Bilkent University
Mechanical Engineering Department
ME 482 – Mechanical Engineering Design
Graduation Project Report

Multi-Directional Power Soccer Chair

Group 8

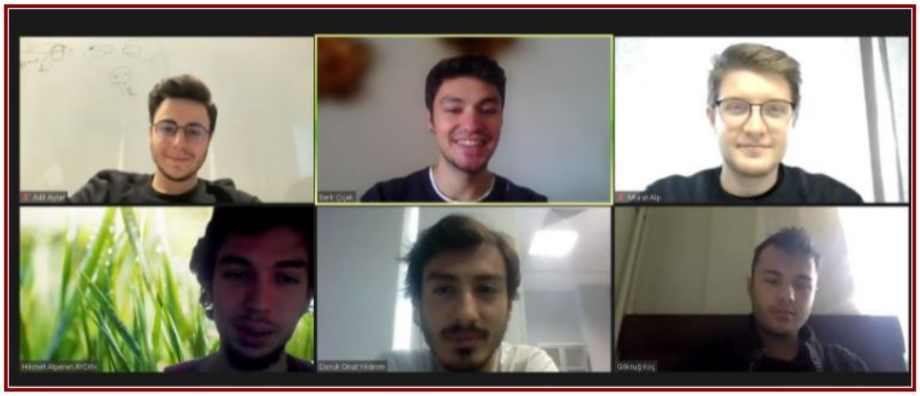
Academic Advisors:
Prof. Dr. Adnan Akay
Dr. Müjdat Tohumcu
Phil Weaver

Industrial Advisor:
Dr. Şakir Baytaroğlu

Group Members
H. Alperen Aydın
Berk Çiçek
Doruk Onat Yıldırım
Murat Alp
Göktuğ Koç
Adil Aytar

Project Assistant: Ayten Gülce Bayram

May 6, 2022



Team members from left to right: *Adil Aytar, H. Berk Çiçek, Murat Alp*
H. Alperen Aydın, Doruk Onat Yıldırım, Göktuğ Koç

Contents

List of Figures

List of Tables

Abstract

Works of this semester is basically composed of two parts, which is the designing and manufacturing of a mini mecanum wheeled prototype, and the design of the actual chair, including the design of the big mecanum wheel. For the mini-prototype, firstly three different mecanum wheel designs are manufactued with 3D printing and tested. Based on those mecanumw wheel tests, the number of rollers is increased to assess the vibration/jumping problem of the wheels, the dimension of the arc of the rollers increased to establish a contact between ground and mecanum wheel, the roller is covered with heat shrink tubes to create enough friction and avoid the slippage, and finally the flange geometry is changed in accord with the changes in the rollers. Having this, the mini prototype is tested using a mini brushed DC motor and a control system which we also use in actual chair. Accordingly, the multi-directional movements are observed on the prototype as desired. For the actual project, further modifications are made on the mecanum wheel design, as on the mini-prototype the rollers can slide back and forth, which was a problem for the actual project, as the chair will carry a significant load unlike the mini-prototype. Therefore, we have added 10 mm protrusion to each inner side of the holes on the flange to stabilize rollers, as it decreases the surface area and the friction. Furthermore, to prevent the shaft movement we observed in mini-prototype, we have place a M4 bolt nest to the outer sides of the holes to fix the shafts that holds the rollers. Assessing those, we have also placed a linear bearings to the inside of the rollers, so that the high load on the rollers can be carried. Testing our completed design on a CAD software, we have started the manufacturing of the flanges of the mecanum wheels with a 3D printing with material being PLA. Furthermore, rollers are made on a CNC machine in machine shop, with material being delrin. The shafts that holds the rollers are purchased from OSTIM and they are cut into the desired size with grinding (94 mm). For the control system appropriate battery is purchased by visiting several battery stores by considering our requirements and the budget. Accordingly, two ORBUS 12V/24 Ah gel batteries are bought, as connecting them in series provides enough voltage for the motors and it can last for sufficient time considering the fair. For the controller we have purchased Dual MC33926 controller, as the motor controller used in mini-prototype cannot draw the required current for the big motors. Lastly, for the chassis an agreement is made with a company on OSTIM (Planet Mekatronik) to build a base from a sigma profile and a sheet metal. For the motors, we have used two pillow blocks, and a motor coupling to lengthen the shaft. Those components are used to ensure that the load on the motor shafts would be distributed to the chassis safely to avoid the motor shaft's failure. Accordingly, the batteries and motors will be placed on that base and the assembled wheels would be attached to run the system with our control model.

1 Introduction

1.1 Problem Definition

With current power soccer chairs, it is not possible to perform most football movements with the battery powered chairs used for power soccer. Power Soccer game plays an important role for disabled people to integrate into society. Thanks to Power Soccer Chairs, disabled individuals can socialize by participating in sports activities and moreover, they can create a new career for themselves by playing this game professionally. With increasing the mobility and the smoothness of the game, the sport will be more attractive and popular. Hence, it increases the number of people integrated into society and increases motivation. Currently, the power soccer wheel chairs are only able to move like a car. Thus, lack of maneuverability makes it hard for players to perform many of the tricks that soccer players without disabilities perform [1]. In order to eliminate this constraint, the goal is to allow the power soccer chair to move in every direction. In particular, the chair should be able to make lateral movement as well as it should be able to rotate around itself. Increased maneuverability, and degrees of freedom would allow players to have more control over the ball and it would also increase the speed of the game; hence, the joy of the game for both viewers as well as players would increase.

1.2 Requirements and Constraints

The constraints and requirements according to the goal of the project are given below.

1.2.1 Regulations

For power soccer wheelchairs, the regulations are determined by FIPFA. Two major constraints introduced by the regulations are given below :

- No way to trap the ball and no concave surfaces are not allowed.
- The maximum speed wheelchair can attain should not exceed 10 kph [2].

Those two are the most strict rules that FIFPA emphasized; therefore those are the main considerations that is considered throughout the design process [2].

1.2.2 Dimensions

FIFPA's regulations are also followed for dimensions, as it can be seen in Fig. 1. Although there is no strict restriction, the dimensions should be such that player should be able to touch the front guard with his/her feet without fully extending his/her leg. Furthermore, the maximum length and the width of the wheelchair is provided in the Fig. 1 [2].

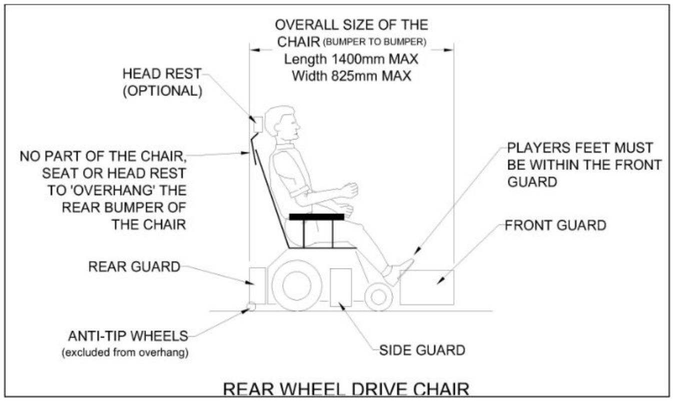


Figure 1: General Structure of Power Soccer Chair

[3]

2 Concept Selection

2.1 Wheels

Since the goal of the project is to provide multi-directional motion, mecanum wheel is selected for the wheels of the chair. Thanks to their design, mecanum wheels can provide multi-directional motion without the need of using a rotary encoder or servo motor. Their comparisons with other wheels such as swivel and omniwheel can be seen in Table 1. Before arriving at final design different mecanum wheel designs are tested with varying plate and roller configurations, as explained in Design section in more detail. Accordingly, the finalized design of the mecanum wheel plate and the roller along with their connecting parts is provided in Appendix B.

		Wheel Concept					
		Mecanum		Swivel		Omni	
Criteria	Weights	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Cost	20	4	0.8	5	1.0	3	0.6
Feasibility	30	4	1.2	3	0.9	5	1.5
Robustness	20	5	1.0	3	0.6	3	0.6
Durability	30	5	1.5	2	0.6	5	1.5
Total Score:		4.5		3.1		4.2	
Rank:		1		3		2	

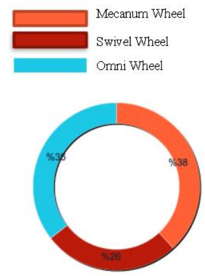


Table 1: Design Matrix for Wheel Selection

2.2 Motor

For the motor, brushed DC motor is selected. It is one of the cheapest option and easy to control. In light of the analysis provided in the detailed design report, the required power to move the chair was found to be approximately 1 kW. Since there would be four motors, 250 W power is required per motor. Accordingly, its comparison with other options such as hub and brushless DC motor can be seen in Table 2.

		Motor Concept					
		Brushed DC Motor		HUB Motor		Brushless DC Motor	
Criteria	Weights	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Design	35	4	1.4	2	0.7	2	0.7
Reliability	30	3	0.9	4	1.2	3	0.9
Cost	20	5	1	4	0.8	4	0.8
Efficiency	15	3	0.45	4	0.6	4	0.6
Total Score:		3.75		3.3		3	
Rank:		1		2		3	

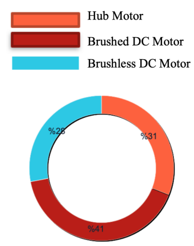


Table 2: Design Matrix for Motor Selection

2.3 Control Architecture

In control algorithms, inverse kinematics of the 4 wheeled mecanum robot is used to determine the required rotation of the each wheel. In addition, DC motor control models are implemented to provide the determined rotational speed to each wheel. Current mecanum wheel robot prototypes use discrete motions such that, robots can move in predetermined directions rather than moving in every direction. In this project, continuous control will be implemented with the help of derived inverse mecanum wheeled robot kinematics.In Fig. 2, the configuration of the detailed circuit design is provided.

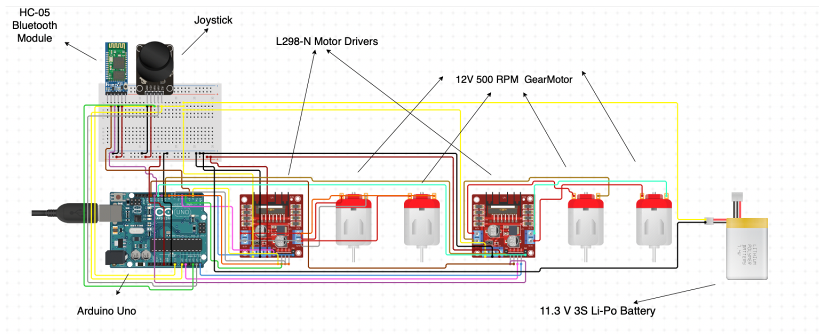


Figure 2: Circuit Scheme

Accordingly, as it can be seen in Fig. 2, for the speed controller L298N is chosen as it allows the simultaneous control of two motors and could withstand the required voltage and the current drawn by the motor. Its comparison with other speed controller options can be found in table 3, and its another advantage is that it is a pulse width modulation controller; therefore, it eliminates the requirement for the usage of extra receiver batteries.

		Speed Controller					
		MOSFET		ESC		L298N	
Criteria	Weights	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Cost	20	5	1	3	0.6	2	0.4
Capability	25	3	0.75	4	1	5	1.25
Flexibility	25	3	0.75	4	1	5	1.25
Accuracy	30	2	0.6	3	0.9	4	1.20
Total Score:		3.1		3.5		4.1	
Rank:		3		2		1	

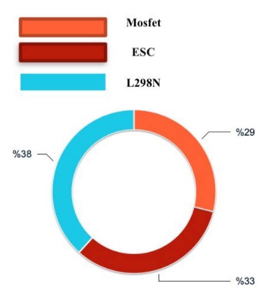


Table 3: Design Matrix for Speed Controller

On the other hand, as it can be seen in Fig. 2, joystick allows us to control the speed and direction of the power soccer chair. Hitting the switches at just the right time in a precise sequence is not easy; therefore the joystick option is more desirable unlike switched systems. Furthermore, Dual Head Array sensors react to proportional input which ensure more intuitive drive. When the situation requires rapidly changing directions, it's very helpful. Dual Head Array also uses two types of sensors: proximity sensors, embedded in the foam, which perceive the head in space; and pressure sensors located behind the foam, which respond to force. However, the immense price of dual head array joystick, makes it undesirable option for us. Hence, we have picked LCD (regular) joystick compared to that as it can also be seen in the design matrix on table 4.

		Joystick			
		Regular		Dual Head Array	
Criteria	Weights	Rating	Weighted Score	Rating	Weighted Score
Cost	20	5	1	1	0.2
Capability	25	3	0.5	5	1.25
Flexibility	25	4	1	3	0.75
Accuracy	30	3	0.9	5	1.5
Total Score:		4.4		3.7	
Rank:		1		2	

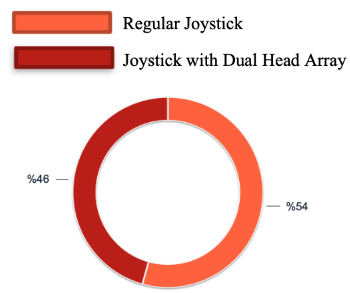


Table 4: Design Matrix for the Joystick

2.4 Battery

In this project, the output values and the capacity of the battery are very important as it should generate electricity during the match period and it should generate enough output values to feed motors to get velocity and control the other mechanisms of the power chair. According to project constraints, the minimum determined volt value is 12V and the maximum determined volt value is 24V. As a result, the gel battery is selected among other options (see table 5), as it has longer capacity and it is the battery type that is currently used in almost every power soccer chairs. Furthermore, gel batteries are resistant to vibration, as it can absorb the impact. The battery that was purchased can also be seen in Fig. 13

		Power Supply Concept					
		AGM Battery		Gel Battery		Lithium Polymer	
Criteria	Weights	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Specification	30	4	1.2	5	1.5	2	0.6
Safety	30	4	1.2	4	1.2	3	0.9
Dimensions	20	3	0.6	3	0.6	4	0.8
Cost	20	3	0.6	3	0.6	5	1
Total Score:		3.6		3.9		3.3	
Rank:		2		1		3	

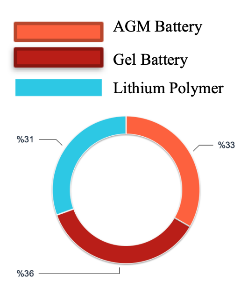


Table 5: Design Matrix for Battery Selection

3 Codes and Standards

Codes and standards that is followed to construct the main design and prototype is presented.

- ISO 16840- 3:2014: This is a standard for wheelchair seating. This code indicates the test techniques to decide ideal wheelchair seating design to endure static and cyclic loading. In addition, it provides disclosure requirements for the hardware components on the wheelchair. In this project, this standard is utilized to take into account the loading and determine the required strength of the chassis. Moreover, it will also be useful when the actual chair is manufactured and the hardware components are attached, as it provides disclosure requirement for the hardware components [5].

- ISO 7193:1985: This standard is utilized while specifying the dimensions of the wheelchair. According to this standard, the dimensiins are resolved in light of mobility as well as accessibility of the engines. It is applied to both manual and electric wheelchairs. Thus, it is used in our project to limit the dimensions of the chassis design. Since maneuverability is significant in a power soccer chair and the accessibility of the motors is important for maintenance, this code was very useful in designing the chassis. [6]

- ISO/FDIS 7176-14: This standard is used for the standardization of the requirements and test methods for electric wheelchairs' power and control system. Thus, the standard is utilized while testing our control system [8].

- IEEE 1187: This standard is for the installation design of the battery. The standard expects basically six centimeters clearance in either sides of the battery [9].

- IEEE 1578: IEEE standard will be utilized for Stationary Battery Electrolyte and Management. This standard is mostly worried about safe use of the battery. It gives control procedures as well as guidelines with respect to fire security. Through electrical maltreatment tests, it informs how batteries act during dangerous circumstances [11].

- ISO 7176-7:1998: This standard will be adhered to measure the dimensions of the wheelchair. It gives standard strategies to measure the dimensions of the wheelchair. For example, leg length is measured by removing the seat pad and measuring the distance between back tip of the cage and the front tip of the seat cover. On the other side, the seat width is measured by determining the distance between the outer edges of the seat covers at the rear of the wheelchair [13].

4 Design

Our design mainly consist of two important concepts. One is mecanum wheel which gives the system omni-directional movement capability. The other one is body of system, which includes chassis, controller, battery unit, motor and transmission system. The Fig. 3 shows the finalized design including all the components.

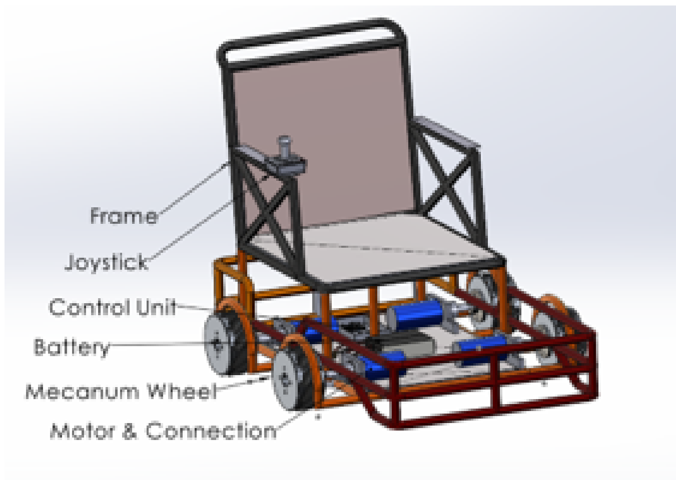


Figure 3: Detailed Design

4.1 Mecanum Wheel Design

In the conventional power soccer chairs, mecanum wheel is not used and price of mecanum wheels are highly expensive since it requires sensitive design and manufacturing process. Therefore, design of mecanum wheel is especially important and should be analyzed under a separate heading.

In the Fig. 18, plates indicates the flange of wheel. There are two flange; one is without shaft and other is together with shaft. There are 10 holes which are 45 degree tilted and aligned uniformly with opposite side (see Fig. 4).

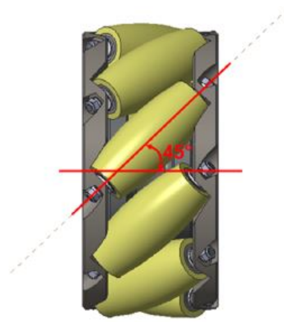


Figure 4: Tilt angle for the holes

In addition to conventional mecanum wheels, in our design flanges have 10 mm protrusion in order to stabilize rollers without sliding on the inner shaft. This protrusion helps the rotation of rollers with decreasing surface area, hence the friction. It was the precaution and development after we faced with sliding problem in prototype. In addition to protrusion, there is one more development was added after the prototype stage; M4 bolt nest is placed to the outer side of the holes in order to prevent shaft movement while the wheel is moving. With these two method first we prevent the sliding of roller on the shaft and secondly the inner shafts are fixed.

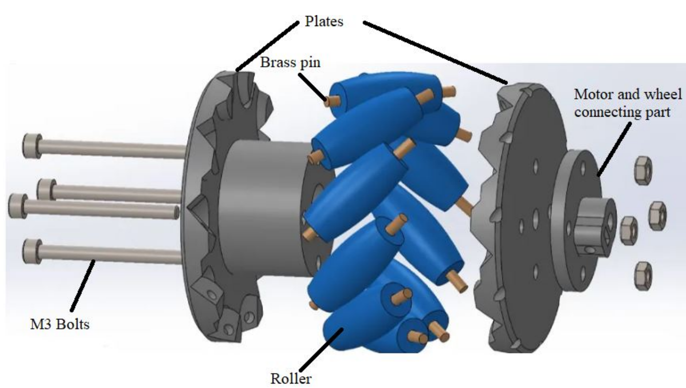


Figure 5: Final mecanum wheel design

For the inner shaft 8mm mercury steel are used for it's precise tolerances. Inside of the rollers, linear bearings are placed since there is high load on rollers. Bearings are fixed by both inner side and outer side. After all the components are placed, fasteners are used to connect all the parts.

4.2 Chassis Design

There are three different frame for power soccer chair and the possible structures for the frame are discussed in previous reports. The chosen structure is rectangular cage design and this is also used in the current power soccer chairs. The advantages of this structural design is impact and movement capability. The rectangular frame made up of aluminium 6063. That is also provide weight gain in positively. The sharp corners of the rectangular frame also another advantages for players. This frame enable user to move faster and hit the ball faster as such design more lighter and create more impact. The determined design shown on Fig. 6. Moreover, further pictures can be found in Appendix A.

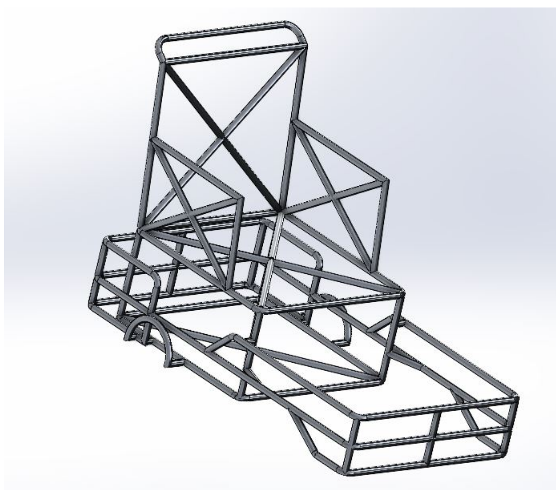


Figure 6: Illustration of designed structure of frame

4.3 Controller Design

To provide uninterrupted control, inverse kinematics are implemented to the software. With that way, velocity in x and y coordinates as well as angular velocity of the body are given by the user as a joystick input and wheel speed is calculated with inverse kinematics. For visualization purposes, forward kinematics are implemented as well. Forward kinematics are used to draw a path and test the controllers. Detailed MATLAB function can be found in Appendix E. Implemented Arduino codes can be found in Appendix F. MATLAB library Mobile Robotics Simulation Toolbox provides the necessary tools.[15] In addition, Arduino library for algebraic operation is also used for implementation on Arduino. MATLAB simulation for the path of the mecanum wheeled robot for given velocity inputs can be seen in Figure 7. Simulink model for the full robot simulation including velocity and rotation inputs can be found in Figure 8.

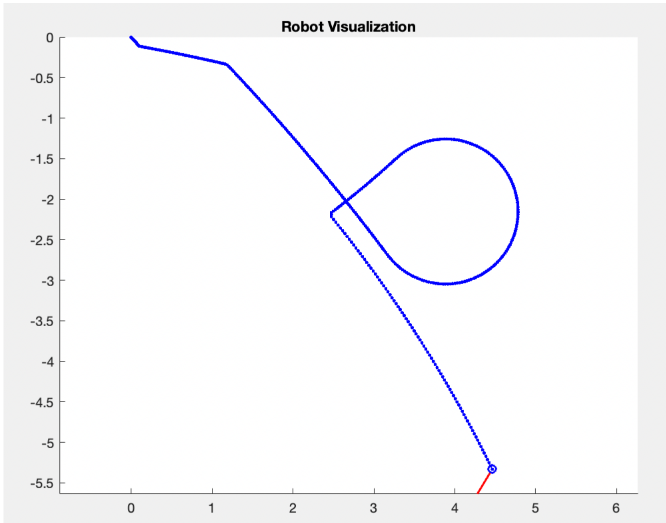


Figure 7: Visualized Robot Control on MATLAB

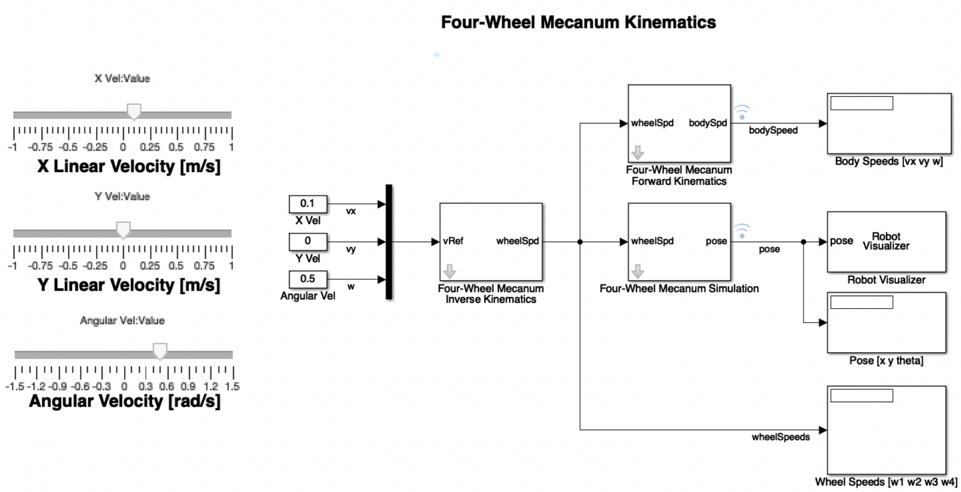


Figure 8: Simulation for Robot Motion

5 Prototype

5.1 Mecanum Wheel Prototype

In this part, the manufactured mini mecanum wheel for the mini prototype is discussed. As mentioned, mecanum Wheels are composed of plates, rollers, and pins (see Appendix B). Plates and rollers are produced with the additive manufacturing method (3D printing) as is mentioned above. Since the manufacturing of the mecanum wheels requires tight tolerances, additive manufacturing is preferred. As it is illustrated in Fig. 9 the aluminum coupler, rollers, inner shafts and the flanges are assembled successfully thanks to the tight tolerance of CNC machine and the method of additive manufacturing. Due to the fact that the movement of the mecanum wheels to be towards the desired direction, the importance of the tight tolerancing must be ensured in order not to experience vibrations and the slippage in the wheels.

Figure 9: mini mecanum wheel

5.2 Controller Prototype

Controller system of the mini prototype follows the same principle with the actual project, which is discussed in section 5.3.Furthermore, the circuit design is provided in Fig. 2, and the assembled circuit for the mini prototype can be observed in Fig. 11. On Fig. 10, the components that are used in the control system is provided. In that figure from left to right, the components are as follows: Dual MC33926 motor controller for the actual chair, L298N motor controller for the mini prototype, Ardunio UNO, and a joystick with a shield.

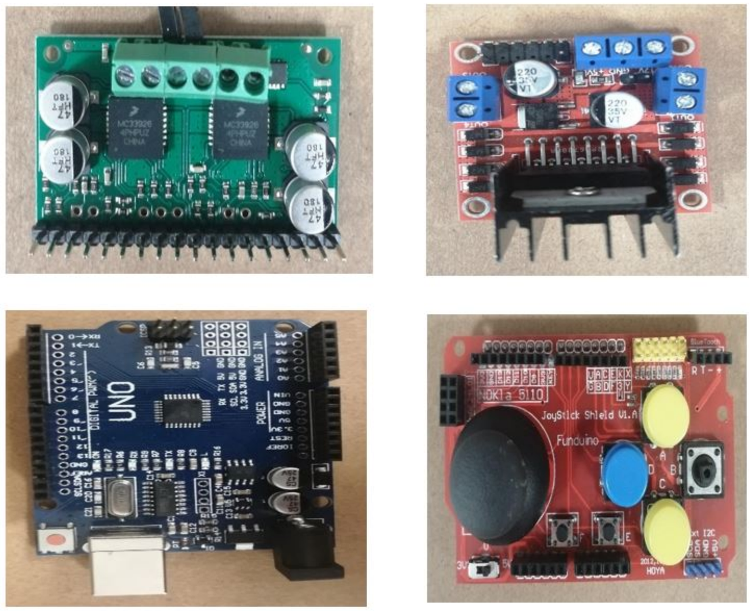


Figure 10: Controller components

5.3 Prototype of Whole System

The mini prototype is assembled with our bare hands. For the assembly of the mecanum wheels, we combined the plates and rollers that we took out of the 3D printer and assemble them using the connecting parts. The assembled configuration of the mecanum wheel is provided in Figure 9.First, we start with the assembly of the mecanum wheel. We manufactured the wheelbase plate as two different components, roller and shaft coupler through the 3D printer. Roller shafts are obtained by cutting with a saw as a piece of 3mm diameter brass rod. Wheelbase plates and shaft are mounted on each other through M4 bolts and nuts. M3 bolts are used for fasteners. Rollers are assembled together with a shaft coupler. The first design was created for the stepper motor but due to the higher cost and controller need, we alternatively solve the motor need with geared DC motor. After the box was manufactured with a 3D printer we also printed the motor case as a holder of a DC motor. M3 bolts are used to mount the motor coupler to the box. Then, motors are connected to the main circuit. Used electronics are basically, Arduino for the controller, motor controllers and Bluetooth receiver. Li-Po batteries for power supply. Electronic parts are combined through cables and holders. Fig. 11 is the finalized version of the mini prototype. The finalized mini prototype is controlled with a Bluetooth module, and it successfully completes the multi-directional motions, as desired.

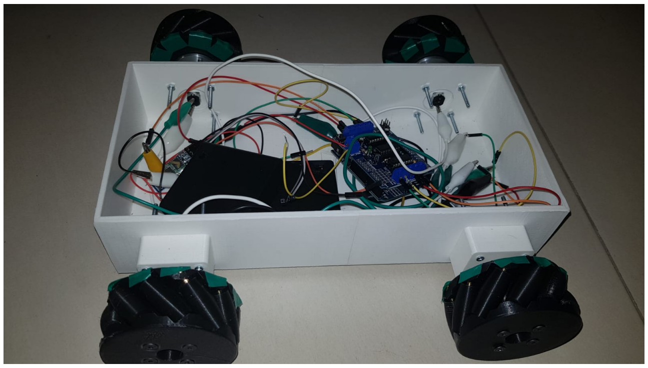


Figure 11: Finalized mini prototype

5.4 Manufactured Mecanum Wheel

In this section manufactured mecanum wheel for the actual chair is provided. Parts are manufactured as indicated in the design section. Rollers are made from Delrin in CNC and we attach linear bearings into the rollers. The two flanges are manufactured with additive manufacturing with material being PLA. Fig. 12 is the assembled version of the mecanum wheel for the actual chair, and its technical drawings are provided in Appendix D. The completed mecanum wheel is manually tested on the ground and it rolled, as desired (mecanum wheels should go with a slight angle when a force is applied parallel to the ground).

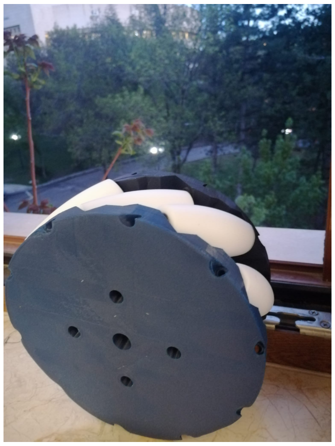


Figure 12: Assembled mecanum wheel

5.5 Battery

Before select the correct battery type, first power requirements of the system is analyzed. Eventually, it is decided that 24V and 24Ah power is adequate. Therefore, two Orbus 12V/24Ah gel batteries are connected in series and used in system. In Fig. 13 you can see the battery. Batteries are connected to system towards, motor controllers and power up the Arduino and system.

Figure 13: Purchased Battery

5.6 Brushed DC Motors

For motors important selection criteria were torque and power need. Besides, for economical solutions pre-purchased motors are selected which is available in the stock of machine shop, Bilkent. There are four 24V brushed DC motor. They combined with shaft motor coupler for transmission. In Fig. 14, the shaft motor coupler (red part), which is manufactured in OSTIM, Planet Mekatronik can be seen. Moreover, pillow block ball bearing attached to the motor shaft can also be seen in Fig. 14

Figure 14: Brushed DC motors

5.7 Chassis

Chassis is one of the main elements of the power soccer chair. All components of the power soccer chair is connected to the chassis including the mecanum wheels. Since the chassis is one of the most crucial elements of the design, the required analyses are employed carefully. The analyses are illustrated in the Appendix I.

Initial design of the chassis was included aluminum 6061-T5 hollow cylinder bars. The bars are planned to welded into each other. However, due to the rough process of welding aluminum bars and the high expenses of the process, it is decided to change the chassis material as the 30 x 30 mm closed sigma profiles. Thanks to the lightness of the sigma profiles, one can obtain mechanically strong and light chassis. Furthermore, sigma profiles offer easy of assembly. The length and the width of the chassis can be adjusted easily if any change is required in the geometry. Thus, flexibility of changing the dimensions, lightness and well-distributed load along the chassis can be obtained using the 30 x 30 closed sigma profiles. As a material of the chassis, aluminum 6063-T5 is selected similar to the previous design. In addition to that, sheet metal is located on the sigma profile in order to obtain more space for cabling, batteries and motors. The design is illustrated in Fig. 15. In light of the analyses provided in Appendix I, 3 mm of thickness is found out to be adequate for operating loads.

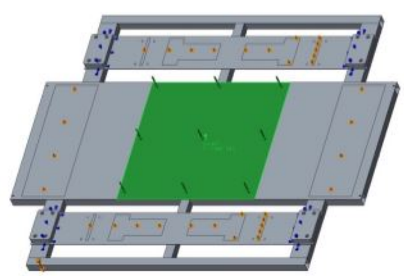


Figure 15: Finalized platform for the chassis

Furthermore, the motor chassis connection in detail can be found on Appendix H.

6 Conclusions and Discussions

In conclusion, the main goal was to provide the multi-directional movement to the power soccer chairs, which includes a rotation around itself and a diagonal motion. Accordingly, manufacturing a mini mecanum wheel, and using the controller architecture mentioned in Section 4.3, we have successfully obtained the multi-directional motion in our mini-prototype, which is controlled with Arduino and a Bluetooth module. For the actual chair, manufacturing the mecanum wheel was the hardest part due to its high cost and the limited budget of the project. Although, we need to use aluminum for the flanges of the mecanum wheels based on our static analysis, we had to change that material to PLA, so that it would be cheaper and easily manufactured by additive manufacturing. Therefore, because of that whether these wheels can withstand a seating of a person is questionable. For the chassis, deciding to go for a base composed of a sigma profile and a sheet metal (see Fig. 15), instead of the actual power soccer chair chassis we have designed (see Fig. 6), is again because of the budget of the project, as the latter option requires welding, which is very costly. All these decisions are made by getting multiple pricing offers from multiple companies; therefore, decisions are made carefully and neatly. Furthermore, original power soccer chairs can reach up to speed of 10 kph; therefore, based on our analysis in our previous reports, we have concluded that the required power for the motors is 250 kW. However, due to the limited budget, we used the brushed dc motors available in the stock of the machine shop, so that no budget is allocated to the motor. Therefore, it would not be possible for chairs to reach that speed with that motor power (90 kW). However, in overall, the main goal was to provide the multi-directional movement, and our mecanum wheel design, and the control system can easily be implemented to other systems (not limited to power soccer chairs). If the platform we made from sheet metal and sigma profile provides the multi-directional motion as desired when it is controlled by a joystick, it can easily be evolved into a power soccer chair by using more powerful motors, stronger material for the mecanum wheel, and building the frame of the chair on top of our platform. Furthermore, it can also be implemented to other systems such as forklifts. Thus, in light of the limited budget we can conclude that we achieved our goal, which can lead to promising further applications. In overall, our work is balanced equally, as we were divided into the control system team, the chassis team, and we worked on the mecanum wheel design all together, as it is the most challenging part of this project. Furthermore, our device would be even safer than the current versions, as it provides more maneuverability; therefore, players/users can avoid accidents more easily.

References

[1] Power chair soccer, Youtube, 2019. [Video file.]. Available: https://www.youtube.com/watch?v=OAt7tFpkCQs

[2] "Powerchair Football - FIPFA." [Online]. Available: https://fipfa.org/wp-content/uploads/2017/08/FIPFA-Laws-of-the-Game-Approved-December-2010.pdf.

[3] "Featherweight 33 lbs. power chair," USA Store View. [Online]. Available: https://www.1800wheelchair.com/product/featherweight-power-chair-only-33-lbs/.

[4] "Featherweight 33 lbs. power chair," USA Store View. [Online]. Available: https://www.1800wheelchair.com/product/featherweight-power-chair-only-33-lbs/.

[5] "ISO 16840-3:2014," ISO, 01-Aug-2015. [Online]. Available: https://www.iso.org/standard/61349.html. [Accessed: 01-Nov-2021].

[6] "ISO 7193:1985," ISO, 03-Jul-2012. [Online]. Available: https://www.iso.org/standard/13810.. [Accessed: 01-Nov-2021].

[7] "ISO 7176-6:2018," ISO, 22-Jun-2018. [Online]. Available: https://www.iso.org/standard/70589.html. [Accessed: 01-Nov-2021].

[8] "IEEE SA - IEEE 1187-2013 - IEEE recommended practice for installation design and installation of valve-regulated lead-acid batteries for stationary applications," SA Main Site. [Online]. Available: https://standards.ieee.org/ieee/1187/4342/. [Accessed: 05-Mar-2022].

[9] "ISO 7176-7:1998," ISO, 05-Mar-2019. [Online]. Available: https://www.iso.org/standard/13783.html. [Ac- cessed: 01-Nov-2021].

[10] "ISO 7176-14," ISO, 23-Oct-2021. [Online]. Available: https://www.iso.org/standard/72408.html. [Accessed: 01-Nov-2021].

[11] "ISO 7193:1985," ISO, 03-Jul-2012. [Online]. Available: https://www.iso.org/standard/13810.html. [Accessed: 01-Nov-2021].

[12] ISO 7176-6:2018," ISO, 22-Jun-2018. [Online]. Available: https://www.iso.org/standard/70589.html. [Ac- cessed: 01-Nov-2021].

[13] "ISO 7176-7:1998," ISO, 05-Mar-2019. [Online]. Available: https://www.iso.org/standard/13783.html. [Ac- cessed: 01-Nov-2021].

[14] "ISO/TC 164 - mechanical testing of metals," ISO, 20-Oct-2021. [Online]. Available: https://www.iso.org/committee/53550.html. [Accessed: 01-Nov-2021].

[15] MathWorks Student Competitions Team (2022). Mobile Robotics Simulation Toolbox. Available: https://github.com/mathworks-robotics/mobile-robotics-simulation-toolbox, GitHub. [Accessed: 06-March-2022]

Appendix A- Final Chassis Design

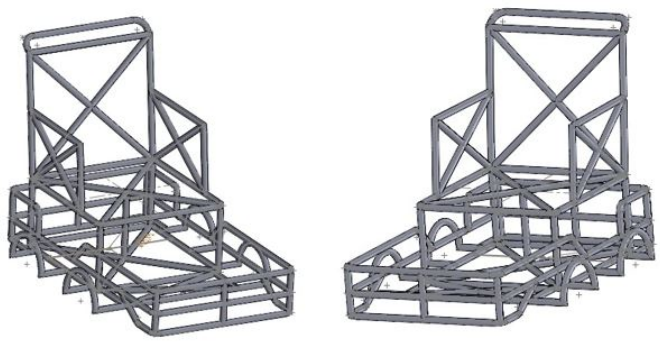


Figure 16: Chassis Design

Appendix B- Final Mecanum Wheel Design

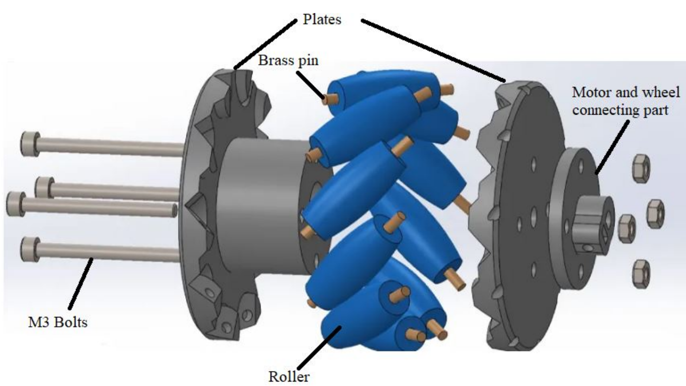


Figure 17: Final mecanum wheel design

Figure 18: Manufactured and assembled final mecanum wheel design

Appendix C- Control Algorithm

Algorithm 1 Main function

Set pin modes according to connection scheme

Initialize motors

Initialize communication channels as data

Initialize inverse kinematic matrix as inverse kinematic

Initialize motor control model as motormodel

while $data \neq 0$ **do**

  Process the incoming data to get required speed and direction

  Index the data to map individual wheel speed

  Put indexed data to inverse kinematic matrix to get motor rpm for individual wheels

  Put individual wheel speeds to motormodel to get required motor voltage

  call setspeed() function for each motor with required motor voltage

end

Algorithm 2 setspeed() function

Input: Required motor voltage, assigned pin

Map input voltage to a PWM signal between 0-255

Output: Send PWM to motors

Appendix D- Technical Drawings

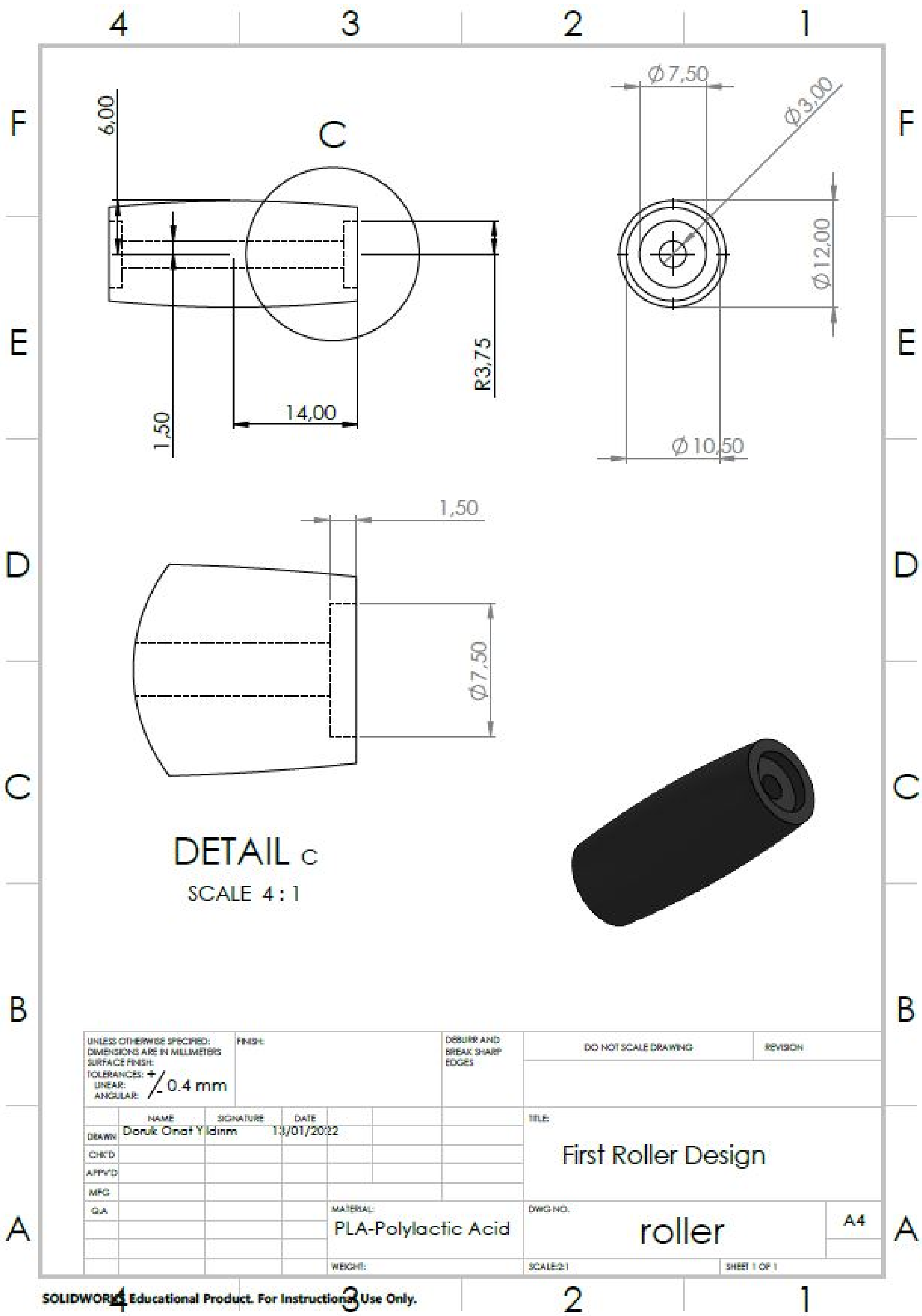


Figure 19: First Roller Design

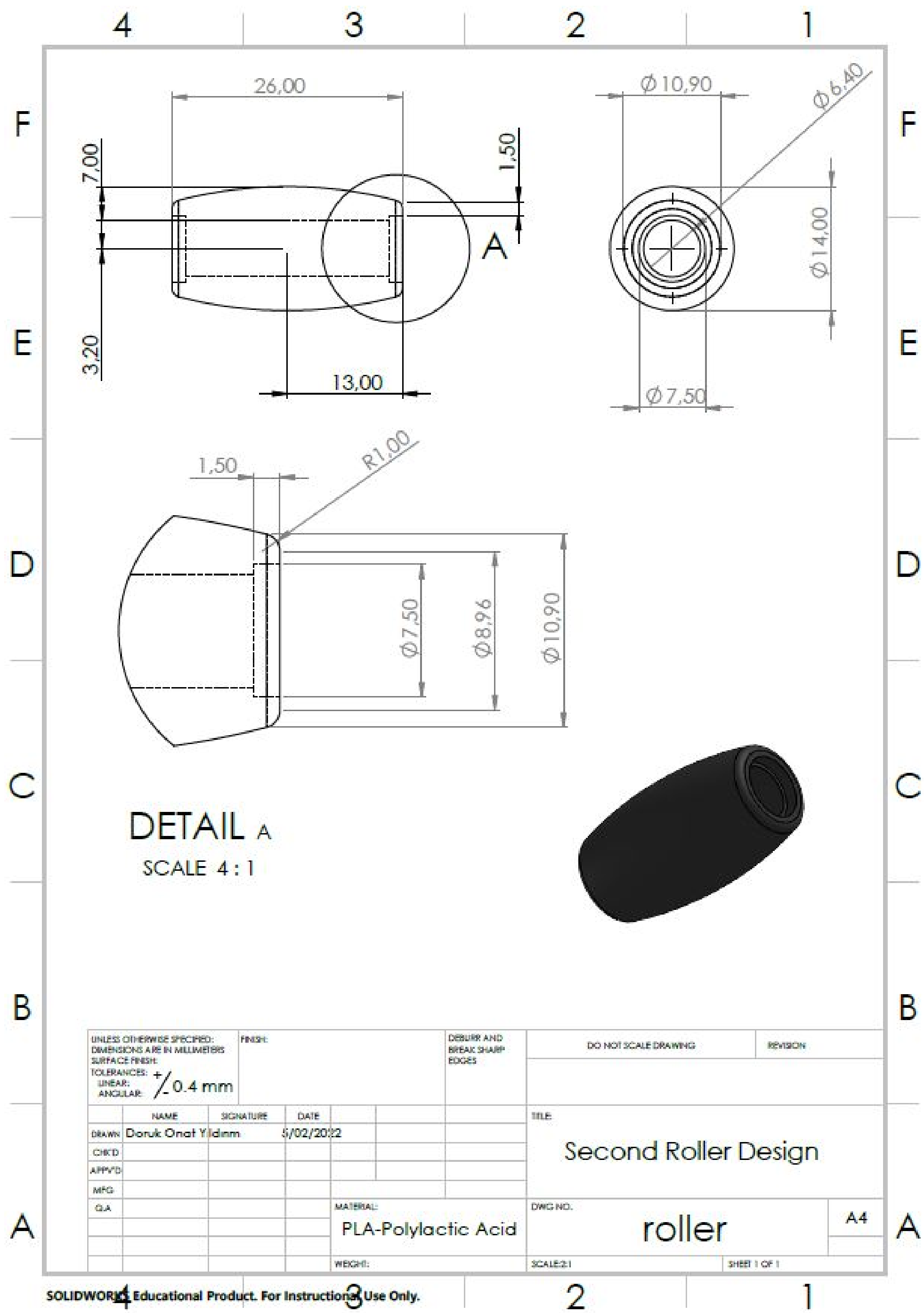


Figure 20: Second Roller Design

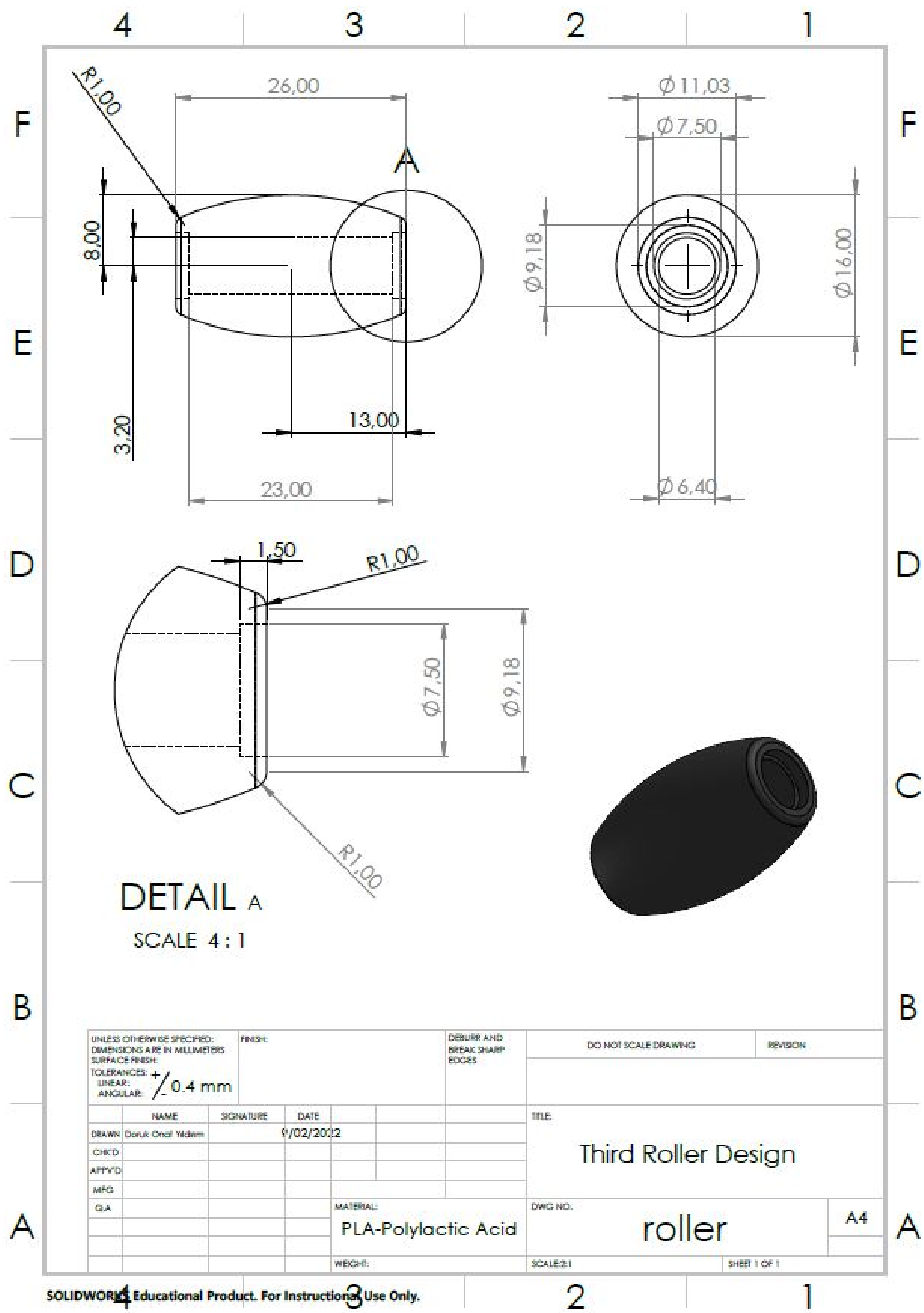


Figure 21: Third Roller Design

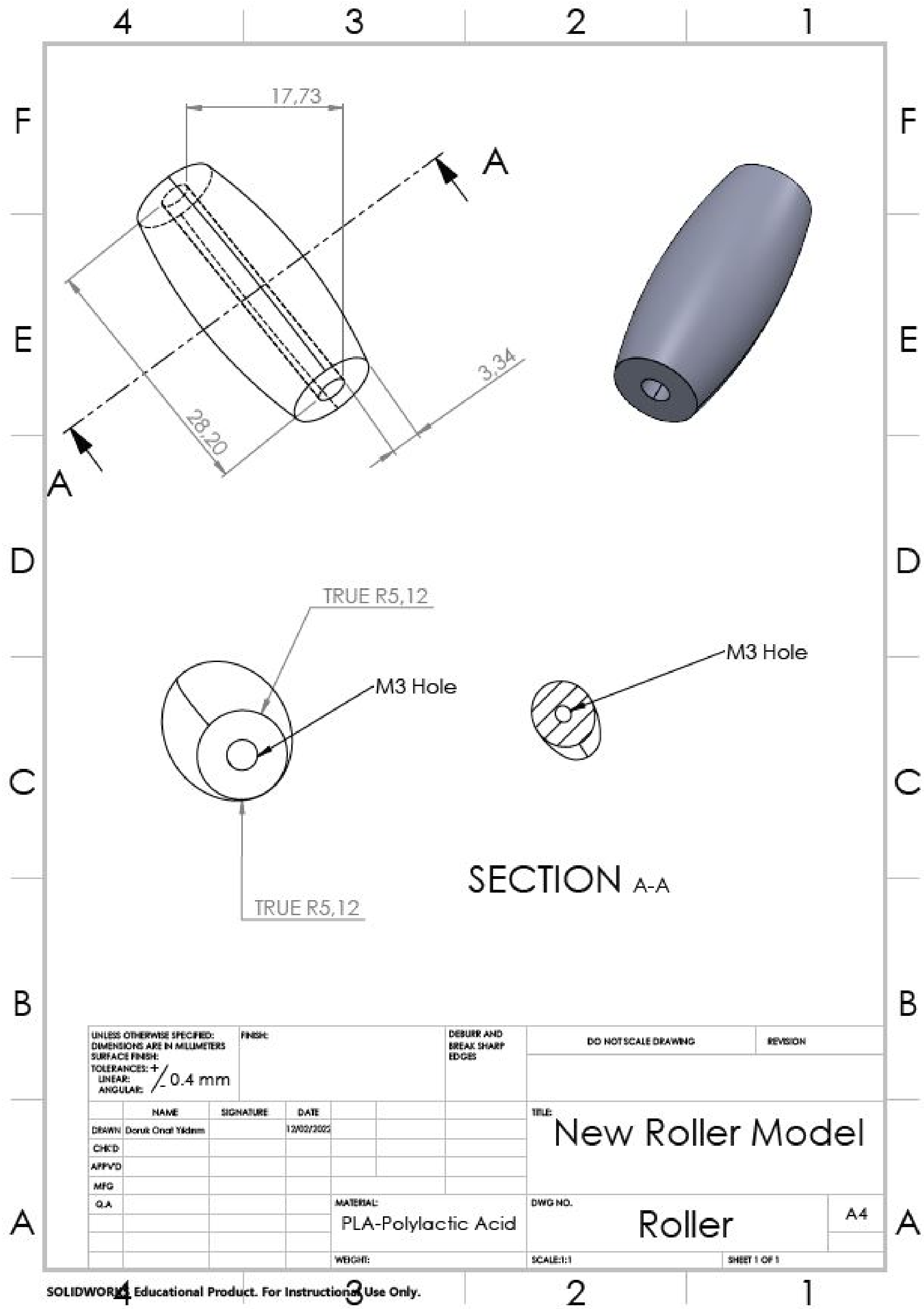


Figure 22: New Roller Design

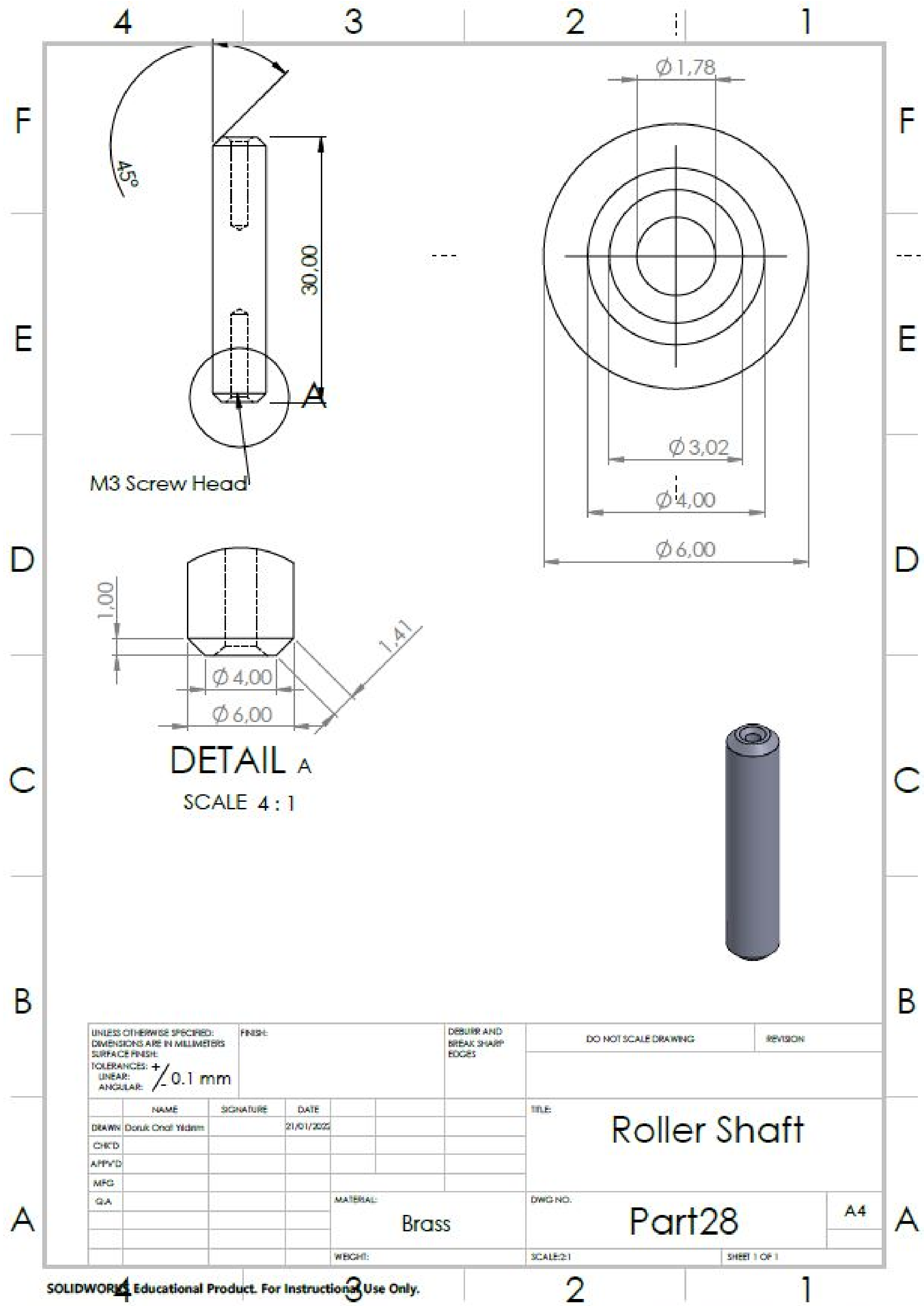


Figure 23: Roller Shaft

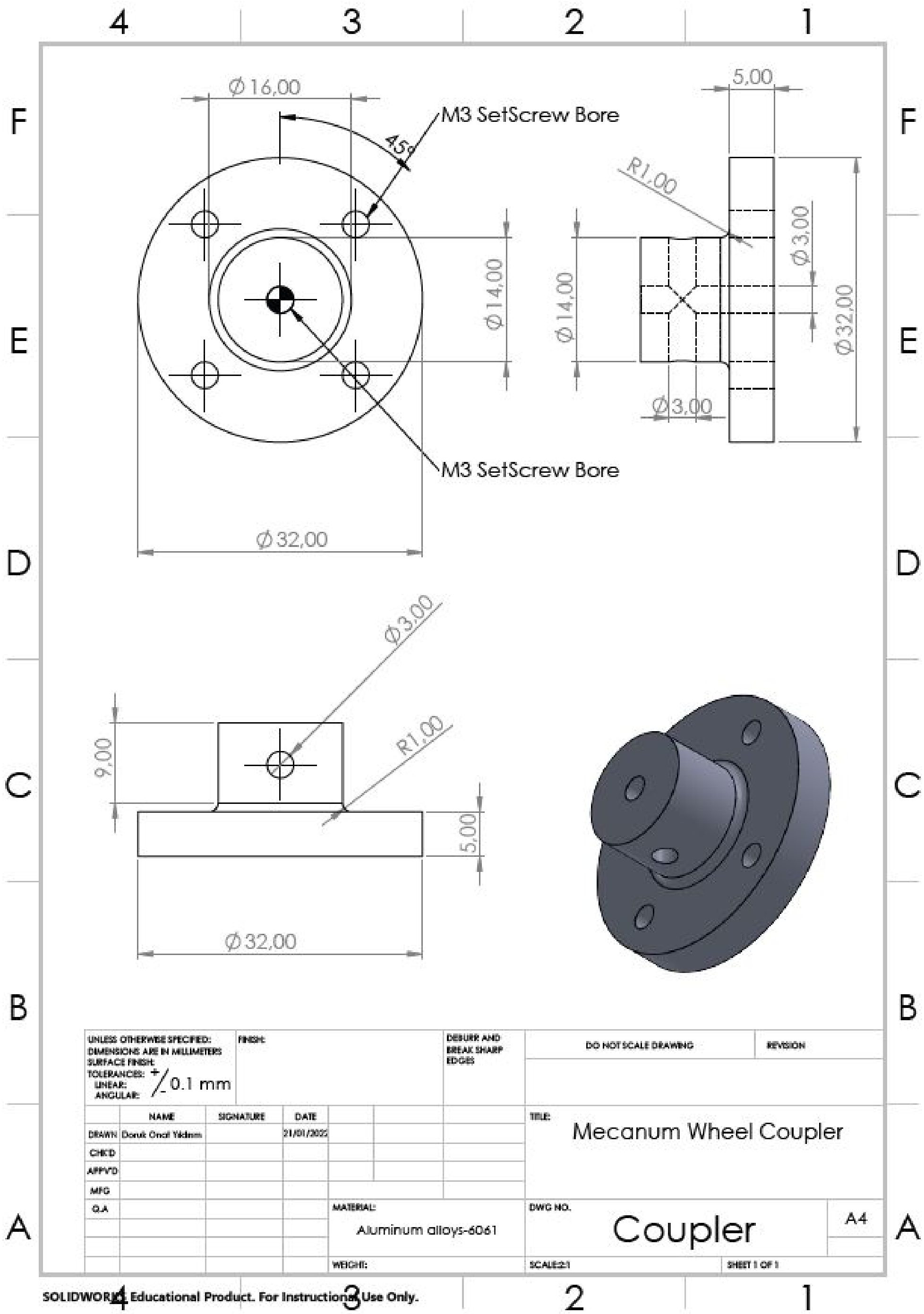


Figure 24: Mecanum Wheel Coupler Design

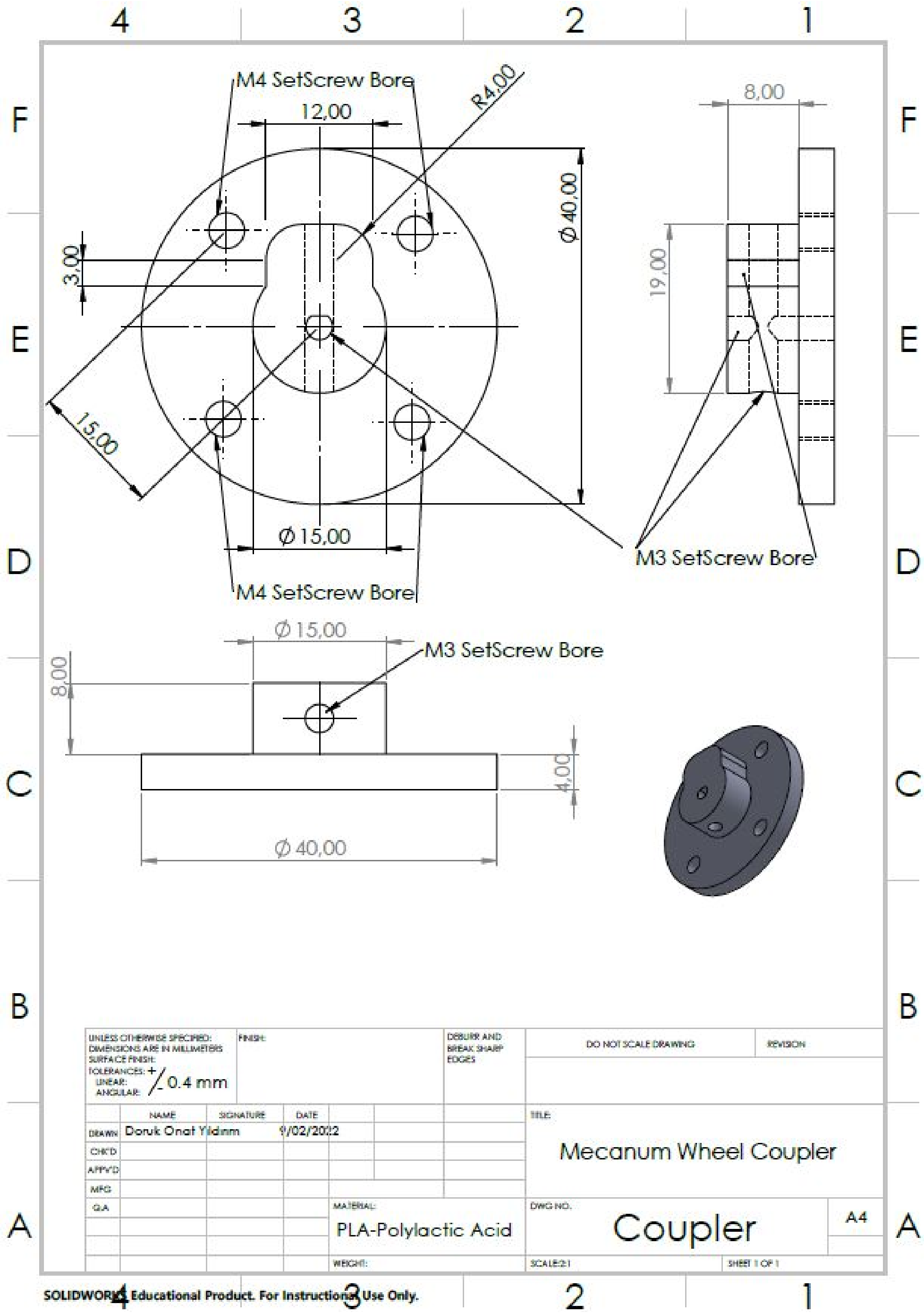


Figure 25: Mecanum Wheel Coupler Design

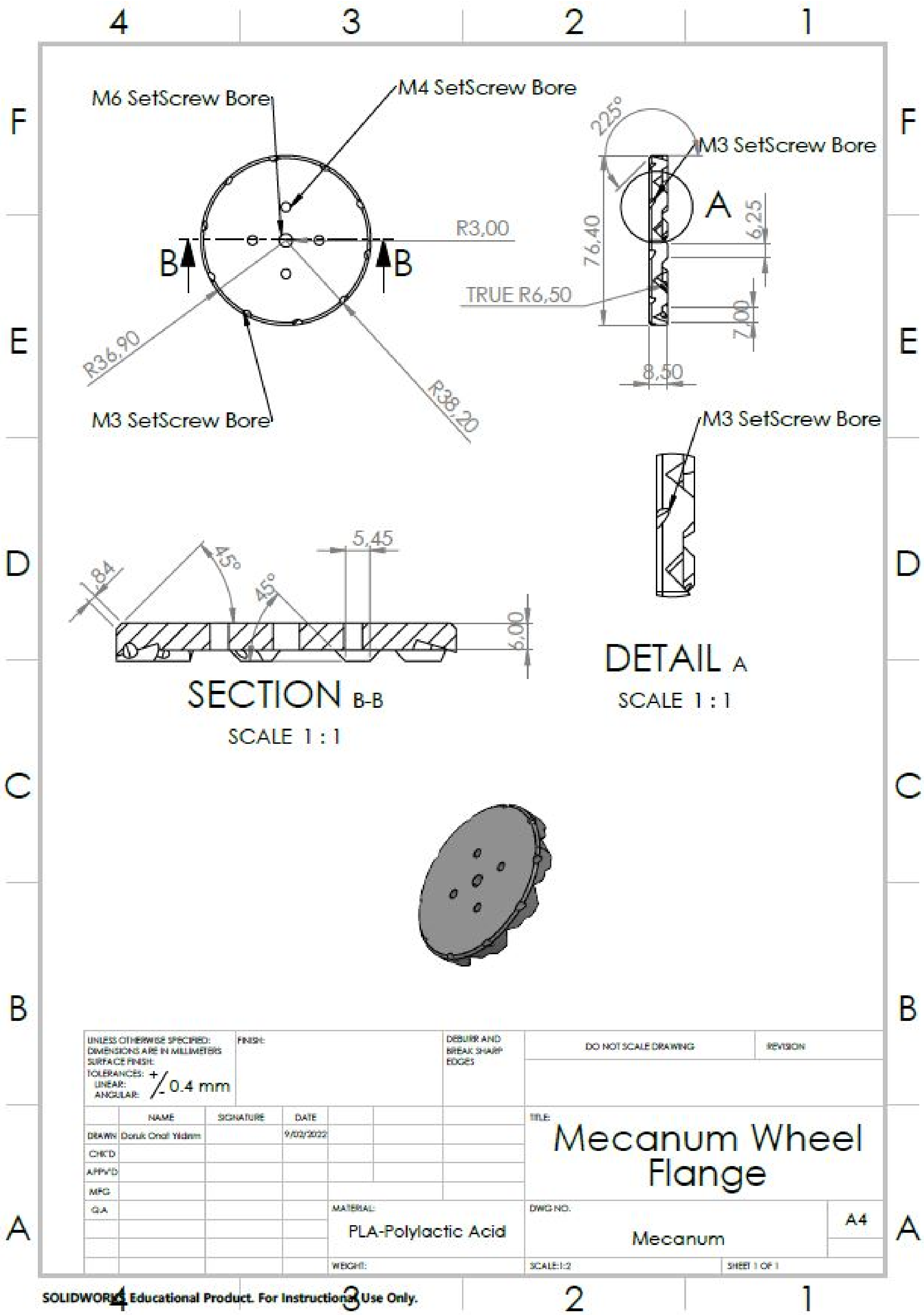


Figure 26: Mecanum Wheel Flange Design

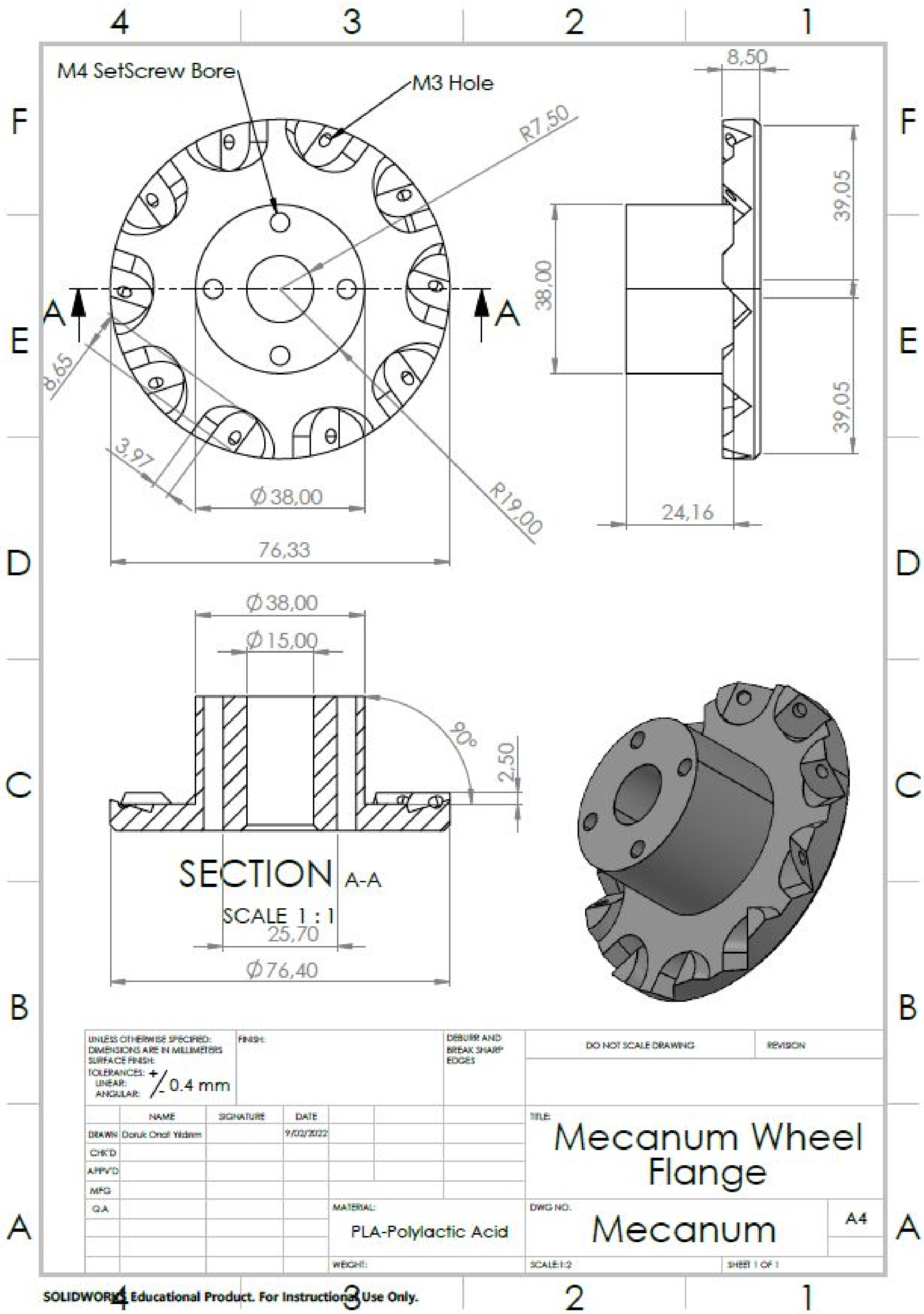


Figure 27: Mecanum Wheel Plate (Flange) Design

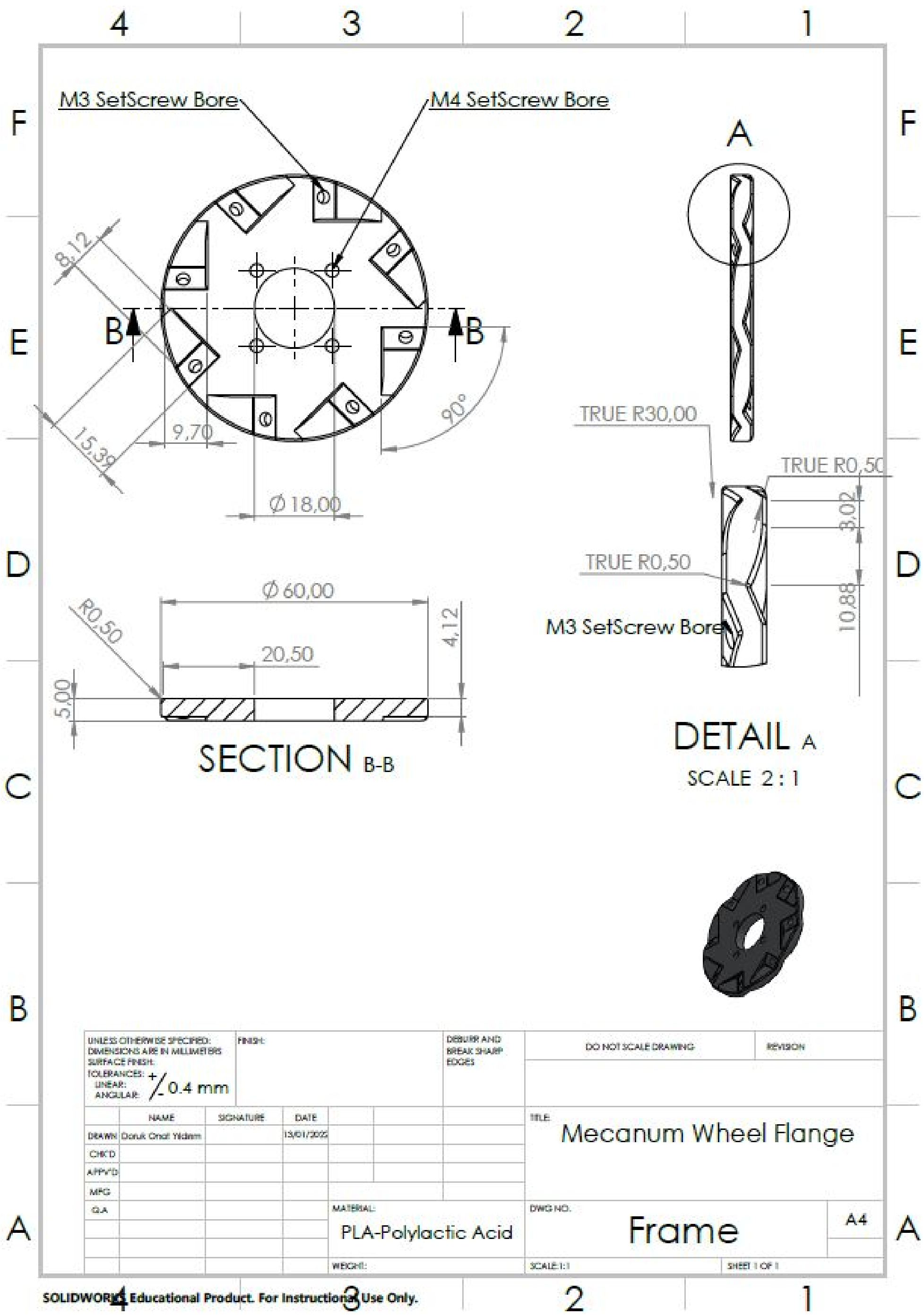


Figure 28: Mecanum Wheel Plate (Flange) Design

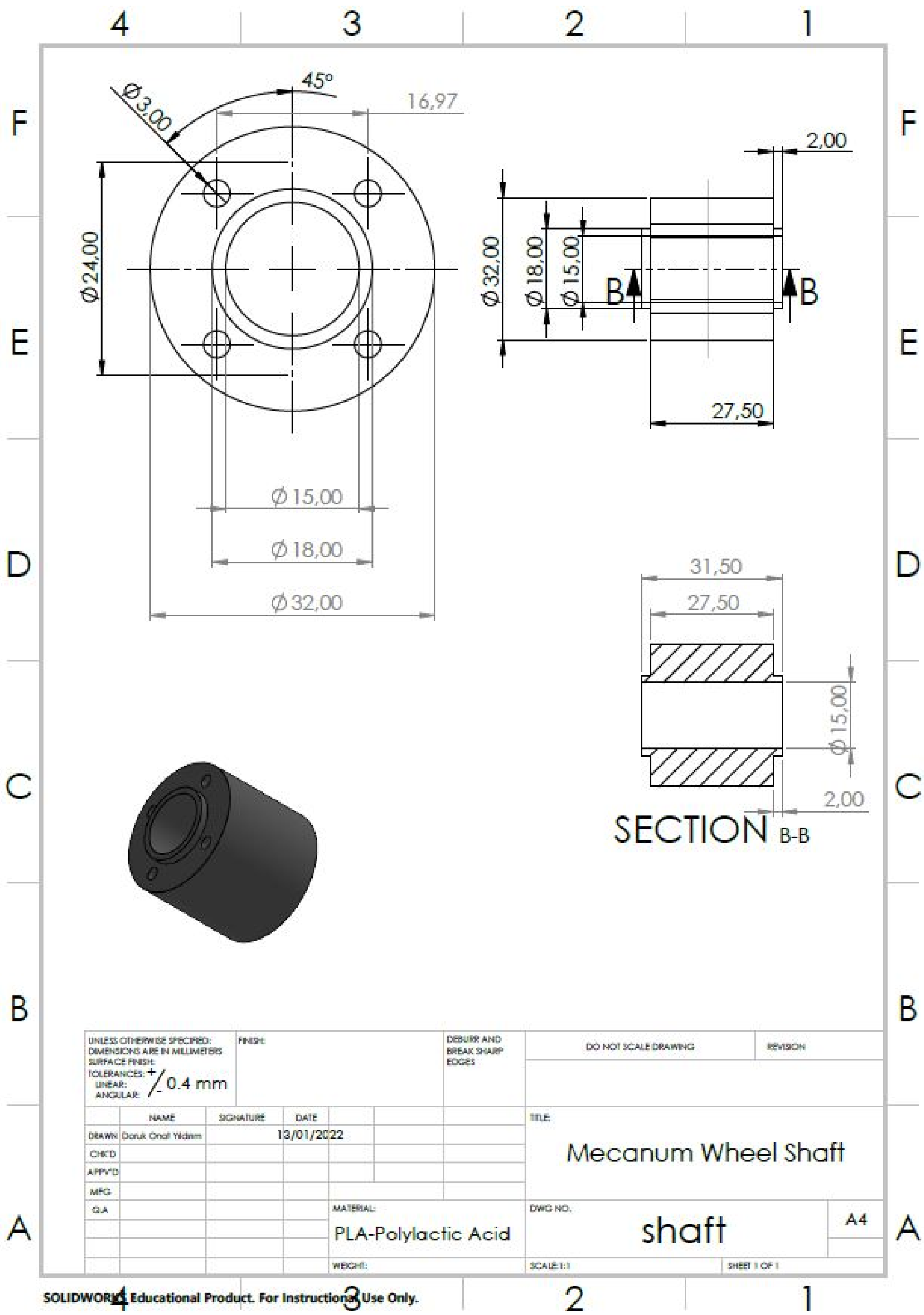


Figure 29: Mecanum Wheel Shaft Design

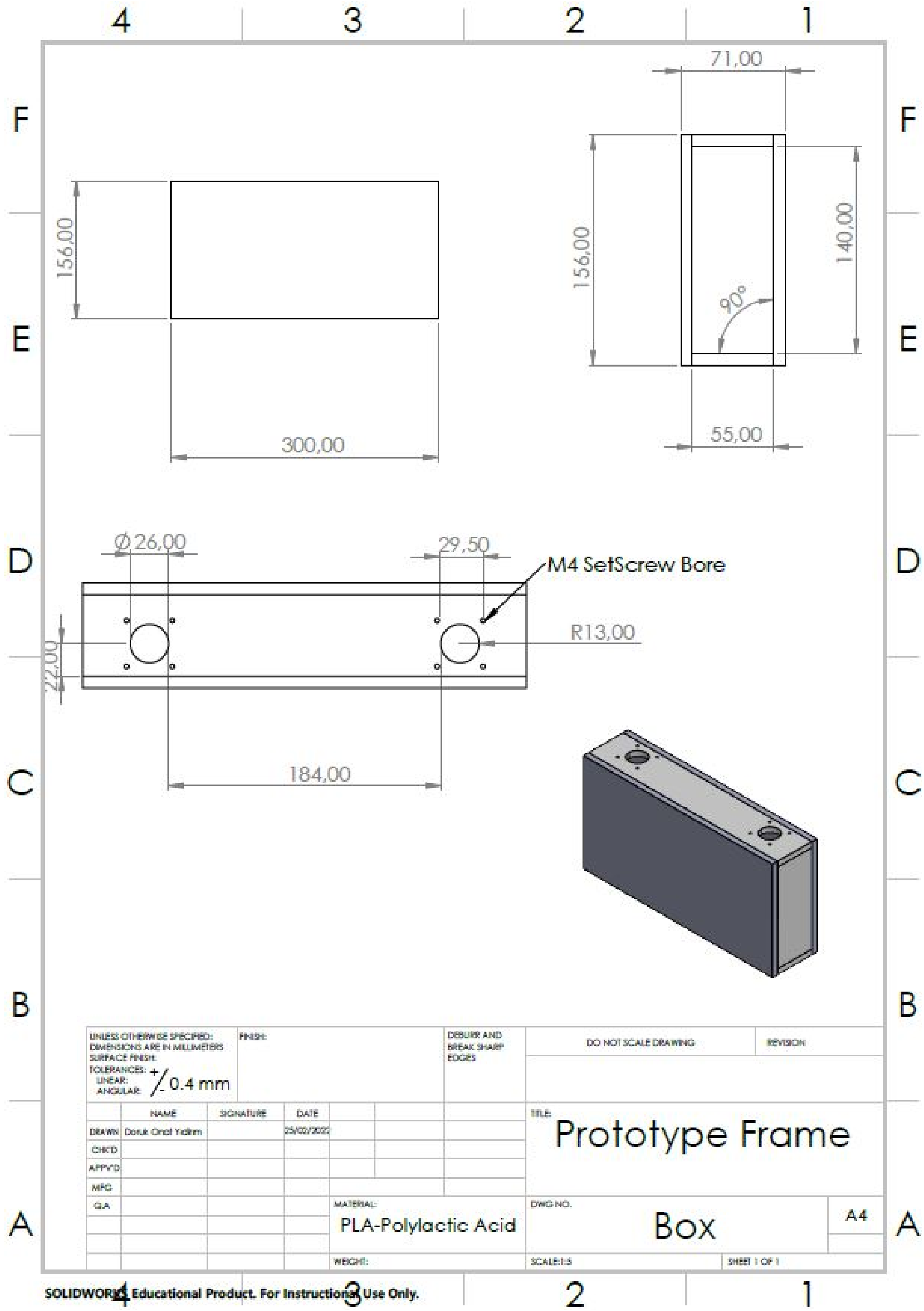


Figure 30: Prototype Frame

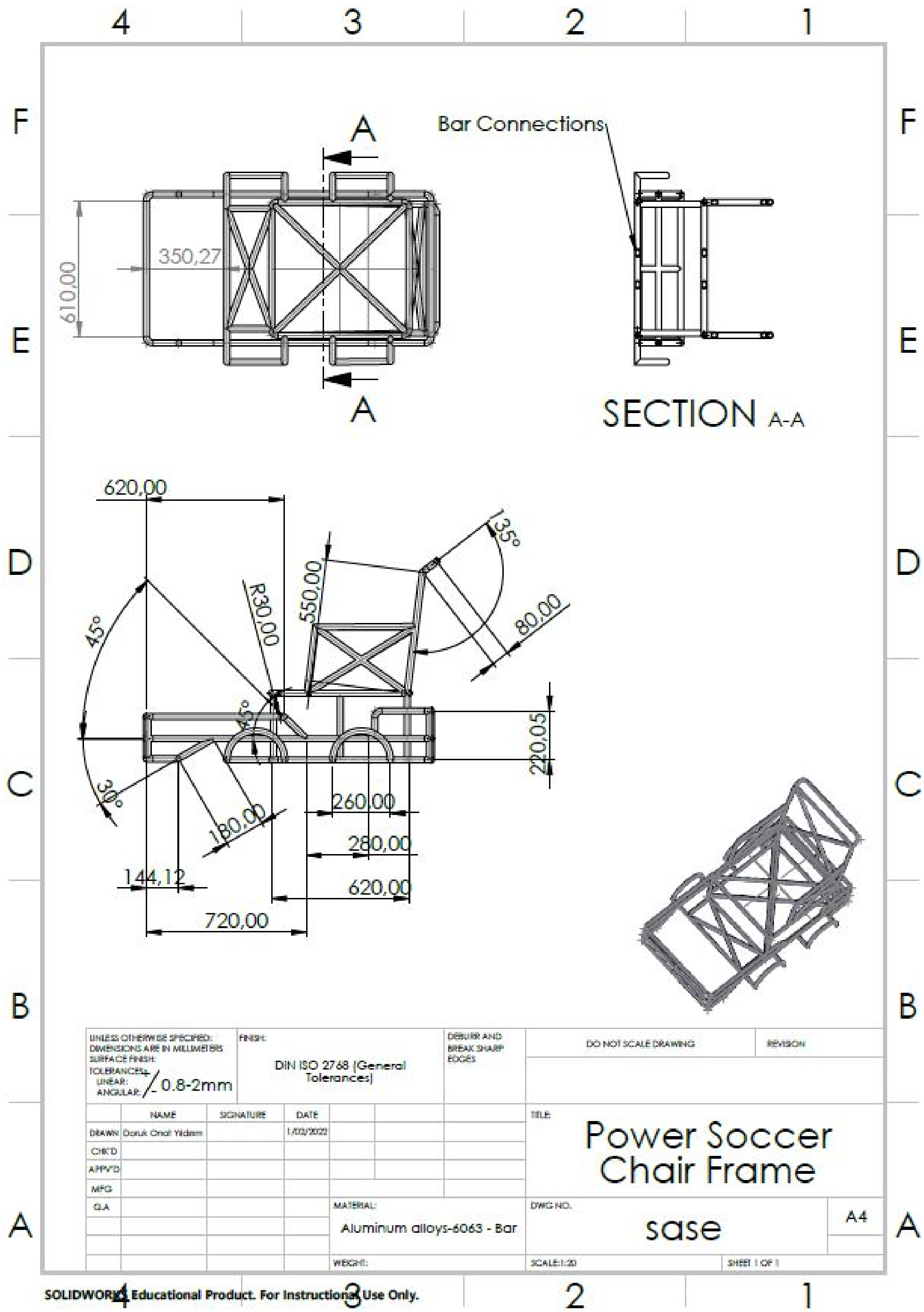


Figure 31: Power Soccer Chair Frame

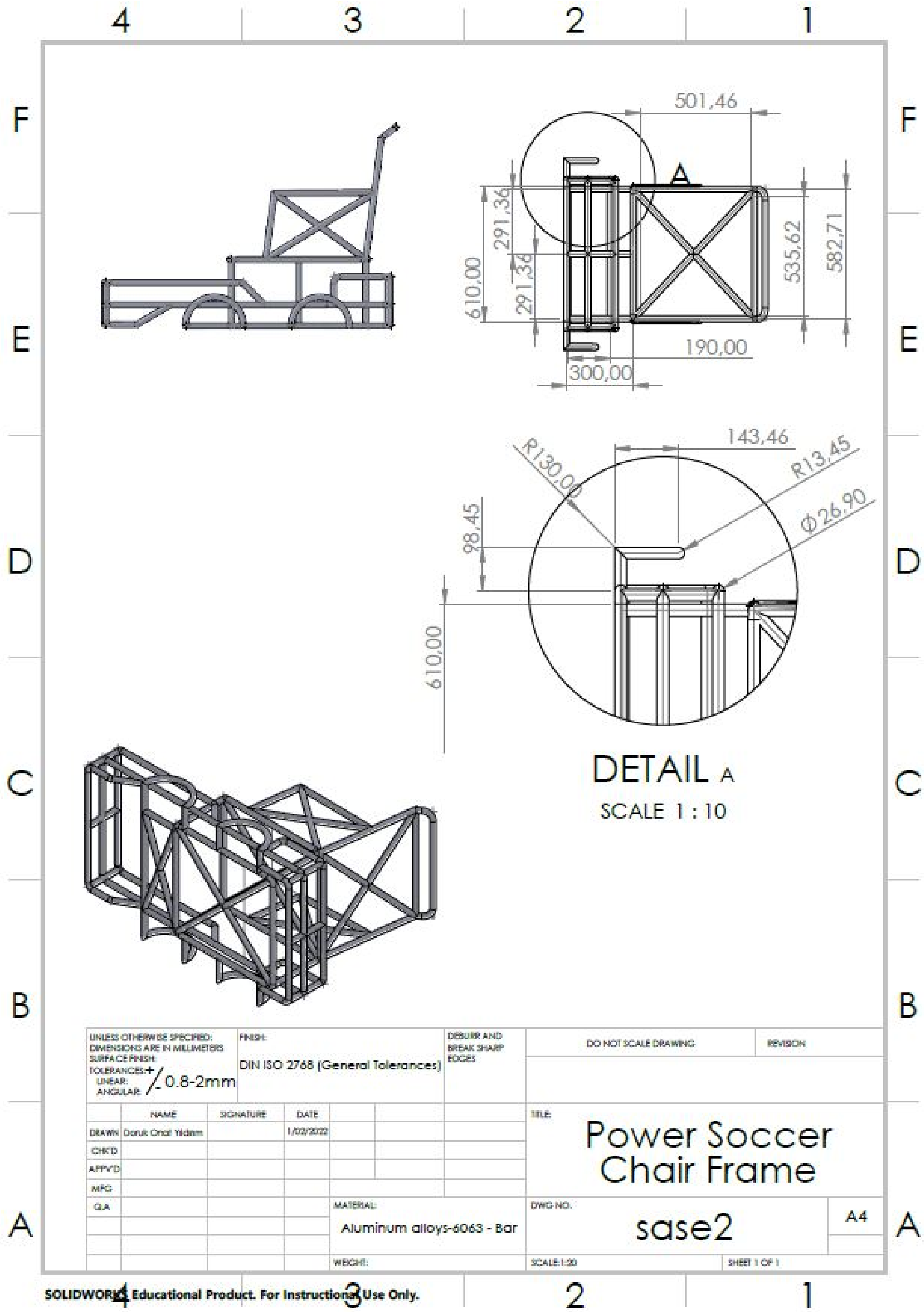


Figure 32: Power Soccer Chair Frame

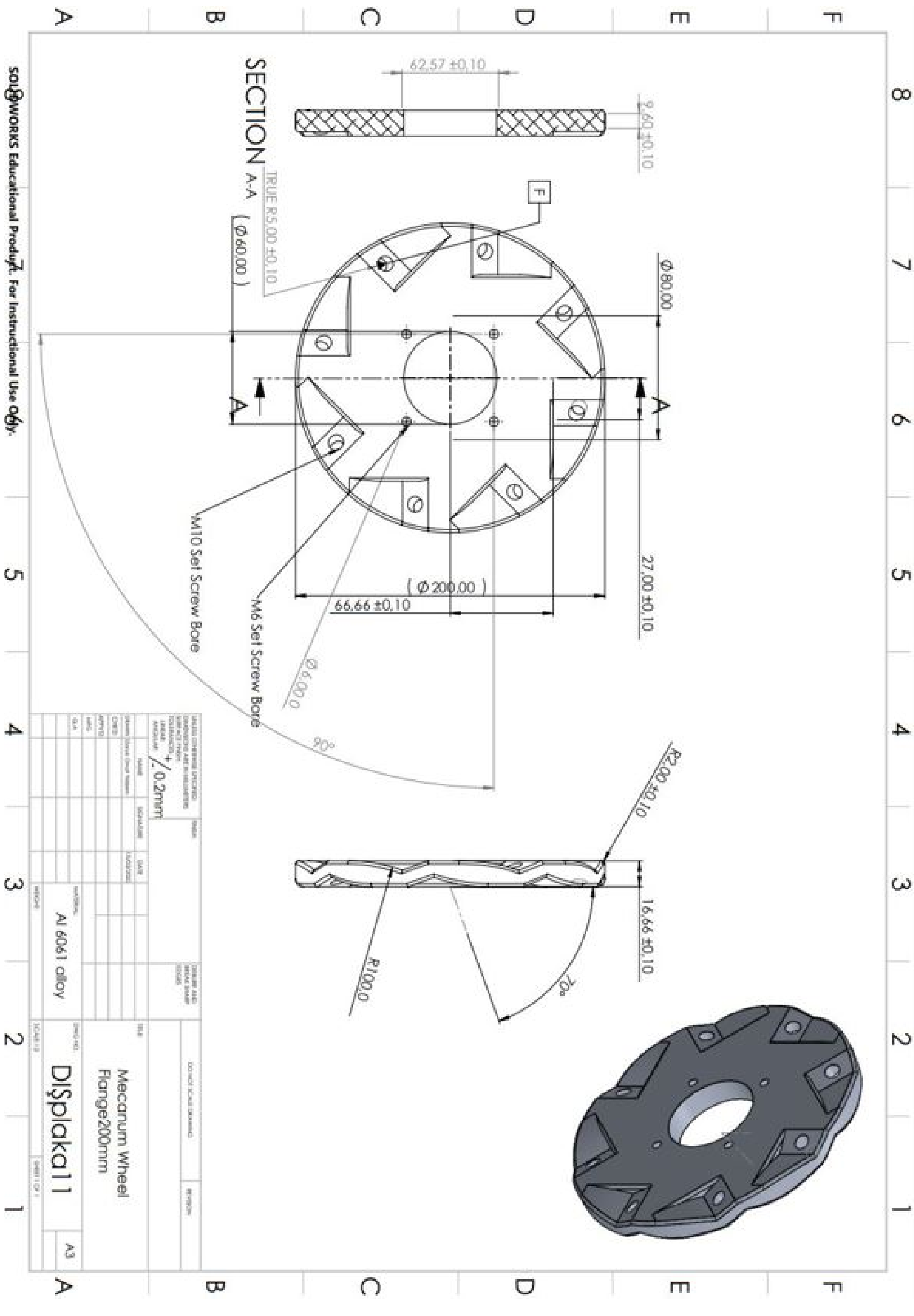


Figure 33: Mecanum Wheel Flange 200 mm

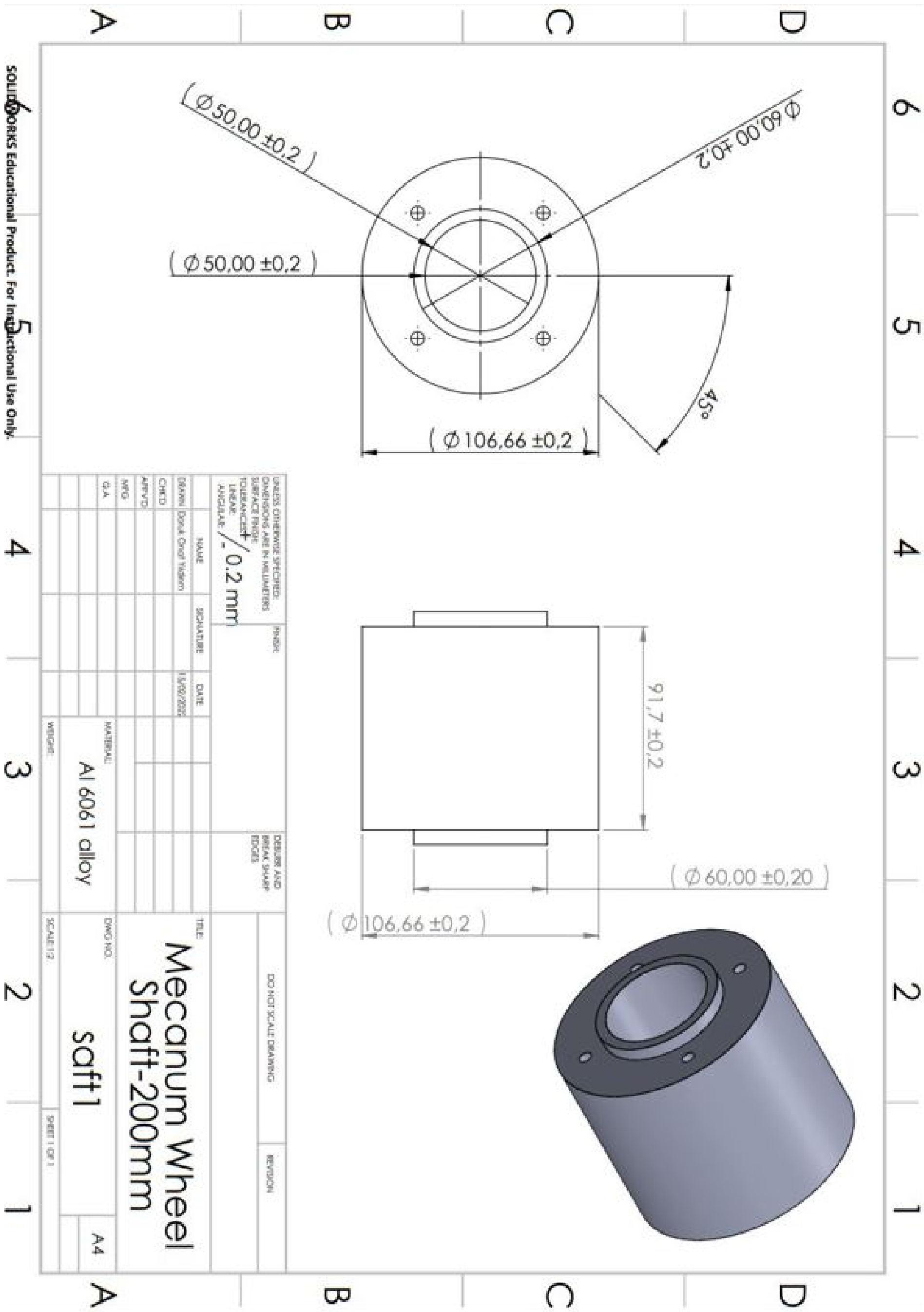


Figure 34: Mecanum Wheel Shaft 200 mm

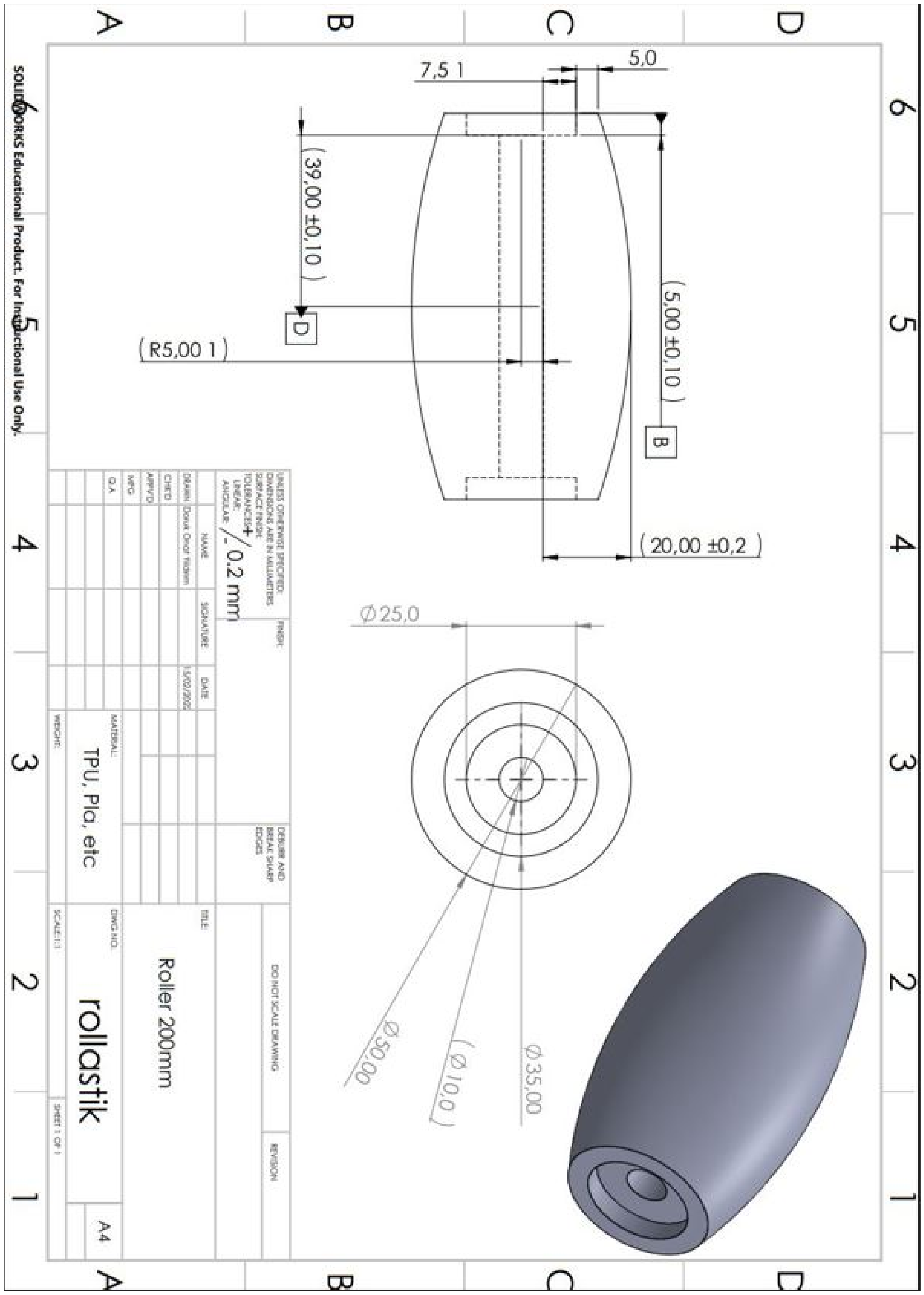


Figure 35: Mecanum Wheel Roller 200 mm

Appendix E- MATLAB Function for Kinematics

```
function computeMatrices(obj)
            % Creates the forward and inverse matrices given the object's
            % kinematic properties
            % Create inverse kinematics matrix (body speeds to wheel speeds)
            obj.invMatrix = [ 1, -1, -R; ...
                              1,  1,  R; ...
                              1,  1, -R; ...
                              1, -1,  R ] / obj.wheelRadius;
        end

classdef FourWheelMecanum < handle
    properties
        wheelRadius = 0.1;  % Wheel radius [m]
        wheelBase = 0.074;    % Wheelbase [m]
        wheelTrack = 0.14;   % Wheel track [m]
    end
    properties(Access=private)
        fwdMatrix = zeros(3,4); % Transformation matrix from wheel speeds to body speeds
        invMatrix = zeros(4,3); % Transformation matrix from body speeds to wheel speeds
    end

    methods

        function obj = FourWheelMecanum(wheelRadius,wheelBase,wheelTrack)
            % FOURWHEELMECANUM Construct an instance of this class

            % Unpack parameters
            obj.wheelRadius = wheelRadius;
            obj.wheelBase = wheelBase;
            obj.wheelTrack = wheelTrack;

```

```
            % Create forward and inverse matrices
            computeMatrices(obj);

        end

        function bodySpeeds = forwardKinematics(obj,wheelSpeeds)
            % Calculates linear and angular velocities [vx;vy;w],
            % in the *body* frame, from wheel speeds [w1;w2;w3;w4]
            bodySpeeds = obj.fwdMatrix * wheelSpeeds;
        end

        function wheelSpeeds = inverseKinematics(obj,bodySpeeds)
            % Calculates wheel speeds [w1;w2;w3;w4] from linear and
            % angular velocities [vx;vy;w] in the *body* frame
            wheelSpeeds = obj.invMatrix * bodySpeeds;
        end

        function M = getForwardMatrix(obj)
            % Returns forward kinematics matrix (wheel speeds to body speeds)
            M = obj.fwdMatrix;
        end

        function M = getInverseMatrix(obj)
            % Returns forward kinematics matrix (wheel speeds to body speeds)
            M = obj.invMatrix;
        end

        function computeMatrices(obj)
            % Creates the forward and inverse matrices given the object's
            % kinematic properties

            R = (obj.wheelBase + obj.wheelTrack)/2;
            % Create forward kinematics matrix (wheel speeds to body speeds)
```

```
            obj.fwdMatrix = [ 1, 1, 1, 1; ...
                             -1, 1, 1,-1; ...
                             -1/R, 1/R, -1/R, 1/R ] * (obj.wheelRadius/4);

            % Create inverse kinematics matrix (body speeds to wheel speeds)
            obj.invMatrix = [ 1, -1, -R; ...
                              1,  1,  R; ...
                              1,  1, -R; ...
                              1, -1,  R ] / obj.wheelRadius;
        end

    end
end
```

Appendix F- Arduino Code with Inverse Kinematic Operations

```
#define enA 11
#define in1 4
#define in2 5
#define enB 12
#define in3 6
#define in4 7



// define the same for 2 more DC motors
// TODO Change the output pins
#define enC 9
#define in5 0
#define in6 1
#define enD 10
#define in7 2
```

```
#define in8 3

int motorSpeedA = 0;
int motorSpeedB = 0;
// two more motor speeds
int motorSpeedC = 0;
int motorSpeedD = 0;

Matrix<4,3> inv_kinematic_matrix;

Matrix<4,1> motor_speeds; // index 0 maps to first motor speed

Matrix<3,1> joystick_input;

int wheelRadius = 0.1 // [m]
int R;
int wheelBase = 0.074;
int wheelTrack = 0.14;
R = (wheelBase + wheelTrack)/2;

inv_kinematic_matrix = {1, -1, -R, 1, 1, R, 1, 1, -R, 1, -1, R};

void setup() {
 pinMode(enA, OUTPUT);
 pinMode(enB, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(in3, OUTPUT);
 pinMode(in4, OUTPUT);
 pinMode(enC, OUTPUT);
 pinMode(enD, OUTPUT);
 pinMode(in5, OUTPUT);
 pinMode(in6, OUTPUT);
```

```
 pinMode(in7, OUTPUT);
 pinMode(in8, OUTPUT);
 Serial.begin(9600);
}

void loop() {
  // TODO change these to read ardu joystick
 int xAxis = analogRead(A0); // Read Joysticks X-axis
 int yAxis = analogRead(A1); // Read Joysticks Y-axis
 int rotation = analogRead(A2); // Read rotation input

 joystick_input = {xAxis, yAxis, rotation}

 //Serial.println(yAxis);
 // Y-axis used for forward and backward control
 // TODO map the numbers for ardu joystick
  y_velocity = map(yAxis, 0, 1024, -255, 255);
  x_velocity = map(xAxis, 0, 1024, -255, 255);
  w_velocity = map(rotation, 0, 1024, -255, 255);

 motor_speeds = inv_kinematic_matrix*joystick_input; // 4x1 matrix (vector)

 // index the matrix to get individual motor speeds
 motor_speed1 = motor_speeds(0,0);
 motor_speed2 = motor_speeds(1,0);
 motor_speed3 = motor_speeds(2,0);
 motor_speed4 = motor_speeds(3,0);

 if (motor_speed1 < -10) {
   // Set Motor A backward
   digitalWrite(in1, HIGH);
   digitalWrite(in2, LOW);
 }
```

```
else if (motor_speed1 > 10) {
 // Set Motor A forward
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
}
if (motor_speed2 < -10) {
  // Set Motor A backward
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
}
else if (motor_speed2 > 10) {
 // Set Motor B forward
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
}
if (motor_speed3 < -10) {
  // Set Motor C backward
  digitalWrite(in5, HIGH);
  digitalWrite(in6, LOW);
}
else if (motor_speed3 > 10) {
 // Set Motor C forward
  digitalWrite(in5, LOW);
  digitalWrite(in6, HIGH);
}

if (motor_speed4 < -10) {
  // Set Motor D backward
  digitalWrite(in7, HIGH);
  digitalWrite(in8, LOW);
}
else if (motor_speed4 > 10) {
 // Set Motor A forward
```

```
    digitalWrite(in7, LOW);
    digitalWrite(in8, HIGH);
 }
// After arranging direction get the abs value for pwm signal

 motor_speed1 = abs(motor_speed1);
 motor_speed2 = abs(motor_speed2);
 motor_speed3 = abs(motor_speed3);
 motor_speed4 = abs(motor_speed4);

 analogWrite(enA, motorSpeed1); // Send PWM signal to motor A
 analogWrite(enB, motorSpeed2); // Send PWM signal to motor B
 analogWrite(enC, motorSpeed3); // Send PWM signal to motor A
 analogWrite(enD, motorSpeed4); // Send PWM signal to motor B
}
```

Appendix G- Arduino Code for Initial Circuit Test

```

#define enA 11
#define in1 4
#define in2 5
#define enB 12
#define in3 6
#define in4 7

// define the same for 2 more DC motors

#define enC 9
#define in5 0
#define in6 1
#define enD 10
#define in7 2
```

```
#define in8 3

int motorSpeedA = 0;
int motorSpeedB = 0;
// two more motor speeds
int motorSpeedC = 0;
int motorSpeedD = 0;

void setup() {
 pinMode(enA, OUTPUT);
 pinMode(enB, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(in3, OUTPUT);
 pinMode(in4, OUTPUT);
 pinMode(enC, OUTPUT);
 pinMode(enD, OUTPUT);
 pinMode(in5, OUTPUT);
 pinMode(in6, OUTPUT);
 pinMode(in7, OUTPUT);
 pinMode(in8, OUTPUT);
 Serial.begin(9600);
}

void loop() {
  // TODO change these to read ardu joystick
 int xAxis = analogRead(A0); // Read Joysticks X-axis
 int yAxis = analogRead(A1); // Read Joysticks Y-axis
 //Serial.println(yAxis);
 // Y-axis used for forward and backward control
 // TODO map the numbers for ardu joystick
 if (yAxis < 450) {
   // Set Motor A backward
```

```
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // Set Motor B backward
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    // Set Motor C backward
    digitalWrite(in5, HIGH);
    digitalWrite(in6, LOW);
    // Set Motor D backward
    digitalWrite(in7, HIGH);
    digitalWrite(in8, LOW);

    motorSpeedA = map(yAxis, 450, 0, 0, 255);
    motorSpeedB = map(yAxis, 450, 0, 0, 255);
    motorSpeedC = map(yAxis, 450, 0, 0, 255);
    motorSpeedD = map(yAxis, 450, 0, 0, 255);
  }
  else if (yAxis > 500) {
    // Set Motor A forward
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    // Set Motor B forward
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    // Set Motor C forward
    digitalWrite(in5, LOW);
    digitalWrite(in6, HIGH);
    // Set Motor D forward
    digitalWrite(in7, LOW);
    digitalWrite(in8, HIGH);

    motorSpeedA = map(yAxis, 500, 1024, 0, 255);
    motorSpeedB = map(yAxis, 500, 1024, 0, 255);
```

```
    motorSpeedC = map(yAxis, 500, 1024, 0, 255);
    motorSpeedD = map(yAxis, 500, 1024, 0, 255);
  }
  // If joystick stays in middle the motors are not moving
  else {
    motorSpeedA = 0;
    motorSpeedB = 0;
    motorSpeedC = 0;
    motorSpeedD = 0;
  }

  // Prevent buzzing at low speeds
  if (motorSpeedA < 20) {
    motorSpeedA = 0;
  }
  if (motorSpeedB < 20) {
    motorSpeedB = 0;
  }
  analogWrite(enA, motorSpeedA); // Send PWM signal to motor A
  analogWrite(enB, motorSpeedB); // Send PWM signal to motor B
  analogWrite(enC, motorSpeedA); // Send PWM signal to motor A
  analogWrite(enD, motorSpeedB); // Send PWM signal to motor B
  Serial.println(motorSpeedA);
}
```

Appendix H- Arduino Code with Bluetooth Module for Prototype Testing

```

//Motor1: Left front
//Motor2: Right back
//Motor3: Left back
//Motor4: Right front
```

```

#include <AFMotor.h>
#include <SoftwareSerial.h>

SoftwareSerial bluetoothSerial(9, 10); // RX, TX

//initialize motors pin
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

char command;

void setup()
{
  bluetoothSerial.begin(9600);  //Set the baud rate to your Bluetooth module.
}

void loop() {
  while (bluetoothSerial.available() > 0) {
    command = bluetoothSerial.read();

    Stop(); //initialize with motors stoped

    switch (command) {
      case 'F':
        forward();
        break;
      case 'B':
        back();
        break;
      case 'L':
```

```
        left();
        break;
      case 'R':
        right();
        break;
      //Forward right
      case '1':
        forward_right();
        break;
      case '2':
        backward_right();
        break;
      case '3':
        forward_left();
        break;
      case '4':
        backward_left();
        break;
      case '5':
        rotate_right();
        break;
      case '6':
        rotate_left();
        break;
      case '7':
        Stop();
        break;
    }
  }
}

void forward()
{
```

```
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD);  //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD);  //rotate the motor clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(FORWARD);  //rotate the motor clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(FORWARD);  //rotate the motor clockwise
}

void back()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD); //rotate the motor anti-clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD); //rotate the motor anti-clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void left()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD); //rotate the motor anti-clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD); //rotate the motor anti-clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(FORWARD);  //rotate the motor clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(FORWARD);  //rotate the motor clockwise
}
```

```
void right()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD);  //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD);  //rotate the motor clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}
//Motor1: Left front
//Motor2: Right back
//Motor3: Left back
//Motor4: Right front
void forward_right()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD);  //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD);  //rotate the motor clockwise
  motor3.setSpeed(0); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(0); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}
void backward_right()
{
  motor1.setSpeed(0); //Define maximum velocity
  motor1.run(FORWARD);  //rotate the motor clockwise
  motor2.setSpeed(0); //Define maximum velocity
  motor2.run(FORWARD);  //rotate the motor clockwise
```

```
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}
void forward_left()
{
  motor1.setSpeed(0); //Define maximum velocity
  motor1.run(FORWARD);  //rotate the motor clockwise
  motor2.setSpeed(0); //Define maximum velocity
  motor2.run(FORWARD);  //rotate the motor clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(FORWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(FORWARD); //rotate the motor anti-clockwise
}
void backward_left()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD);  //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD);  //rotate the motor clockwise
  motor3.setSpeed(0); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(0); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}
void rotate_right()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD);  //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD);  //rotate the motor clockwise
```

```
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(FORWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}
void rotate_left()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD);  //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD);  //rotate the motor clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(FORWARD); //rotate the motor anti-clockwise
}

void Stop()
{
  motor1.setSpeed(0);  //Define minimum velocity
  motor1.run(RELEASE); //stop the motor when release the button
  motor2.setSpeed(0);  //Define minimum velocity
  motor2.run(RELEASE); //rotate the motor clockwise
  motor3.setSpeed(0);  //Define minimum velocity
  motor3.run(RELEASE); //stop the motor when release the button
  motor4.setSpeed(0);  //Define minimum velocity
  motor4.run(RELEASE); //stop the motor when release the button
}
```

Appendix I- Analysis of the platform

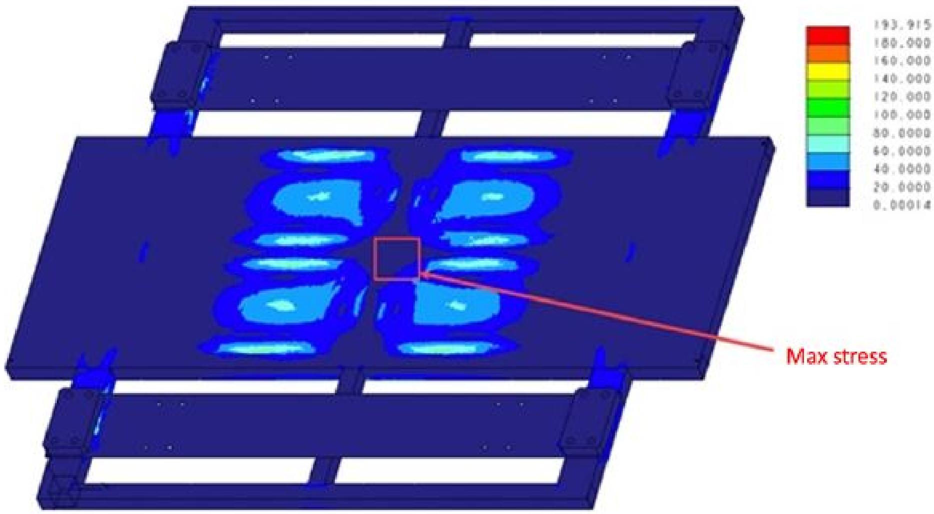


Figure 36: Analysis of the chassis platform

Appendix H- Motor Chassis Connection

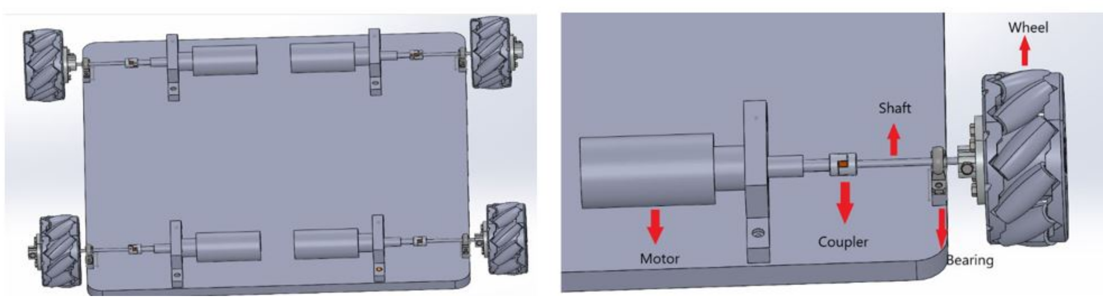


Figure 37: Motor Chassis Connnection