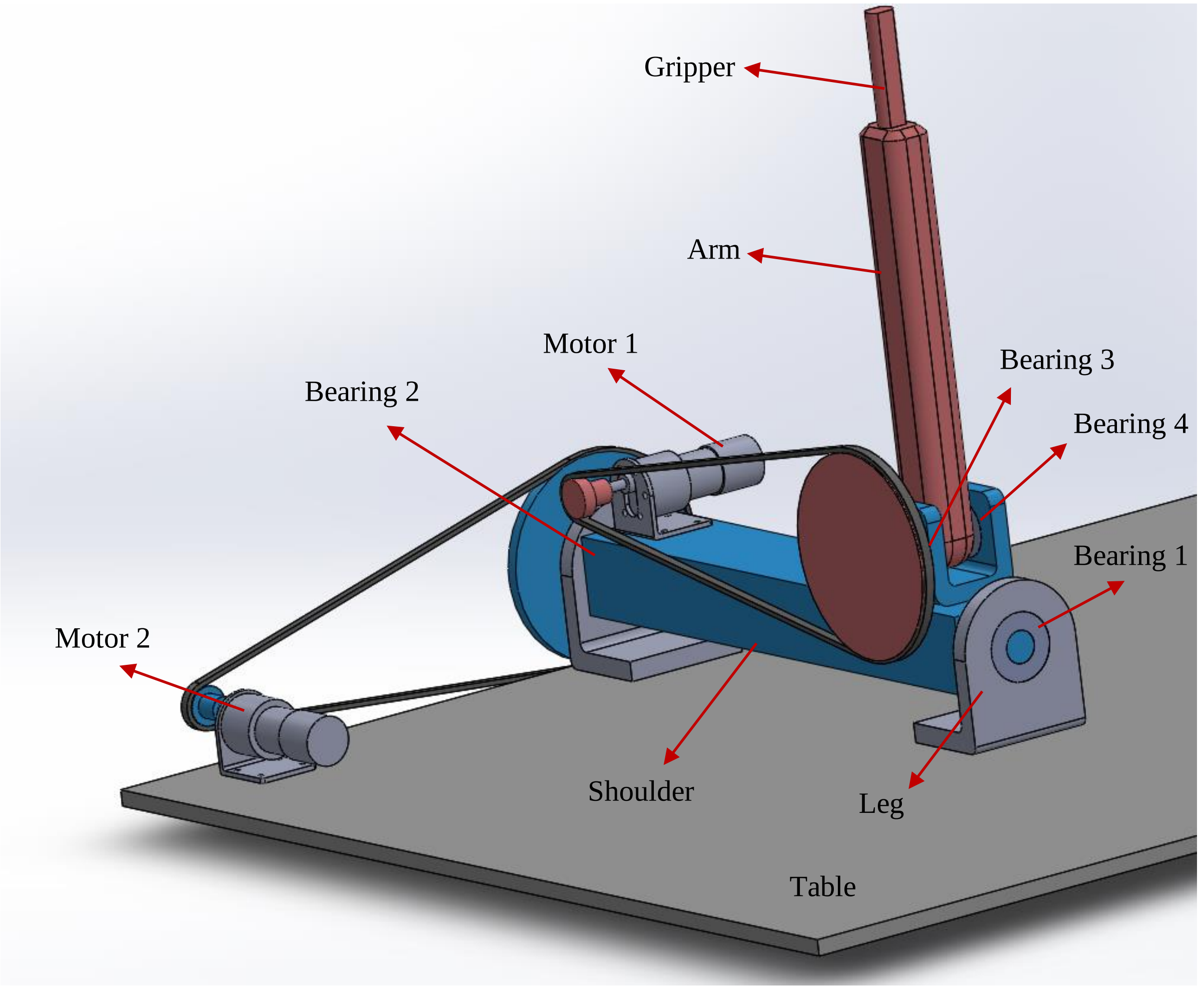


Figure 11. Overall setup of the arm wrestling robot. Important parts are pointed out and labeled.

Position convention for the motor is from the robot perspective. When looking robot side, the starting point corresponds to 90°, with 0° to the CW direction. The game is played between 30° and 120° for safety reasons. This means that the opponent needs to move the arm 60° CCW from the starting point of to win the match. Unless spesified otherwise, all position information is given according to this convention.

Detailed analysis is carried out to choose appropriate bearings, sprockets and chain. Stress calculations are made at critical points to determine certain design parameters.

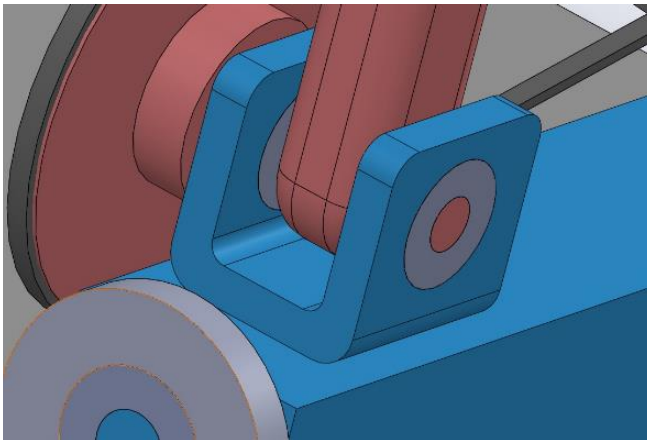


Figure 12. Close-up of the joint holding the arm and the sprocket.

First critical point is the joint, spesifically the pin, holding the arm. In order to make the analysis in detail, it is necessary to draw the free body diagram of this point. The highest load is applied to the joint when the arm is stationary and maximum torque (design torque) is applied by the opponent in both degrees of freedom. The FBD of the pin is shown schematically in the figure below for this case.

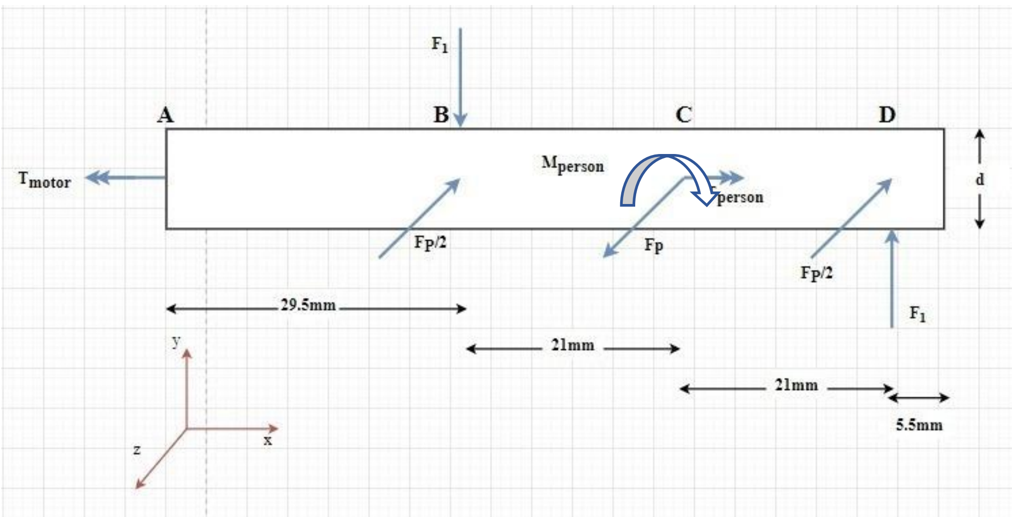


Figure 13. FBD of the pin for the maximum load case.

In the figure F_p, M_p and T_p show the force and moments applied by the person. The bearings apply reactionary forces and a force couple, shown with F, to balance the applied moment. Torque applied by the motor to hold the arm in place is shown with T_{motor}. As seen in Figure 11, there are forces in both y and z axes, so shear and moment diagrams should be drawn for these two different axes and critical points should be calculated.

- *Fp = 248.72 N* *F = 1714.28 N* *Tp = Tmotor = 72 Nm*

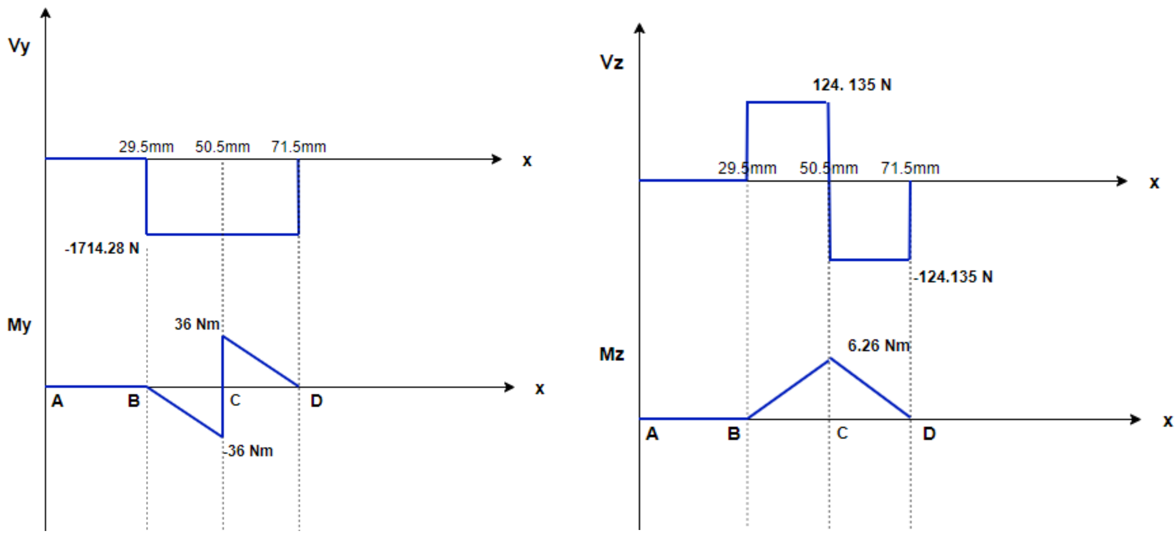


Figure 14. Shear force and moment diagrams for the pin.

By looking at the graphs it can be seen that the maximum shear force, moment and torsion are at point C. It is the most critical point in the pin.

- $\sigma = \frac{My}{I} = \frac{32M}{\pi d^3}$ Normal stress (1)
- $\tau = \frac{4V}{3A} = \frac{16V}{3\pi d^2}$ Transverse shear stress (2)
- $\tau = \frac{Ty}{J} = \frac{16T}{\pi d^3}$ Shear stress caused by torsion (3)
- $\sigma_e = \sqrt{\sigma^2 + 3\tau^2}$ Equivalent stress (4)

Diameter of the pin, d, is found iteratively as 15 mm. AISI 1040 steel is chosen for the pin, which has 413.7 MPa yield strength (S_y). When we put the values in place and calculate:

$M = \sqrt{36^2 + 6.26^2}$ T = 72 Nm V – assumed negligible

Calculations are made again using Equations (1), (3) and (4).

$$\sigma = 110.27\ MPa$$

$$\tau = 108.65\ MPa$$

$$\sigma_e = 218.11\ MPa$$

➢ $SF = \frac{S_y}{\sigma_e} = \frac{413.7}{218.11} = \mathbf{1.89}$

According to the calculated safety factor, it can be seen that the material does not yield at point C. The designed pin can withstand the applied forces. Under normal conditions, there is force distributed along B-D, but we have operated with concentrated forces instead. In this way, we have made a more conservative calculation and designed a system that does not fail even under these conditions.

Motor Specifications and Control Mechanism

Two 12V, 10 rpm servo motors are used in the project to actuate the two DOF. The chosen motor is equipped with a standard in-line gearbox with a gear ratio of 1:1620 and can provide a stall torque of 12 Nm in this configuration. Its high ratio means that the gearbox is not back-drivable. When used together with the chain-sprocket system which has a 1:6 reduction ratio, the motor moves the arm with a maximum speed of 1.67 rpm or 10 deg/sec. This is one of the safety precautions against injuring opponents, as many arm wrestling injuries occur because of sudden and fast movements. The maximum torque that can be applied to the arm is 72 Nm.

The motor is packaged with a quadrature encoder, which has 164 counts per revolution (CPR) on the base motor. This means 265,680 CPR on the output shaft, which corresponds to a resolution of about 0.002 degrees. This resolution is more than enough for precise position and speed control of the motor. The terminals of the motor are given in Table 2.

Table 2. Motor terminals.

Terminal	Color
Ground (GND)	Black
VCC	Brown
Encoder A	Red
Encoder B	Orange
Motor +	Yellow
Motor -	Green

The motors have a no-load current of 0.8A and can draw up to 7.5A current under load. A dual 15A H-bridge motor driver is chosen to electrically isolate the motor from the microcontroller and to control the direction and speed of the motor. The driver has two sets of terminals. The first set is to connect the motors and the supply voltage. It can work with a power source between 5-35V. The second set is for control operations and has 6 terminals: +5V, DIR1, PWM1, DIR2, PWM2 and GND. DIR terminal is connected to the microcontroller to control the direction of the motor, where 1 corresponds the CW motion and 0 corresponds to CCW motion. PWM terminals are used to determine the speed of the motor with pulse width modulation.

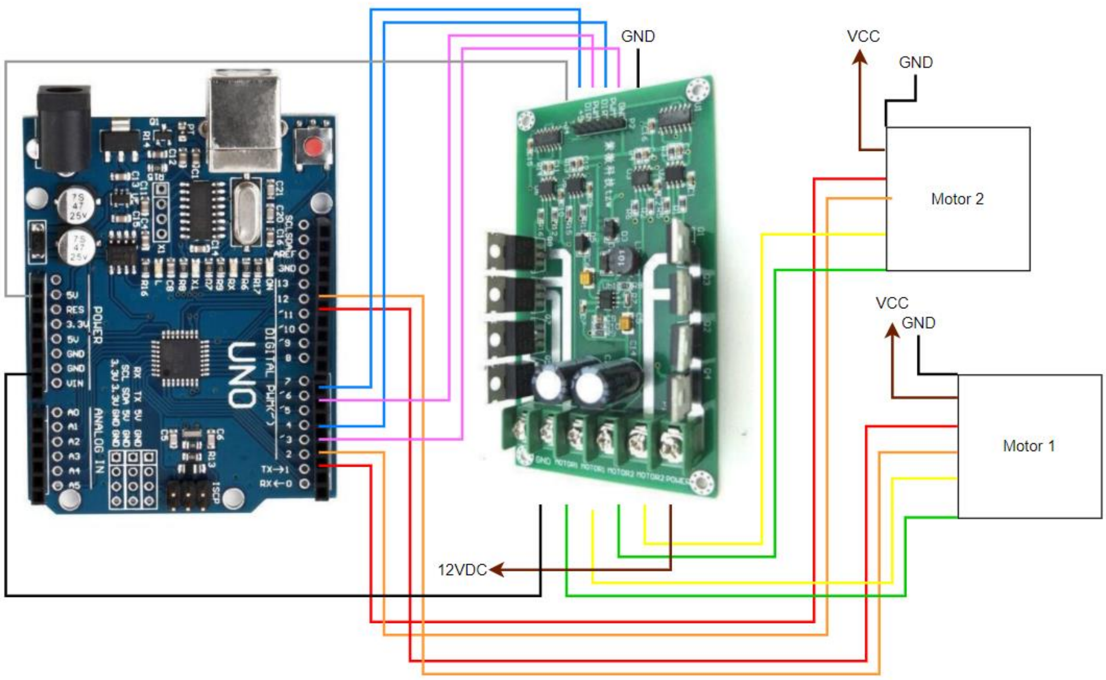


Figure 15. Arduino connections of the motors and the driver.

The Arduino connections for the motors and the driver are shown in Figure 10. Motor terminals are shown with the corresponding colors from Table 2. The DIR terminals are shown with blue wiring. The PWM terminals are shown with pink wiring and connected to Arduino pins with PWM capabilities, which is shown with a tilda symbol on the Arduino. Voltage necessary for the logic operations of the driver is supplied from Arduino’s 5V output.

One of the most critical stages in the analysis and implementation of mechanical systems is the applied control mechanism. The main goal is to develop a model that manages the application of system inputs to bring the system to the desired state [21]. As explained before, the robotic arm is not back-drivable and the algorithm gives incremental position commands in the direction to win or lose, based on the applied torque. Speed control of the servo motor is achieved with closed loop (CL) PID control. It is schematically shown below, where the position command (θ_set) is given from the microcontroller.

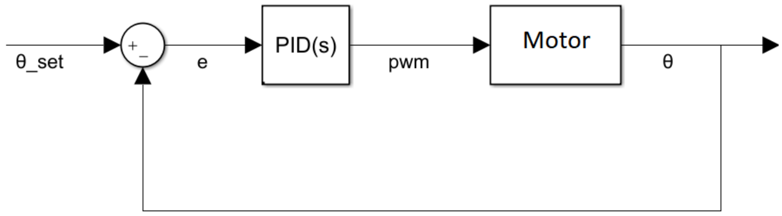


Figure 16. Schematic representation of the closed loop PID control system.

The figure below shows the graphical representation of the control parameters such as overshoot, steady state error and settling time:

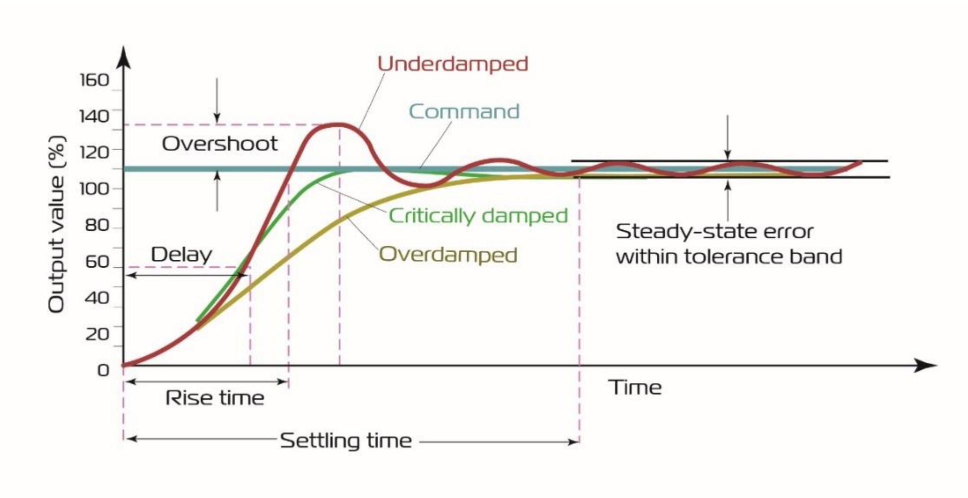


Figure 17. Graphical representation of feedback control parameters [33].

The control signal (u) of a PID controller is calculated with the following formula [21]:

$$u = K_p e + K_i \int_0^t e\,dt + K_d \frac{d}{dt} e \tag{5}$$

As seen in the above equation, the output of the PID controller is calculated from the error in the time domain. Kp, Ki, and Kd refer to the proportional, integral, and differentials terms of the controller, respectively. In order to understand control better, it is necessary to understand the effects of these terms on control parameters. The proportional gain affects the rise time, for example as Kp increases, the rise time decreases. But this also leads to an increase in overshoot. Besides, it significantly reduces steady-state error, although it does not eliminate it completely.

Overshoot increased due to Ki can be reduced with Kd (derivative term), thus adding damping to the system. But Kd does not affect steady-state error. The integral term Ki helps to reduce steady-state error, but the downside of this term is that it can cause the system to become oscillatory [21]. All of these are summarized briefly in the table below:

Table 3. Control parameters summary [21].

CL Response	Rise time	Overshoot	Settling time	SS error
Kp	Decrease	Increase	Small change	Decrease
Ki	Decrease	Increase	Increase	Decrease
Kd	Small change	Decrease	Decrease	No change

Normally, by obtaining the transfer function of a system, control simulations can be performed on Matlab Simulink for this system. In other words, by manipulating the Kp, Ki and Kd values, it is observed how the current system will behave, and thus most suitable values are found. The transfer function of a DC motor in the Laplace domain is given as follows:

$$P(s) = \frac{\theta(s)}{V(s)} = \frac{K}{(Js+b)(Ls+R)+K^2} \quad \left[\frac{rad/sec}{V}\right] \qquad (6)$$

J = moment of inertia of the rotor

b = motor viscous friction constant

K = electromotive force constant (motor torque constant)

R = electric resistance

L = electric inductance

As can be seen, some motor parameters are necessary to find this transfer function. Unfortunately, we do not know these values for our motor and therefore cannot perform a simulation on Matlab.

Another method to find suitable Kp, Kd and Ki values is by experimentation. By building the motor setup and making necessary encoder connections, it is possible to measure the real-time position and speed of the motor. With the the code shown in the video [22],we can manually assign Ki, Kp, and Kd values and run the system. In this way, how the system responds according to these values is observed graphically. These values are calibrated until the desired response is achieved, and thus the PID controller is obtained [22].

The interface of a similar setup to calibrate a PID controller for a speed control application is shown in Figure 13. The real-time output speed is be shown in the graph while the motor is running and it can be observed how the system behaves according to the gain terms.

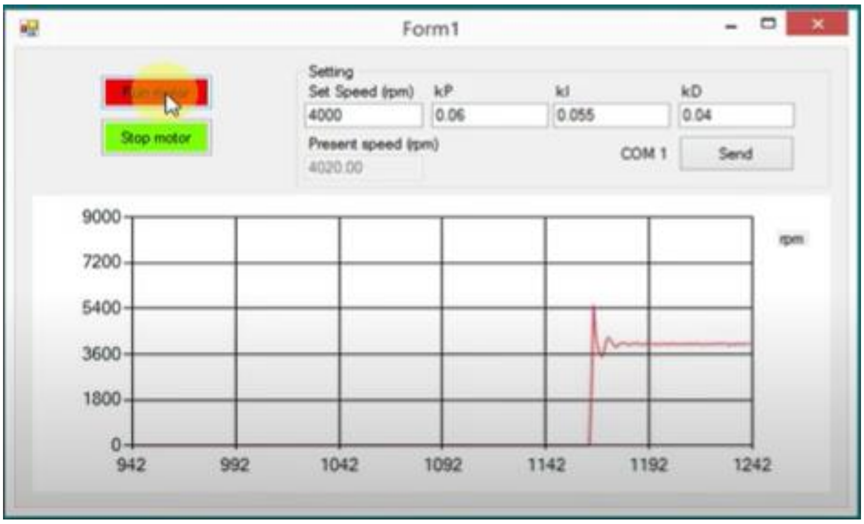


Figure 18. The obtained data by applying PID [22].

Torque Sensors

There are two different torque measurement systems in the arm wrestling robot. The first one is a magnetic sensor that consists of an electromagnet and a Hall effect sensor. Hall effect sensors give a voltage output proportional to the strength of the magnetic field [23]. The electromagnet consists of a coil of wire wrapped around an iron core. It produces a magnetic field proportional to the current, number of windings, and the length of the solenoid [24].

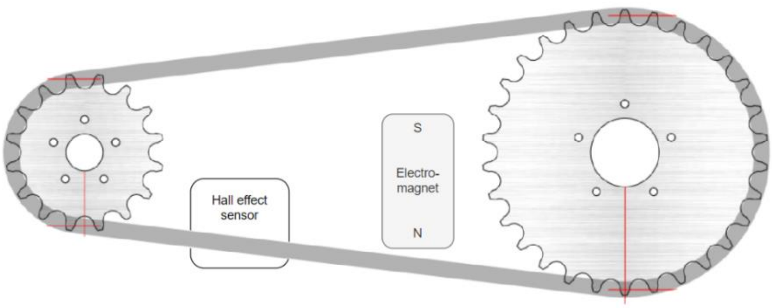


Figure 19. Schematic representation of the proposed magnetic tension sensor.

The proposed sensor works with the same principle as a guitar pick-up. An electromagnet is used to magnetize the chain. When the chain vibrates during operation the magnetic field fluctuates, which is captured by the Hall effect sensor. The frequency of the vibration is found. This corresponds to the first harmonic mode of the chain. A schematic representation of the proposed sensor can be seen in Figure 20. This configuration is arbitrary. The parameters of the electromagnet, such as its number of windings, and its distance to the hall effect sensor will be calibrated via experimentation to best capture the vibration in the chain when the chain sprocket system is actualised.

The second sensor is a ACS712 current sensor connected to the motors. As mentioned before, the applied torque is linearly related to the current drawn by the motor when the motor is in motion. ACS712 is a Hall effect-based sensor and produces an analog voltage signal proportional to the current. It works within ±20A range and has a resolution of 100 mV/A. A picture of the sensor is given in Figure 21. It is connected to the motor circuit in series using the terminals at the top. VCC terminal is connected to the 5V output of the Arduino and the signal (OUT) is connected to one of the analog input channels.

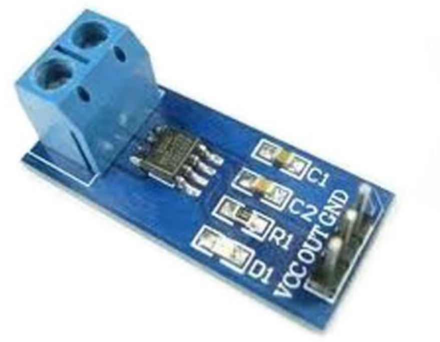


Figure 20. ACS712 current sensor [37].

During operation, the torque values measured from the magnetic and current sensors are averaged. The reason to have two different sensors is increasing the accuracy…

Bearing Calculations

a) Bearing 1 and Bearing 2

Bearing selection is one of the important analysis points in our project. It is necessary to choose the bearing so that it does not fail and does not allow the system to fail. Since it is costly to produce special bearings for the needs, after certain calculations a selection should be made from the catalog. In our analysis, there are 2 basic scenarios: the robot wins or the human wins. Separate calculations are made for these two scenarios and selections are determined according to the most critical situation. In the analyzes we made using Adams, we found that the radial force in bearing 1 in the scenario where the human wins is the highest.

To select bearing, we need to calculate Creq. We can find this with the formula given:

$$C_{req} = F_r K_a \left(\frac{L}{K_r L_R}\right)^{0.3} \tag{7}$$

C_{req} = the required value of C for the application.
L_R = life corresponding to rated capacity
F_r = radial load involved in the application
L = life required by the application
Some corrections factors are used, K_a for shock loading and K_r for reliability.

Ka = 1.75 for “moderate impact” case [25]

Kr = 0.2 for 99% reliability [25]

L = 10^5 required life cycles, assumption

Lr = 90* 10^6 revolution life with 90% reliability

In this case there is no trust load, only axial load, so Fe = Fr. (Fe is used in the presence of thrust load and extra calculations are needed.)

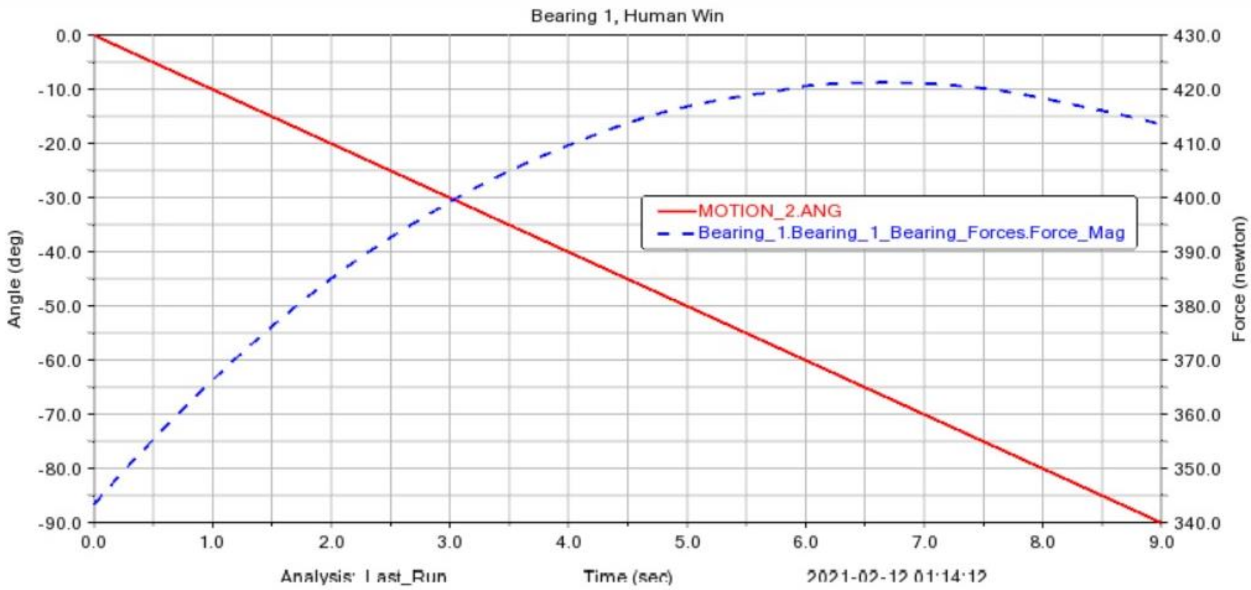


Figure 21. Angular position / time / force graph for Human Win case, Bearing 1

As seen from the graph, the maximum force in magnitude is around 420 N. In order to be a more conservative process, we will continue with 425 N.

Fr = 450 N

Putting all values in Eq (7):

Creq = 165. 83 N

After calculating Creq, bearing from Table 14.2 [25] has to be chosen accordingly. However, if we examine this table, we see that all the bearings we will choose for whatever radius will meet our needs. Even in the most critical case, we did not encounter any restrictions for the bearing. For this reason, we can choose the ball bearing type, taking into account the shaft diameter found in line with other analyzes. At this point, one of the most important reasons we did not have any problems is that we did not set a very high life requirement. Because our robot is not expected to work regularly and for a long time, the required L value is quite low.

b) Bearing 3 and Bearing 4

Same calculations are made for Bearing 3 and 4. All other variables remain constant and only F_r changes. In addition, since the same amount of force will be applied on these two bearings, it is sufficient to consider only one of them.

$$Fr = \sqrt{\left(\frac{F_P}{2}\right)^2 + F^2} \quad \text{---- From Figure 13}$$

Fr = 1718.76 N

Creq = 633. 39 N

Table 14.2 Bearing Rated Capacities, C, for $L_R = 90 \times 10^6$ Revolution Life with 90 Percent Reliability

	Radial Ball, $\alpha = 0°$			Angular Ball, $\alpha = 25°$			Roller		
Bore (mm)	L00 XIt (kN)	200 lt (kN)	300 med (kN)	L00 XIt (kN)	200 lt (kN)	300 med (kN)	1000 XIt (kN)	1200 lt (kN)	1300 med (kN)
10	1.02	1.42	1.90	1.02	1.10	1.88			
12	1.12	1.42	2.46	1.10	1.54	2.05			
15	1.22	1.56	3.05	1.28	1.66	2.85			
17	1.32	2.70	3.75	1.36	2.20	3.55	2.12	3.80	4.90
20	2.25	3.35	5.30	2.20	3.05	5.80	3.30	4.40	6.20

Figure 22. Table 14.2 Bearing Selection [25].

When we look at the Creq values given in Table 14.2 for the bore diameter of 15 mm, we see that our result is lower than all of them. So we can choose the bearing we want. As with previous bearings, we will use radial ball bearing for this. It would be appropriate to use 2 15 mm bore diameter L00 XIt radial ball bearing.

Note: In Turkey, there is a different cataloging system for bearings. Therefore, we need 2 pieces of 6202 and 2 pieces of 6204 ball bearings. These are codes corresponding to 15 and 20 mm bore diameter bearings, respectively.

Chain Calculation and Selection

Roller chains are standardized by ANSI according to size, so we need to select the appropriate chain-sprocket set for our system from the catalog. Chain selections are made according to certain numbers and these numbers are given based on chain pitch, which is shown in the figure above. Numbers 25, 35, 41, for example, correspond to a pitch length of 0.25, 0.375 and 0.5

inches, respectively. While doing our calculations, we decided to use the most preferred chain number 25, which corresponds to 6.35 mm pitch length [26].

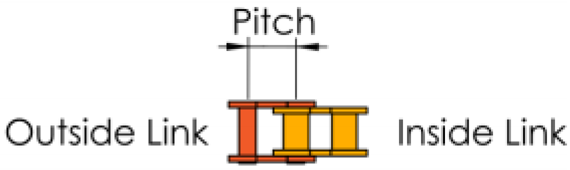


Figure 23 - Chain structure [26].

Roller chains often fail because of long hours of operation. This situation may be caused by the wear of the shafts on the pins or the fatigue of the cylinder surfaces. Therefore, manufacturers in this field have compiled tables giving horsepower capacity corresponding to a life expectancy of 15 kh for various speeds [27]. In order to find the appropriate chain- sprocket set, first of all, we have to find horsepower by the given equation [26]:

$$H_1 = 0.004 N_1^{1.08} n_1^{0.9} p^{(3-0.07p)} \tag{8}$$

N_1 = number of teeth in the smaller sprocket

n_1 = sprocket speer, rpm

p = pitch of the chain, in

There is also H_2 value, which refers for roller limited nominal horsepower, but since we have to dealt with min (H_1, H_2), our calculations are based on H_1 (link-plate limited). On the other hand, this calculation is for 17-tooth sprockets, so we need to correct it again for our own sprocket teeth numbers. Therefore, correction factors should be used [26].

$$H_a = K_1 K_2 H_{tab} \tag{9}$$

K_1 = correction factor for tooth number other than 17

K_2 = strand correction

From Table 17.22, we obtain K_1 approximately as 0.62. From Table 17.23, we obtain K_2 as 1.0 (single strand) [26].

N_1 = 10
n_1 = 10 rpm
p = 0.25 in

Then we put all values into Equation 9:

- **H_a = 3.79 * 10^{-3} hp**
- **Sprocket velocity = 10 rpm**

When we go back to table 17.20 [26] with corrected values, we see that our values in terms of both horsepower capacity and speed are well below those in the table. Because our system works at a very low speed. Also, since it is not expected to run for a long time, fatigue failure is out of the question. From here, we see that we are not faced with any restrictions in terms of the strength and durability of the parts. We can choose a chain-sprocket set that dimensionally fits the system. As we mentioned before, we chose the chain number 25 and the sprockets with 10 and 60 teeth. In fact, no less than 17 is recommended for the minimum number of teeth, but this is a limit for long-term durability. Since long life is not expected in our mechanism, we decided to choose the smallest number of sprockets produced.

Since we know the number of teeth and pitch length, the sprocket diameters are automatically calculated from here. Sprockets with a diameter of 24 mm for 10-teeth and 126 mm for 60 teeth are selected. [28]

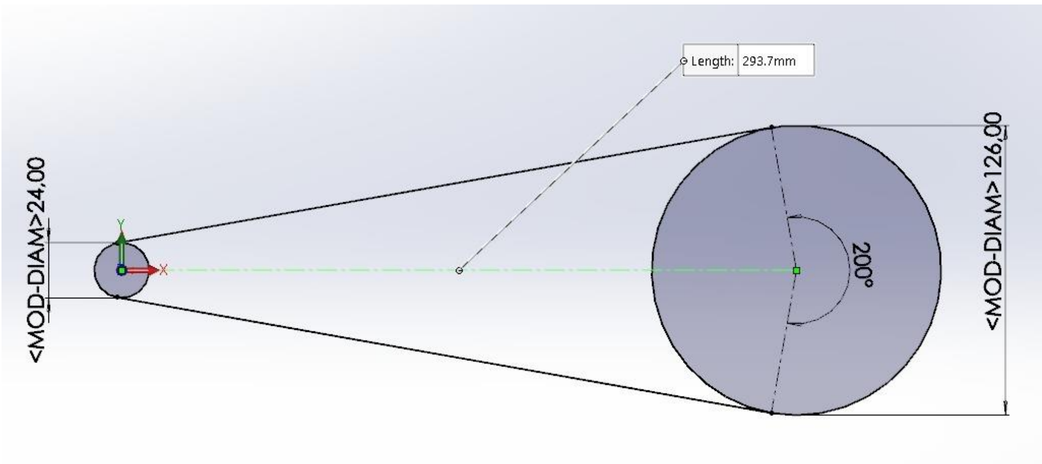


Figure 24. CAD Drawing for chain-sprocket system.

Another important point in chain design is the wrap angle. This shows the contact between the chain and the sprocket. Therefore, this angle is desired to be between 120 ° and 200 °. [27] We realized our design by taking this into consideration while making our drawing. Another important point is center to center distance. At this point, if we are using a chain of a certain length, this distance can be calculated from here, or vice versa, the chain length can be found according to the center to center distance. While we were drawing the design, we determined a center to center distance to fit our system and provide the wrap angle, and this is 293.7mm.

Since we know this distance and the chain pitch, we can now calculate the required chain length from the given equation:

$$\frac{L}{p} = \frac{2C}{p} + \frac{N_1 + N_2}{2} + \frac{(N_1 - N_2)^2}{4\pi^2 C/p} \tag{10}$$

- N_1 and N_2 are the teeth numbers of sprockets
- p = pitch
- C = center to center distance

After we put the variables into the equation we obtain, L = 818.3 mm (chain length)

Modeling and Simulation

a) Algorithm

In order to simulate the motion and to get results of the dynamic motion, it is needed to obtain the sequences of random numbers having different types of distributions. Even though it is called random numbers, most of the software programs use pseudo-random number generation techniques. This technique is especially useful in arm-wrestling robot simulation since these numbers are reproducible. That is, it is easy to simulate the same scenario which is considered important.

The basic idea behind the pseudo-random number generation techniques is to use starting seed and to generate the sequence of random numbers from the seed via recurrence relation. Most of the pseudo-random generators have the following type of relation, where a represents the multiplier, c represents the increment and m represents the modulus. This technique is used for creating uniform random numbers in the simulation via software [29].

$$X_{n+1} = (aX_n + c) mod\, m \tag{10}$$

In the simulation, estimation of arm muscle force distribution is critical to have the correct results. In different scenarios, different mean values are used to represent the people having different strengths since anatomic parameters change from person to person and normal distribution is used to generate the input torque numbers as it is used in the literature [30]. Muscle force and responding torque values are predicted and used with respect to the normal distribution. Since different people have different variabilities, different variance and sigma values are used for different scenarios. As you can see from Figure 26, which is graphed by Python as a normal distribution with a mean 40 and a standard deviation 2. These variable forces are used in the simulation to represent the forces that are given by the human during the simulation.

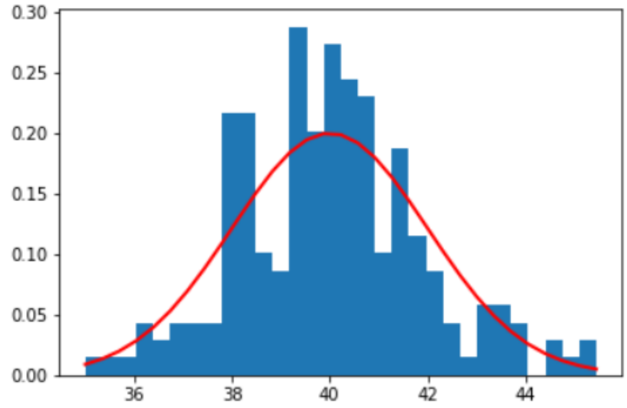


Figure 25. Random numbers generated via Python with normal distribution.

In order to have a fair game, the uniform distribution is used for the rest of the model. Discrete uniform distribution has constant and same probability for each integer in the interval and continuous uniform distribution has constant and same probability for each same length interval [31]. This property is the reason why uniform distribution is used in the model. The sequence of numbers having the same probabilities is used in the model to formulate different probabilities.

b) *Flowchart*

In the model, uniform and normal pseudo-random number generation techniques are used to simulate motion. The flowchart of the algorithm can be seen in the Appendix.

Firstly, the simulation waits 3 seconds and takes torque values from the sensors. After that, it processes these values and assigns the highest one to the starting torque parameter. It also assigns position value 90 and time 0 initially. Then, it goes to the next step and if starting torque value is bigger than 30 Nm, the robot moves 3 degrees in the clockwise direction and waits 2 seconds. Otherwise, it moves in the counter-clockwise direction and again waits 2 seconds. This means that a strong person has a starting advantage. Also, the model updates the position value according to its direction and goes to the next step.

The algorithm reads the torque values in each iteration. For safety concern, if the new torque value is 10 Nm smaller than the starting torque value, the model stops the motion of the robotic arm. If not, it calculates the Formula 1 given below and finds the probability of choosing direction in which the robotic arm would move randomly. Also, time/10 division is an integer division in Formula 1 so that it will only affect after every 10 seconds. Formula 1 calculates the probability of the person winning as follows.

$$\mathit{Formula}\ 1\text{: }[(\text{Torque Value} - \text{Starting Torque Value})\ +$$

$$(3 \text{ if torque value} > 30\text{ Nm}))\ \times 4) - (\text{time}/10) \times 3\ + 50]$$

This formula considers three different factors while calculating the probability of a human win situation. The first factor is the improvement consideration, which means that if the player gets tired and the player gives less torque compared to starting torque value, the win probability of the player decreases. On the other hand, if the player starts giving more torque compared to starting torque value which is measured in the first 3 seconds, the win probability of the player increases. The second factor is the strength of the player, if the player can give more than 30 Nm, the win probability of the player increases with a constant value. The last factor is the time. Every 10 seconds, the robot gets “tired” and probability to win increases for the human. This factor aims to compensate for the fatigue factor of the player.

The probability calculated from Formula 1 is compared with uniform random numbers between 0 and 100. If the probability found in Formula 1 is greater than the generated random number, it moves in the clockwise direction, which means the player wins. Since the probability of random numbers between 0 and 100 are all same in the interval, the player wins in this iteration with the probability found in Formula 1. After finding the direction, it finds X degree defined by random numbers from Formula 2, which generates uniform random numbers between 3 and 7. After that, it moves in the direction found by Formula 1 by the amount of degree found by Formula 2. These two formulas work with the principle of uniform distribution since it is a fair game. Finally, the position value is updated according to direction and amount of movement.

After the movement, the model iterates again only if the updated position value is between the interval of 30 and 120 degrees because of safety concerns. According to Allen Fisher who is Arm Wrestling World Champion, the safety in arm wrestling becomes a problem especially outside of this interval due to the wrong alignment of the arm and the shoulder. If the position value is in the described interval, the model and the robotic arm wait in that position Y seconds which is defined by Formula 3 and saves all the information produced in the iteration. Formula 3 generates uniform random numbers between 0.5 and 2. The model also increases the time value and counts the time from the start at this step.

The second direction motion which is to pull or to push is also random since it depends on the choice of the player. It works in the principle of moving average system and moves in the direction randomly with the value of Z defined by Formula 4 around 50 degrees, which is the initial value of the out-of-plane position, only if the moving average value is between 7.5 and –7.5. Otherwise, the robot stays at the last position and does not change its out of plane position anymore. Since the mean is zero and the interval is small in Formula 4, which generates uniform random numbers between negative and positive values of the same number, it is hard to go outside of this interval when calculating the moving average if the interval is chosen small enough to return to its starting value. However, the model still checks it because of the safety concerns. Formula 4 generates random numbers between –3 and +3 in the first scenario and –2 and +2 in the other scenario. They change according to the player's choice of the importance of the out of plane motion.

b) Scenarios

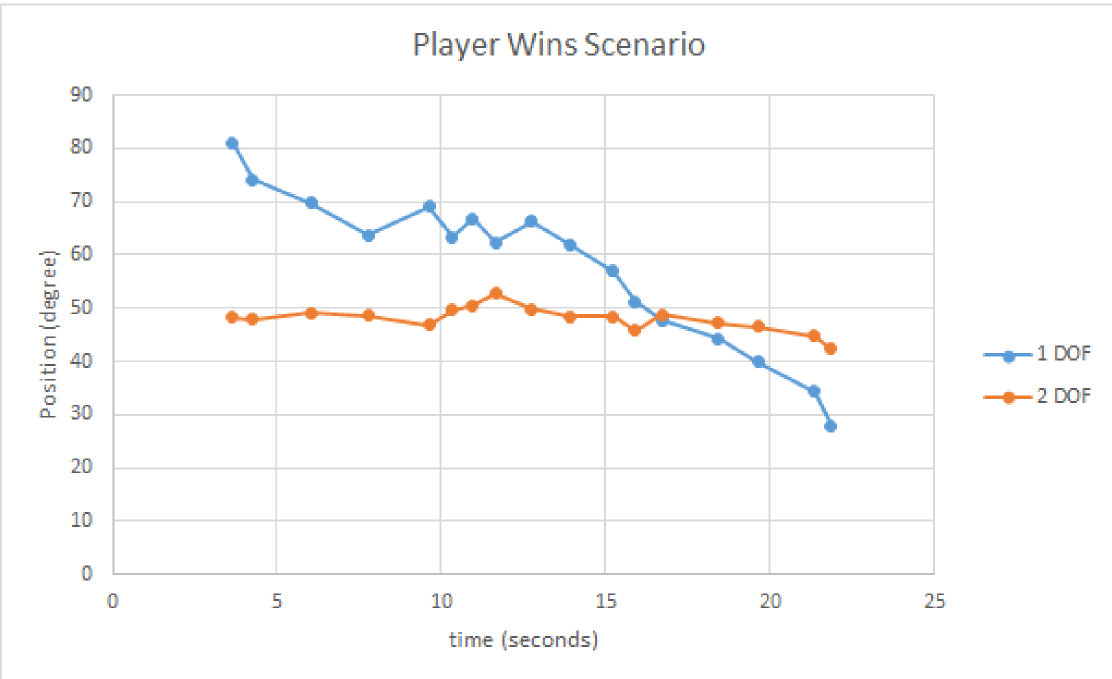


Figure 26. Player Wins Scenario

According to the flowchart, the code is written in Python. Player's torque values generated with normal distribution with mean 40 Nm. As you can see in the graph above, time changes randomly and independently from other parameters, and also the position of out of plane motion which is 2^{nd} DOF also changes randomly and independently. Position of in-plane motion which is shown in the graph as 1^{st} DOF depends on Formula 1. According to Formula 1, the simulation decides which direction it should go and then chooses how much it would go in that defined direction randomly and independently. The amount of movement is between 3 and 7, which are only integer numbers. In the 'Player Wins Scenario', when the in-plane position drops below 30 degrees, the robot stops and the game finishes. Out of plane position can also be seen in the graph below, which moves around 50 degrees. The aim of this movement is to change the out-of-plane position so that players could work in different conditions. This scenario ends after 23 seconds and information measured from the game is given to the player as feedback.

This scenario is simulated in Adams to make the analysis. One of the most critical factors in the control and the simulation of this scenario is that the angular velocity of the robot and motor should stay within the motor's interval. The maximum angular speed that the robot arm move is 10 deg/seconds. In Adams simulation, the angular velocity of in-plane motion and out-of-plane motion is graphed below and as you can see both velocities are lower than the critical value.

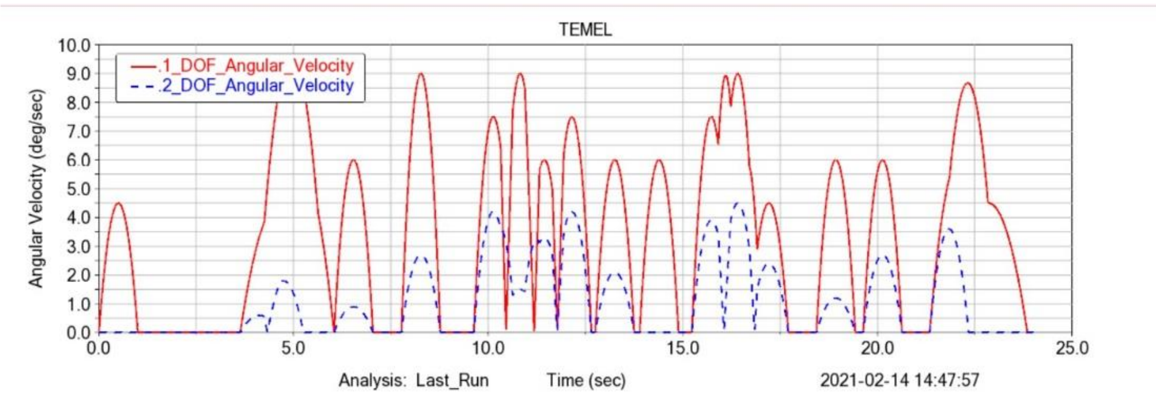


Figure 27. Angular velocities of both positions vs. time

One of the feedback mechanisms is to give the correlation between torque values that the player can give and the position of the arm. This correlation can be seen through every iteration like in the graph below. Since the torque is generated in the model for this scenario, it does not depend on the position value as you can see from the graph. Even though position values decrease from iteration to iteration, the torque values have an almost zero trendline. This feedback mechanism will be given to the user for the 2nd DOF as well to see how good the player is in every position.

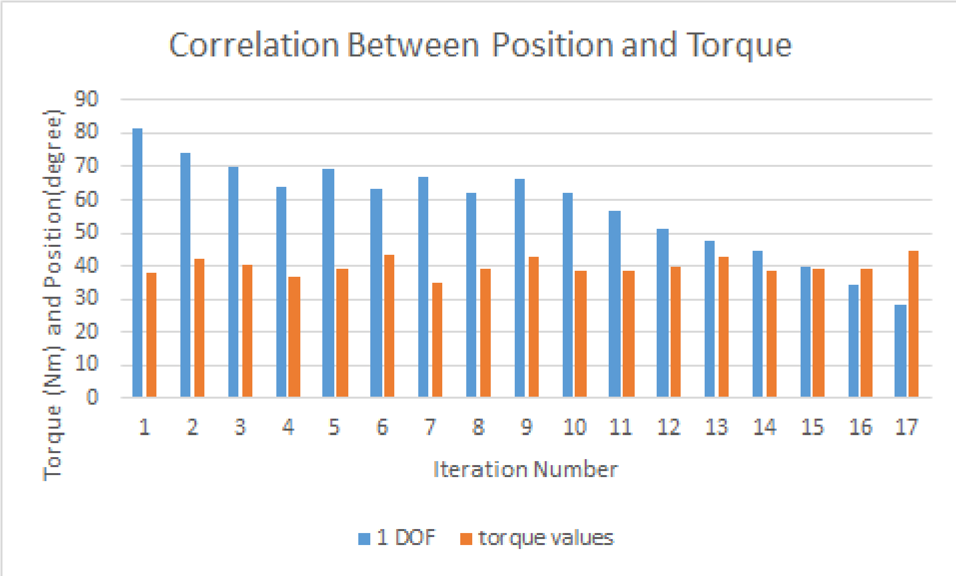


Figure 28. First degree of position and torque values vs. iteration number.

Another feedback mechanism is to measure the fatigue of the player, which means that how the torque values of the player change with respect to time. If there is a sudden decrease in trendline, which indicates that the player gets tired easily. On the other hand, if there is almost no trend, the endurance limit of the player is very good. As you can see from the graph below, torque values are independent of the time and the trendline is almost zero, which is expected in this simulation since torque values are randomly and independently generated.

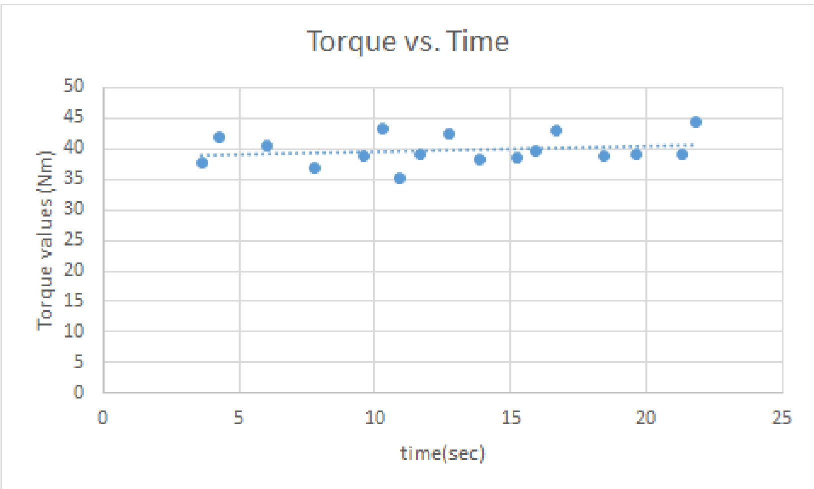


Figure 29. Torque vs. time.

The second scenario which is simulated in Adams is the ‘Robot Wins Scenario’. In this simulation, only the seed of random numbers and the interval of out-of-plane motion is changed and everything stays the same. Even though nothing is changed, the winner changes and this indicates that different random numbers would result in different results in arm-wrestling. This is a strong indication of the fair game. In this scenario, the model stops when the position value is beyond 120 degrees which is the upper limit of the model. Since it is a fair game, even though the player starts winning in the first iterations, the robot wins eventually. The position of out-of-plane motion again varies around 50 degrees.

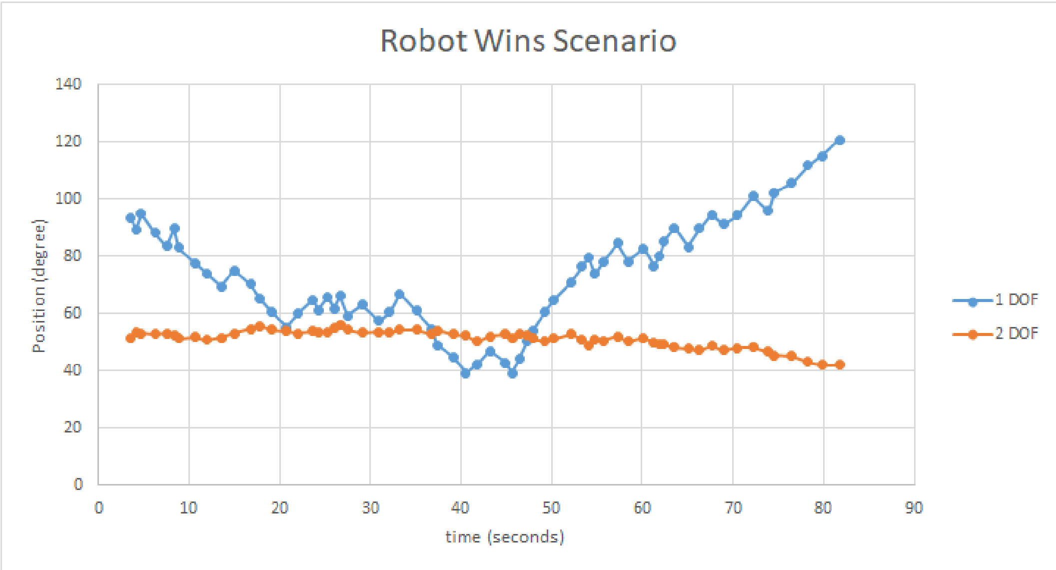


Figure 30. Robot wins scenario

c) Feedback Screen

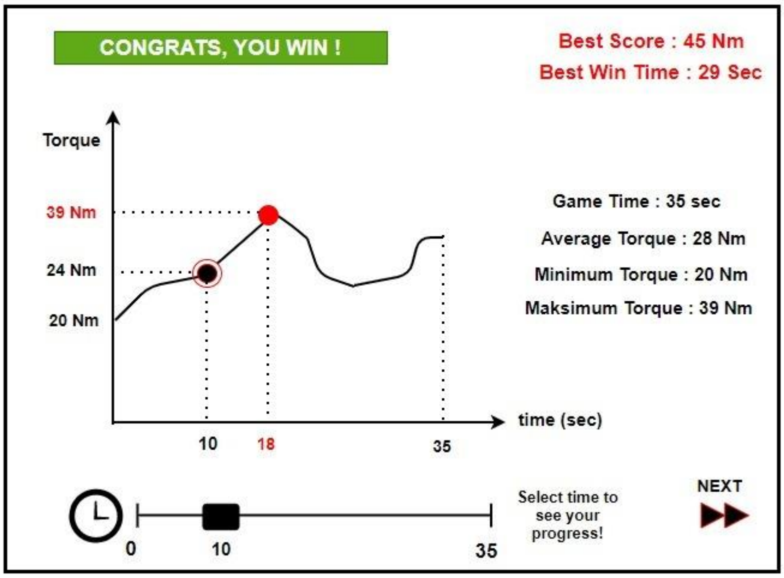


Figure 31. Feedback Screen - Homepage

As we mentioned at the beginning, the aim of our robot while playing games is to ensure the development of the person who is playing, and therefore it has to give him/her some feedback. We designed a feedback screen for this purpose. On this screen, the home page shows the torque vs time graph first. In this way, the person who plays can see how much they have applied at what time. Maximum, minimum, and average torque values are also displayed on the screen. In addition, the elapsed time is also seen. Since the data is stored in the memory, the best score and time before this are also in the upper right corner of the screen.

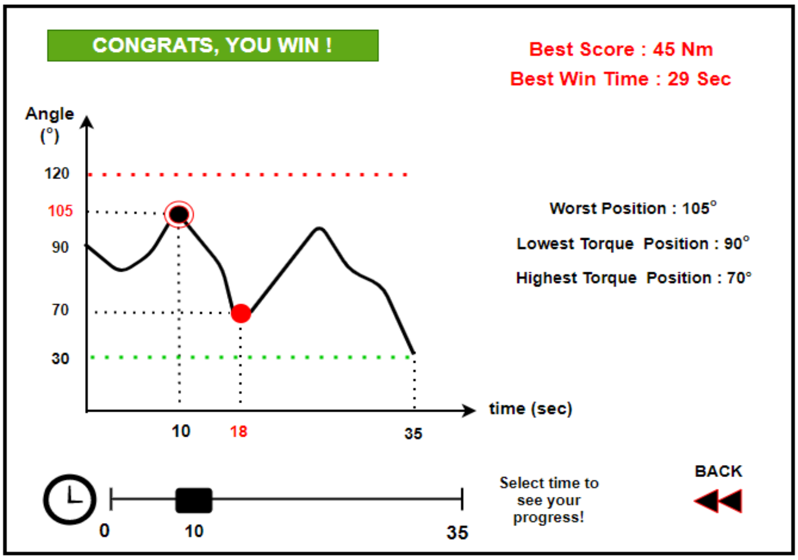


Figure 32 - Feedback Screen Second Page

When you go to the next page, this time, the angle value depending on time is given in the graph. This shows what angle the person playing is at each second. In addition, we can see the angle values where it applies the highest and lowest torque on the screen. With this screen, we will also receive the angle value where the arm is closest to defeat as feedback.

Cost Analysis

Product Name	Quantity	Unit Cost(TL)	Price
Motor Drive	1	216	216
60T sprocket (#25)	2	199	398
Chain (#25)	2	179	358
Motor	2	175	350
10T sprocket (#25)	2	33	66
Current sensor	2	16	32
Hall effect sensor	2	13	26
Bracket	2	12	24

Bearing 15x35x11	2	5	10
Bearing 20x42x12	2	5	10
M5 screw-nut	8	0,25	2
M5 screw	8	0,25	2
M3 screw	2	1	2
Arduino	1	22.89	22.89
TOTAL COST			1533,2 TL

The cost of the shoulder and the arm is not provided, because it is dependant on the method of manufacture, which is not yet determined.

CONCLUSION

In our project, the aim is to enable arm wrestling players to both play games and train themselves with the help of the developed robotic arm. After deciding to use an electric actuator, DC servo motor is chosen for the project. The aim was to design a quiet, clean, and easily usable arm. As a result of the conducted experiments, it is found that a person can apply a maximum torque of around 50 Nm, and a design torque value of 72 Nm is determined by using a safety factor of about 1.5 times. However, most commercial motors cannot supply the required torque. Therefore, we needed an additional system to increase torque and we decided to use the chain-sprocket mechanism. A sprocket system with 1: 6 reduction ratio is used together with a motor that can supply 12 Nm. The system was designed with the help of Solidworks and simulated by using Adams. According to the data which was received from Adams, appropriate material choices are made so that failure would not occur. At this point, AISI 1040 steel was chosen, and a 1.89 safety factor was obtained even at the most critical point. Bearing selections were also made using the force analysis. Bearing and chain should be selected from ready-made catalogs since they are not specially produced to order. Another important part of the project is the algorithm. It was aimed to achieve completely fair games with random number generation on Phyton using a uniform distribution. In other words, whether the human or the robot wins will depend entirely on the current situation, and no unfair situation will occur. During these games, data will be collected at what angle, at what speed, and how much force the person applies, and results will be given to them with a screen help at the end of the game. In this way, it was planned to successfully carry out the training of people, which was the main goal of the project. In other words, it will be a fun competition for the person playing the game, and with the help of the collected data, they will be able to see how they are strong at what point and train themselves in this way. It is planned to make the control mechanism required for the error-free operation of the system, after taking the necessary parts and installing the system, manually controlling it with the code which is processed on Arduino, and obtaining the necessary Kp, Ki and Kd values.

REFERENCES

[1] "History of Arm Wrestling," [Online]. Available: https://sportsmatik.com/sports-corner/sports-know-how/arm-wrestling/history.

[2] "Youtube - Kids Arm Wrestling Against Strong Mechanized Robots," [Online]. Available: https://www.youtube.com/watch?v=BREdjAGlnUA. [Accessed 9 2 2021].

[3] T. Yamada and T. Watanabe, "Development of a Pneumatic Cylinder-Driven Arm Wrestling Robot System," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2008.

[4] C.-G. Kang, K.-S. Ryu, Y. Kim, I. S. Sohn and E. J. Park, "Mechanism and Scenario Design of an Intelligent Arm-Wrestling Machine System," in *International Conference on Control Automation and Systems*, 2004.

[5] H. Kim, L. M. Miller, N. Byl, G. M. Abrams and J. Rosen, "Redundancy Resolution of the Human Arm and an Upper Limb Exoskeleton," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, 2012.

[6] Z. Gao, Q. Song, Y. Nie, J. Lei, Y. Yong ve Y. Ge, «Conceptual Design and Implementation of Arm Wrestling Robot,» %1 içinde *International Conference on Intelligent Robots and Systems*, 2006.

[7] "Complete Guide to Actuators," [Online]. Available: https://www.thomasnet.com/articles/pumps-valves-accessories/types-of-actuators/.

[8] V. O. D. Cura, F. L. Cunha, M. L. Aguiar ve A. Cliquet, «Study of the Different Types of Actuators and Mechanisms for Upper Limb Prostheses,» *Artificial Organs,* cilt 27, no. 6, pp. 507-516, 2003.

[9] «How Does an Electric Actuator Work?,» [Çevrimiçi]. Available: https://www.indelac.com/blog/how-does-electric-actuator-work.

[10] [Çevrimiçi]. Available: https://www.motioncontroltips.com/faq-choose-safety-factor-motor-design-lasts/.

[11] B. Toulas, "How does a Harmonic Drive work? Why are they used?," 2008. [Online]. Available: https://www.engineeringclicks.com/harmonic-drive/. [Accessed 8 2 2021].

[12] D. Collins, "When are worm gears self-locking, and where is this useful?," 2017. [Online]. Available: https://www.motioncontroltips.com/when-are-worm-gears-self-locking-and-where-is-this-useful/. [Accessed 8 2 2020].

[13] "Youtube - What is Strain Wave Gear a.k.a. Harmonic Drive? A Perfect Gear Set For Robotics Applications!?," 2020. [Online]. Available: https://www.youtube.com/watch?v=xlnNj9F37MA. [Accessed 8 2 2021].

[14] "Kinematics in Belt and Gear Driven Systems," [Online]. Available: http://mechanicsmap.psu.edu/websites/11_rigid_body_kinematics/belt_and_gear_driven_systems/belt_and_gear_driven_systems.html. [Accessed 10 12 2020].

[15] J. Warner, «A Look at Belt, Chain and Gear Drive Technology,» 2017. [Çevrimiçi]. Available: https://www.powertransmission.com/blog/a-look-at-belt-chain-and-gear-drive-technology/.

[16] D. Collins, "What's the relationship between current and dc motor output torque?," [Online]. Available: https://www.motioncontroltips.com/faq-whats-the-relationship-between-current-and-dc-motor-output-torque/.

[17] [Online]. Available: https://www.researchgate.net/figure/Torque-measurement-strain-gauges-on-shaft_fig3_323758865.

[18] «Torque Measurement on Rotating Shafts Using Stain Gauge,» BesTech, 9 5 2020. [Çevrimiçi]. Available: https://www.bestech.com.au/blogs/torque-measurement-on-rotating-shafts-using-strain-gauge/.

[19] S. Lawing, "How Does a Pickup Really Work?," 2017. [Online]. Available: https://lawingmusicalproducts.com/dr-lawings-blog/how-does-a-pickup-really-work. [Accessed 10 2 2021].

[20] S. Fong, "Polar S710 + Power Measurement," 2002. [Online]. [Accessed 10 2 2021].

[21] "Introduction: PID Controller Design," [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID.

[22] «Arduino - DC motor speed control PID,» [Çevrimiçi]. Available: https://www.youtube.com/watch?v=Oai8qMH2yTU.

[23] "Hall Effect Sensor," [Online]. Available: https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html. [Accessed 14 2 2021].

[24] R. Nave, "Selenoid," [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/solenoid.html. [Accessed 14 2 2021].

[25] R. C. Juvinall and K. M. Marshek, "Bearing Selection," in *Fundamentals of Machine Component Design*, 5th Edition, p. 606.

[26] J. K. N. Richard G. Budynas, «Roller Chain,» %1 içinde *Shingley's Mechanical Engineering Design*, 9th Edition, , pp. 907-915.

[27] «SPROCKET AND CHAIN GUIDE,» %1 içinde *REV Robotics*, pp. Created by REV Robotics 2018, Licensed Under CC BY-SA .

[28] «Sprocket Diameter Chart By Chain Size,» %1 içinde *Table*, pp. https://www.nitrochain.com/sprocket-diameters.

[29] R. D. Peng, Advanced Statistical Computing, Online, 2021.

[30] Y.-W. Chang, R. H. Hughes, F.-C. Su ve E. Itoi, «Prediction of muscle force involved in shoulder internal rotation,» *Journal of Shoulder and Elbow Surgery,* cilt 9, no. 3, pp. 188-195, 2000.

[31] J. Wiley, «Uniform Distribution: Definitions and Properties,» %1 içinde *Wiley StatsRef: Statistics Reference Online*, 2014.

[32] [Online]. Available: https://www.indiamart.com/proddetail/hydraulic-actuator-14525161288.html.

[33] D. Collins, "How to address overshoot in servo control?," 2017. [Online]. Available: https://www.motioncontroltips.com/how-to-address-overshoot-in-servo-control/.

[34] [Online]. Available: https://www.exportersindia.com/seeco_industries/chain-sprocket-kit-se-9353-new-delhi-india-135535.htm.

[35] [Online]. Available: https://www.processindustryforum.com/article/what-is-a-pneumatic-actuator.

[36] [Çevrimiçi]. Available: https://www.direnc.net/acs712-akim-sensoru-20a-20a-modul.

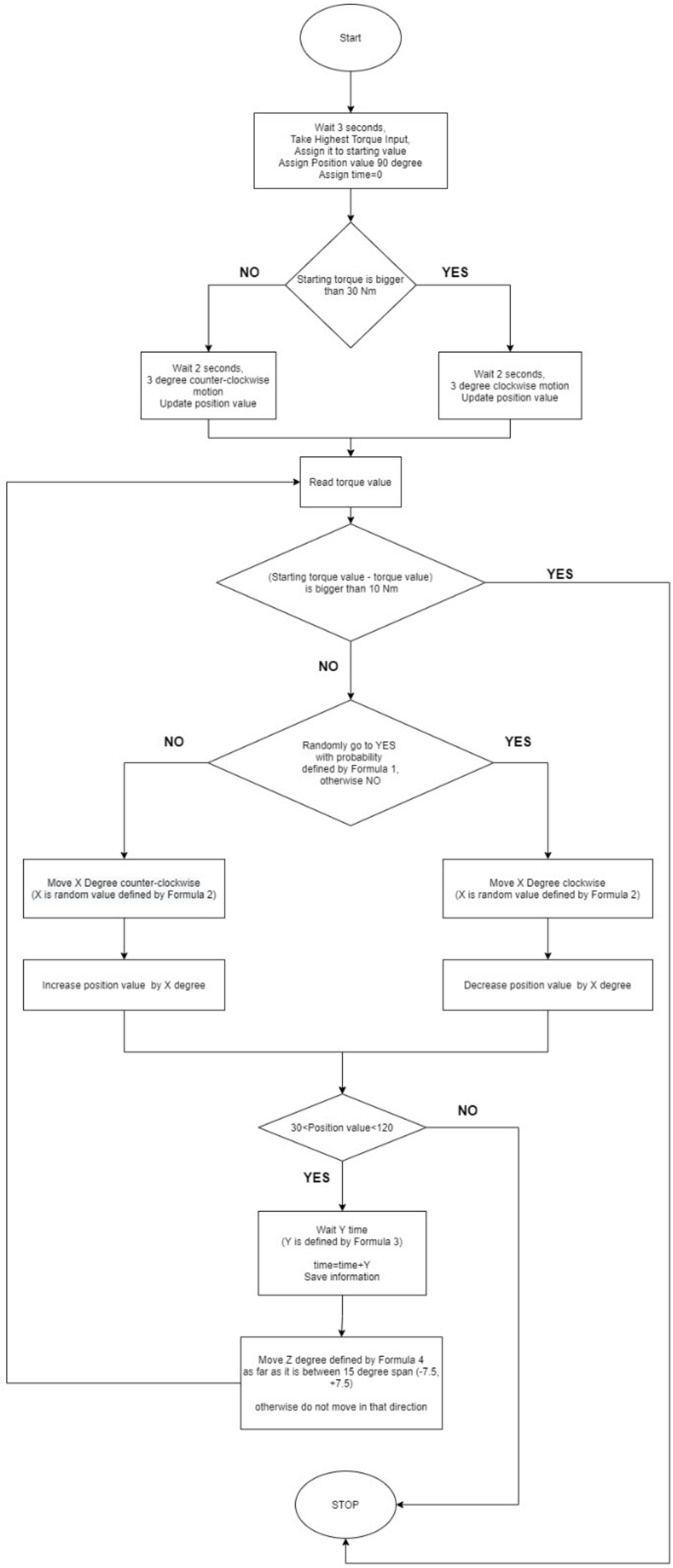
[37] [Çevrimiçi]. Available: https://www.direnc.net/acs712-akim-sensoru-20a-20a-modul.

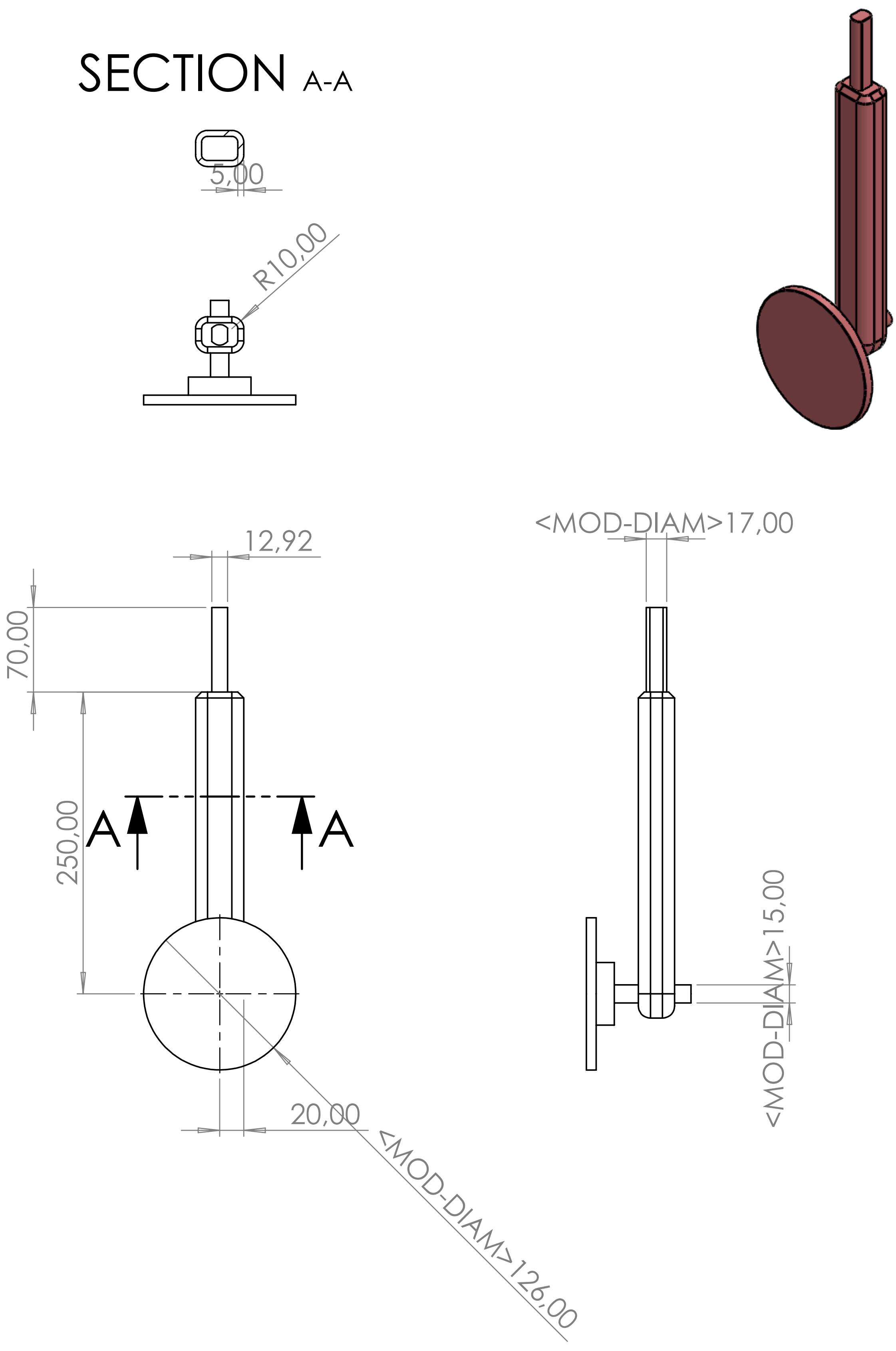
APPENDIX

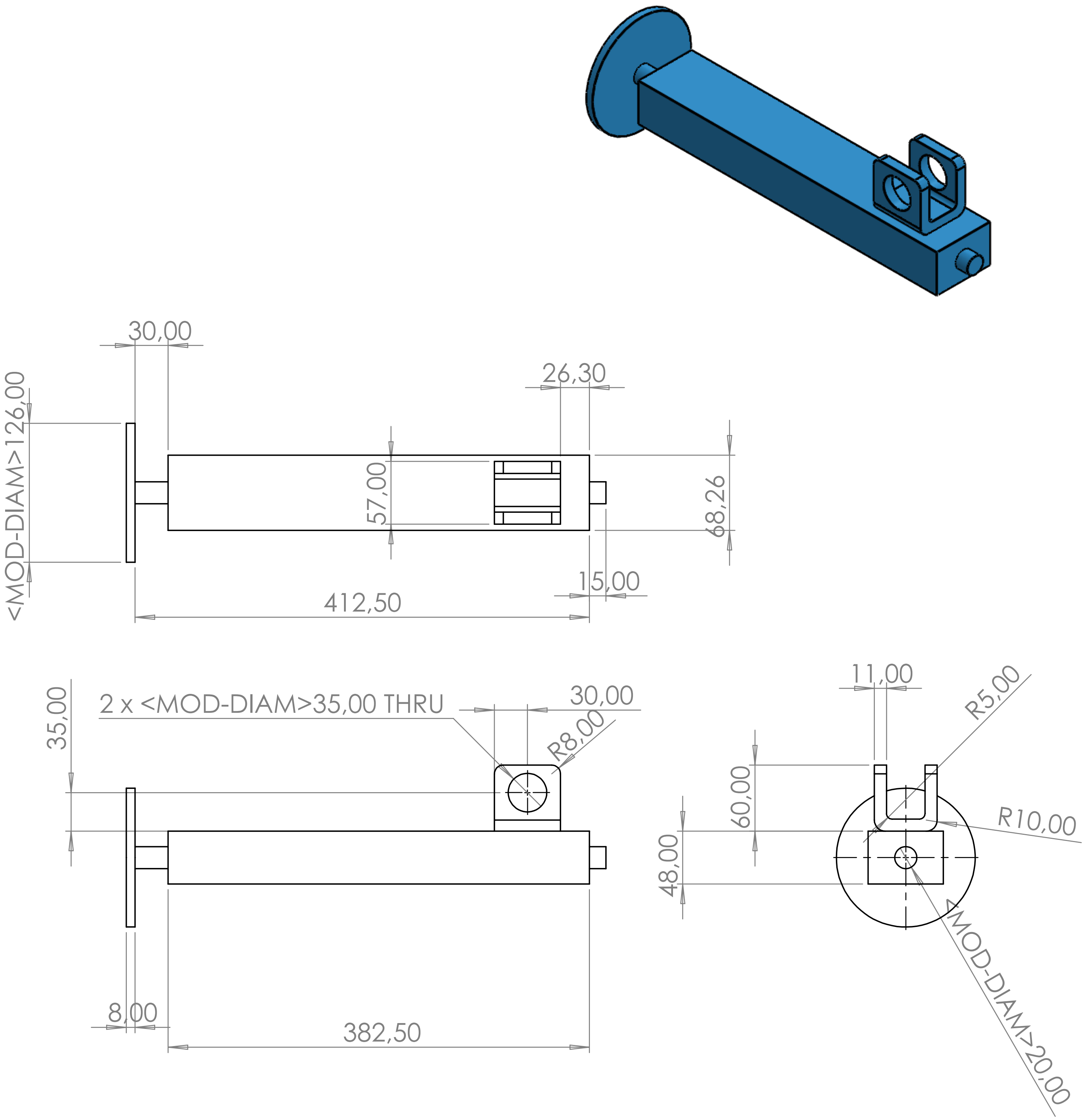
```
from random import seed
from random import random
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(seed=123)
seed(123)
pos=90
second_pos=50
time_temp=0

starting_torque=40

if starting_torque>30:
    pos= pos - 3
else:
    pos=pos + 3

time_temp= time_temp + 2
torque_current= starting_torque

# This is for Torque values, randomly genareted with N(starting_torque, sigma)

sigma = 2                   # standard deviation
torque_array = np.random.normal(starting_torque, sigma, 80)
torque_array
```

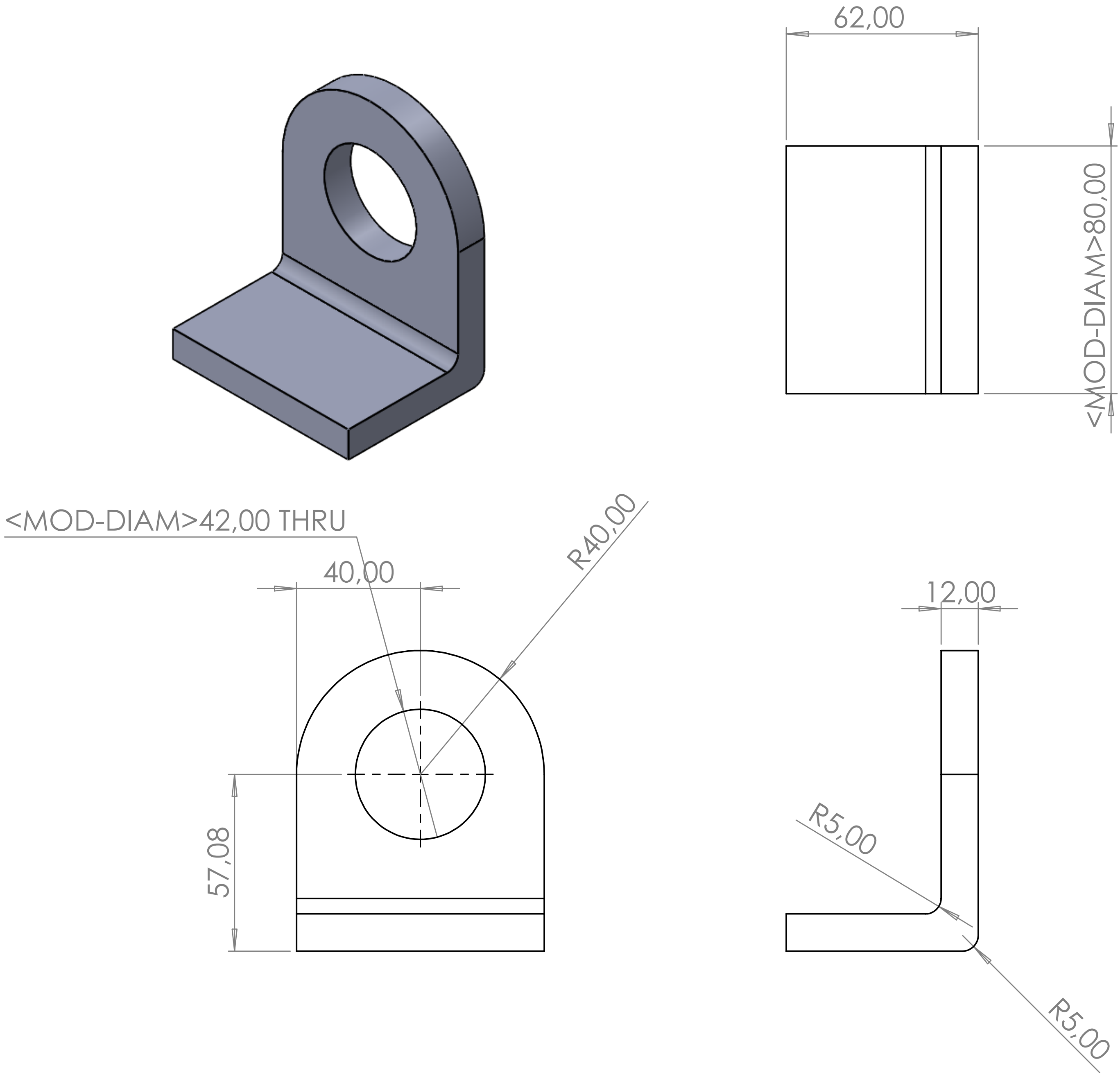
```
# Random Number geneartor for Formula 2

uniform_3to7 =np.round( np.random.uniform(3,7,80), 3)

# Random Number geneartor for Formula 3

uniform_05to2 =np.round( np.random.uniform(0.5,2,80), 2)

# Random Number geneartor for Formula 4

uniform_3to3 =np.round( np.random.uniform(-3,3,80), 2)


# Random Number geneartor for  [0 1]
rank_number= [np.round(random(), 3)]
for i in range(1, 81):
    rank_number.append(np.round(random(), 3))
```

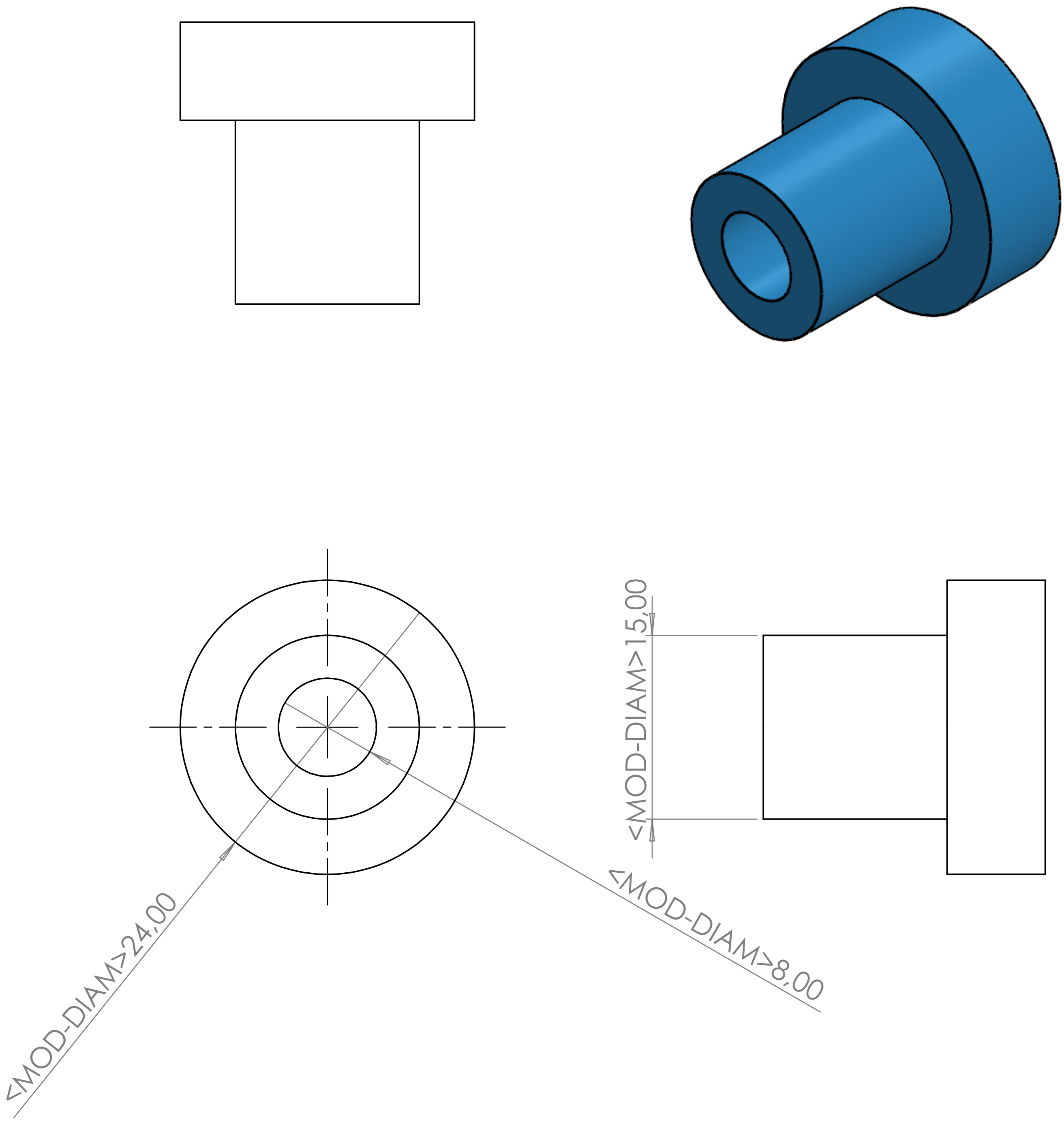
```python
i=0
while pos<120 and pos>30:

    torque_current = torque_array[i]


    if (starting_torque-torque_current)> 10:
        break

    formula_1 = (torque_current-starting_torque)*4  - (time_temp/10)*3  +50

    if torque_current>30:
        formula_1 = formula_1 + 12
    else:
        formula_1 = formula_1 - 12



    if formula_1 > 100 * (rank_number[i]):
        pos = pos - uniform_3to7[i]
    else:
        pos = pos + uniform_3to7[i]


    time_temp= time_temp + uniform_05to2[i]

    if second_pos<57.5 and second_pos>42.5:
        second_pos = second_pos + uniform_3to3[i]

    print( np.round(second_pos, 2))


    i=i+1
```

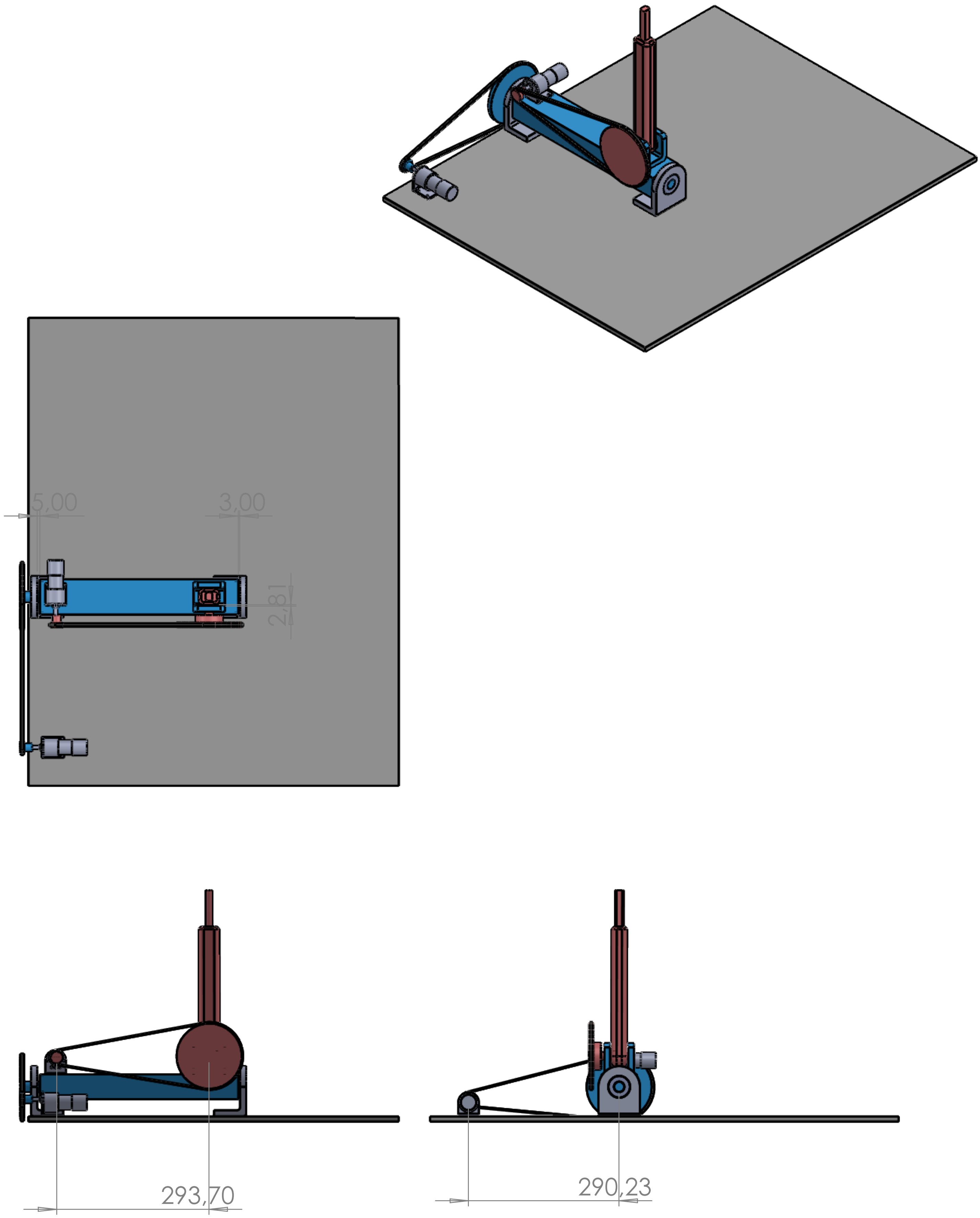

Start
Wait 3 seconds,
Take Highest Torque Input,
Assign it to starting value
Assign Position value 90 degree
Assign time=0
Starting torque is bigger than 30 Nm
NO
YES
Wait 2 seconds,
3 degree counter-clockwise motion
Update position value
Wait 2 seconds,
3 degree clockwise motion
Update position value
Read torque value
(Starting torque value - torque value) is bigger than 10 Nm
YES
NO
Randomly go to YES with probability defined by Formula 1, otherwise NO
NO
YES
Move X Degree counter-clockwise
(X is random value defined by Formula 2)
Move X Degree clockwise
(X is random value defined by Formula 2)
Increase position value by X degree
Decrease position value by X degree
30<Position value<120
NO
YES
Wait Y time
(Y is defined by Formula 3)
time=time+Y
Save information
Move Z degree defined by Formula 4
as far as it is between 15 degree span (-7.5, +7.5)
otherwise do not move in that direction
STOP


SECTION A-A
5,00
R10,00
12,92
70,00
250,00
A
A
20,00
<MOD-DIAM>126,00
<MOD-DIAM>17,00
<MOD-DIAM>15,00


30,00
26,30
<MOD-DIAM>126,00
57,00
68,26
15,00
412,50
2 x <MOD-DIAM>35,00 THRU
30,00
35,00
R8,00
8,00
382,50
11,00
R5,00
60,00
R10,00
48,00
<MOD-DIAM>20,00


62,00
<MOD-DIAM>80,00
<MOD-DIAM>42,00 THRU
40,00
R40,00
57,08
12,00
R5,00
R5,00


<MOD-DIAM>15,00
<MOD-DIAM>24,00
<MOD-DIAM>8,00


5,00
3,00
2,81
293,70
290,23