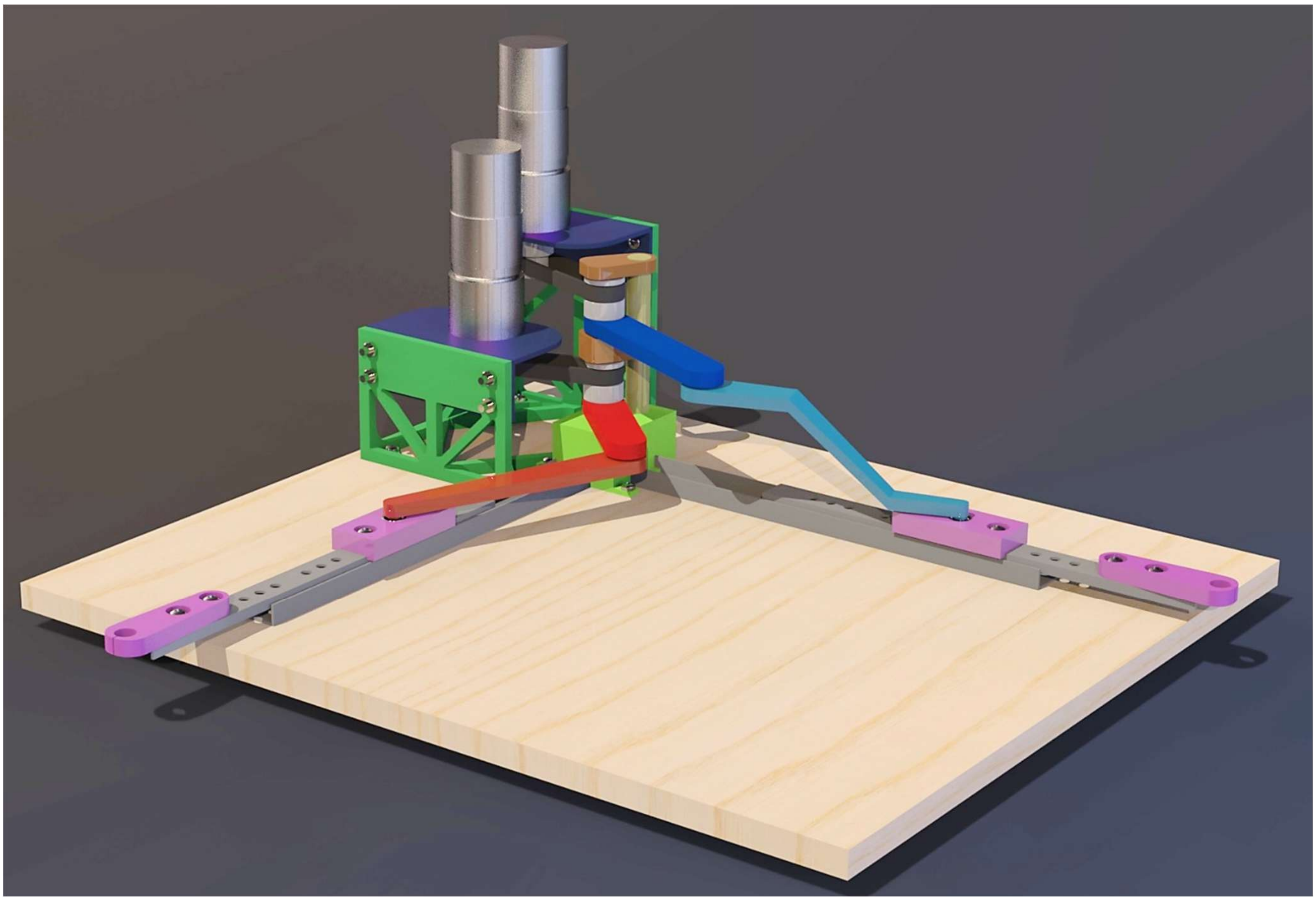


## Introduction

The device consists of a dual three-link manipulator with 2 actuators and 2 end-effectors on perpendicular axes.

The design process starts with the study of the 2D model of the robot and the evaluation of its kinematics. The workspace of the robot is then explored and, after considering several concepts, the most suitable design is assessed and explained.

The programming is executed using MATLAB Simulink, also for the construction of the kinematics block diagrams.

First a closed-loop system is run to test the application of PID controller for running the DC motor, then the robot kinematics functions are built and implemented in the control system.

## Tools and Components

The motors used are 2 simple EMG30 12 volts brushed motors, equipped with encoders and a gearbox reduction of 30:1, the encoders are used for the feedback signal in the control system.

The motors are driven by a RKL298 PCB, it can power up to two motors with the use of an external DC power supply and consists of an H-bridge drive chip that adjusts the rotation of each motor by controlling the polarity of the voltage applied to it, the sense of rotation depends on the combination of two inputs pins that come from the microcontroller. The speed of the motor is also regulated by the microcontroller sending Pulse Width Modulation -PWM- pulses to the motor driver.

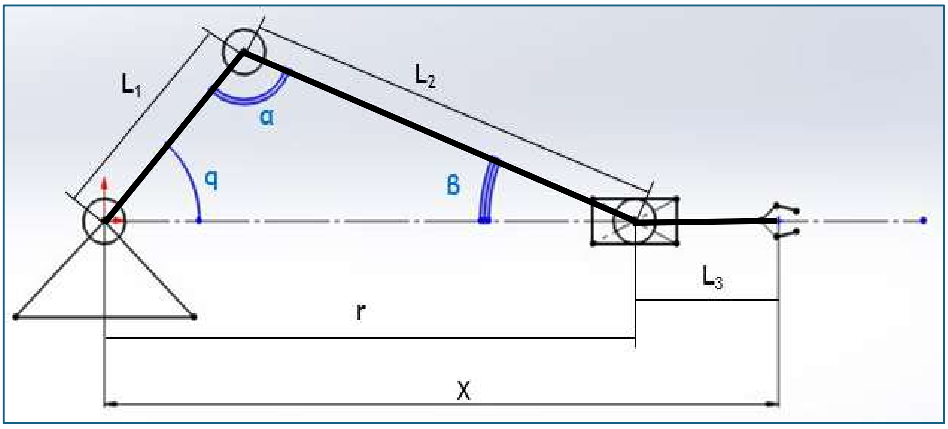
The microcontroller stated is the Launchpad F28069M board by Texas Instruments interfaced with MATLAB, the code is generated by Code Composer Studio on the basis of a block diagram model built on Simulink.

## Robot Kinematics

Due to the structure of the robot, the kinematics of the two manipulators is the same but on different axis and angles. Note that the shape of the robot leads to establish some limits to the movement of the actuators.

Each manipulator consists of 3 links and 3 joints, the first joint is attached to the actuator on the base, while the last one is fixed on a linear slider. The third link with the end-effector move along a defined axis and therefore the rotational movement of the actuator is converted into linear displacement. The two manipulators have a similar shape but with links of different lengths, that affect their workspace and movement.

### Set 1)




The first manipulator lies on the horizontal axis. The actuator is placed on the base, and it is considered as the origin of the x-axis on which the end-effector moves.

*Schematics for kinematics of first manipulator*

## Forward Kinematics

Forward kinematics consists in the derivation of the total displacement X for any given angle of q. The lengths of the links are predefined while the angles change, of which only one is known.

The total displacement X is considered as the sum of $L_3$ and the hypotenuse r of the triangle.

$$1.\ \ X = r + L_3$$

Two sides and one angle of the triangle are known; hence the third side r can be found using a few trigonometric functions:

From sine rule:

$$2.\ \ \frac{L_1}{sin\beta} = \frac{L_2}{sinq} = \frac{r}{sin\alpha}$$

To find r, angle α is needed, and it is derived from the subtraction of the triangle's angles to the total of 180°.

First, angle β is calculated from sine

$$3.\ \ \sin\beta = \frac{L_1\ \sin q}{L_2}$$

and

$$4.\ \ \beta = \sin^{-1}(\frac{L_1}{L_2} sinq)$$

Now the third angle is derived

$$5.\ \ \alpha = 180\ - q - \beta$$

And r can be calculated as

$$6.1.\quad r = \frac{L_2\, sin\alpha}{sinq} \qquad \text{or} \qquad 6.2.\ r = \frac{L_1\, sin\alpha}{sin\beta}$$

Thus, the resultant displacement is equation 1. with the r found in either 6.1 or 6.2.

## Inverse Kinematics

Inverse kinematics is the derivation of the angle of the actuator q needed to achieve a desired displacement in X.

As for forward kinematics, the total displacement X is the sum of $L_3$ and r, so for the solution of q, r is derived from X

$$7.\ \ r = X - L_3$$

Cosine rule can be applied to the triangle as in equation 8.

$$8.\ \ {L_2}^2 = r^2 + {L_1}^2\ - 2rL_1 \cos q$$

From this equation, the angle q can be simply derived:

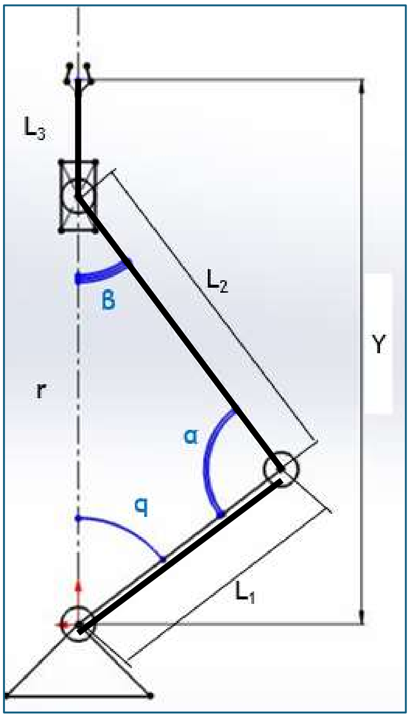
$$9.\ \ cosq = \frac{r^2 + {L_1}^2 - {L_2}^2}{2rL_1}$$

and

$$10.\ \ q = \cos^{-1}(\frac{r^2 + {L_1}^2 - {L_2}^2}{2rL_1})$$

For inverse kinematics, when coming to the resolution of the final angle, always the smallest value is taken, e.g. if $\cos^{-1}({}^1\!/_2) = 60°\ or\ 300°$ only 60 is considered. The angle of the first actuator is limited from 0 to 90°.

## Set 2)




*Schematics for kinematics of second manipulator*

The second manipulator lies on the vertical axis. The actuator is placed on the base, aligned with the one of set 1, and it is considered as the origin of the y-axis on which the end-effector moves.

### Forward Kinematics

Forward kinematics is the computation of the total displacement Y for a given angle q.

The total displacement Y is defined as

$$11. Y = r + L_3$$

Equations from 2 to 6 apply in the same way, but with the respective lengths and angle q, and the resulting r is used in equation 11.

### Inverse Kinematics

Inverse kinematics is the derivation of the angle of the actuator q needed to achieve the desired vertical displacement in Y.

$$12. \quad r = Y - L_3$$

Equations 8 to 10 are used in the same way, but with the respective lengths applied, and the resulting angle is always the smallest between y-axis and the first link $L_1$.

Note that the value of second actuator is considered as negative and is limited from 0 to 60°.

### Workspace Analysis

The workspace derived from the robot kinematics is quite simple: the two end-effectors move along two perpendicular axes x and y. The robot moves on a 2D plane and has a total of 2 degrees of freedom.

Since, for building reason, limits are set to the actuators' angle, the workspace only depends on the lengths of the links.

The workspace on the 2 axes goes from the minimum extension of

$$13. \; X_{min} = \sqrt{{L_2}^2 - {L_1}^2} + L_3 \quad for \quad q_1 = 90°$$

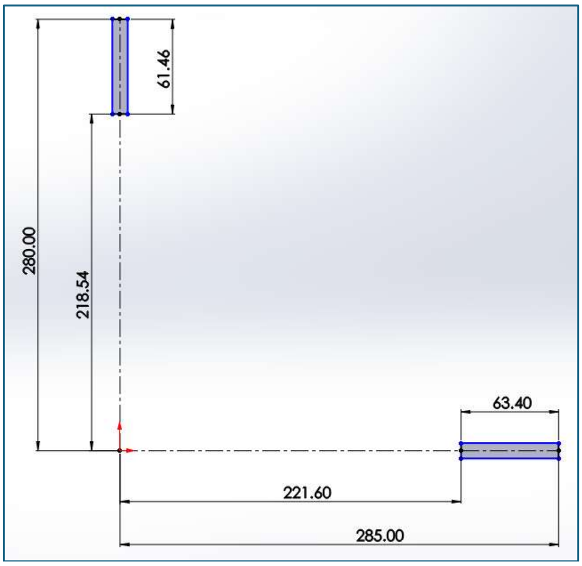
$$14. \; Y_{min} = \frac{L_2 \, sin\alpha}{sinq} + L_3 \quad for \quad q_2 = 60°$$

To the maximum of

$$15. \; L_1 + L_2 + L_3 \quad for \quad q_1 = q_2 = 0°$$

For the prototype design the following lengths are used, in millimetres (mm):

| SET | LINK 1 | LINK 2 | LINK 3 |
|---|---|---|---|
| 1 | 50 | 100 | 135 |
| 2 | 75 | 100 | 105 |



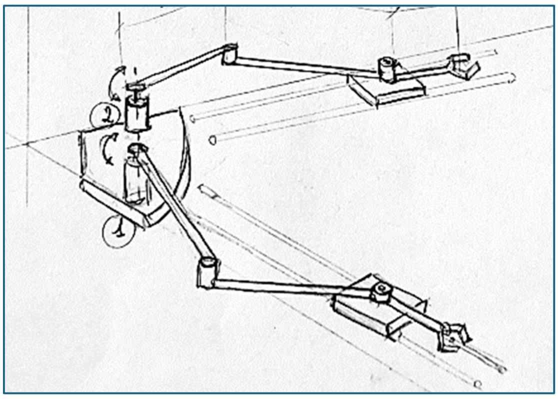

*Workspace drawing with measures*

The workspace measured seems small and very limited, for the two manipulators it appears to be just above 60 mm.

This happens due to the concept and the application of the robot, the actuators cannot rotate fully otherwise there would be conflicts between the different parts that would interfere with each other, preventing the free movement of the manipulators.

## General Design and Concept

*Robot general concept sketch*

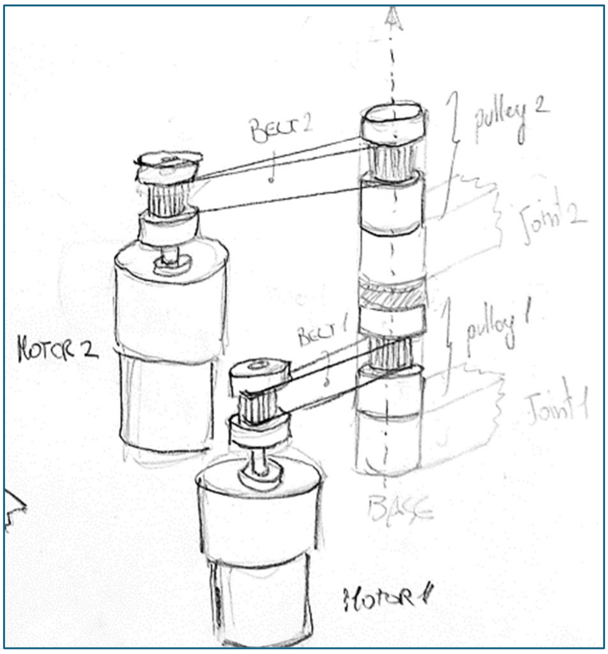


The robot lies on a 2D plane and has 2 degrees of freedom which correspond to the linear movement of the end-effectors on the two axes of the plane. The first joints of the two manipulators are those connected to the actuators, they are placed on the base, and they are aligned to the same vertical z-axis, going out of the workspace plane.

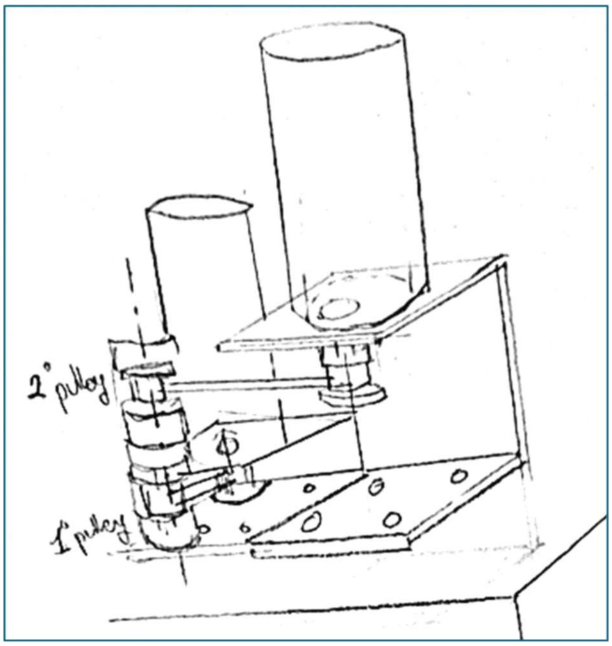
The end joints and the two end-effectors are fixed on linear slides; hence the third link is made up of a carriage, the slide and the end effector. The rotational movement of the joints is allowed by the use of small ball bearings.

For each actuator, the transmission of angular motion can be ensured by either different gear systems or with pulleys and belt between the motor and the first joint. For the purpose of this robot, the belt and pulley system is chosen as easy to make and accurate. Small and precise gear systems are difficult to print unless a high precision printer is used. Therefore, the two motors can be positioned away from the base but at a defined height that ensures the correct coupling of the belt on the pulleys.

For this concept, the two motors are fixed on the respective mounting brackets that are installed on vertical supports that holds the motors in position, at the correct height, and upside-down.




sketch of the pulley and belt system for motion transmission

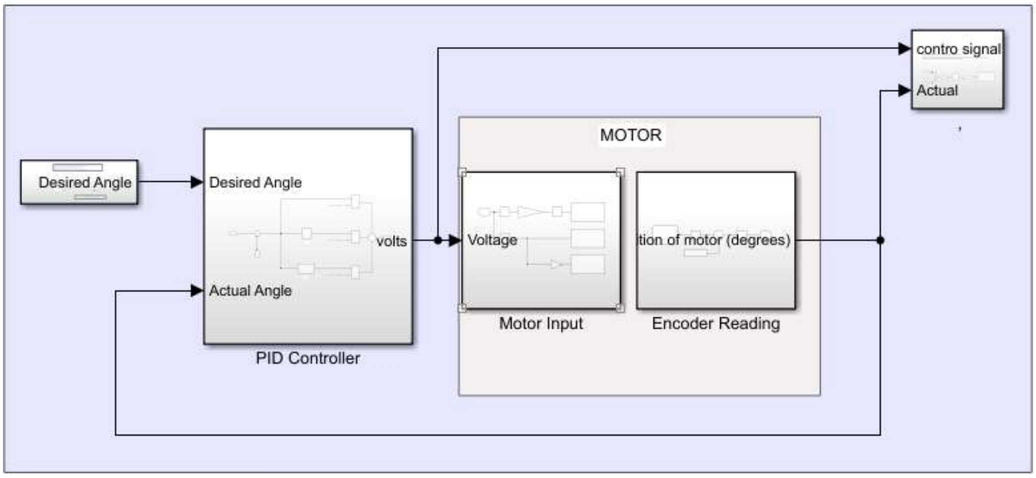



sketch of the supports for the motors

## MATLAB Control Simulation

The program consists of a Simulink model which includes a closed-loop system with PID controller. The controller that takes the angular position of the motor from the encoders, compares it with a given input value and finally delivers a suitable voltage to achieve the exact position. The values of both the reference and the angular position are read in degrees, while the output signal from PID controller is in volts.

The control system design starts with the control test of one DC motor driven by a PID controller implemented within the program. The robot kinematics is then introduced, the respective block diagrams are assembled on Simulink, and subsequently implemented within the motor control diagram.

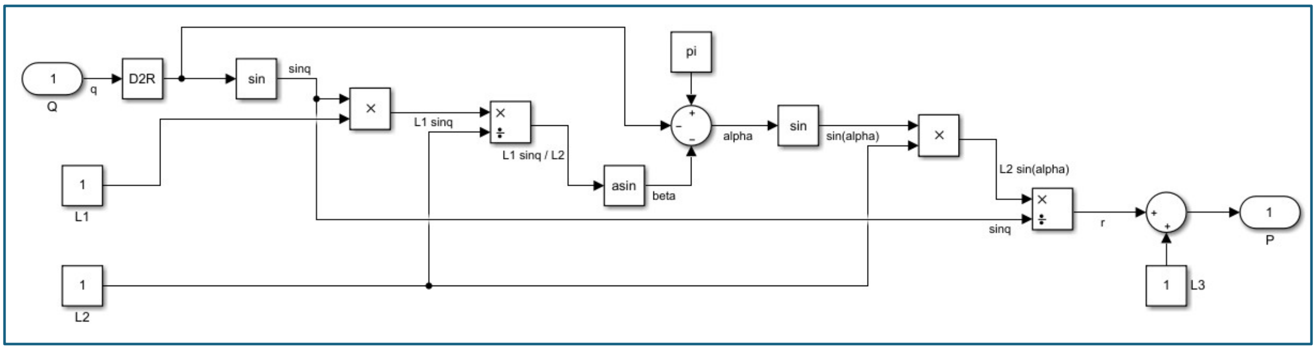



*Control Diagram for one DC motor with PID controller*

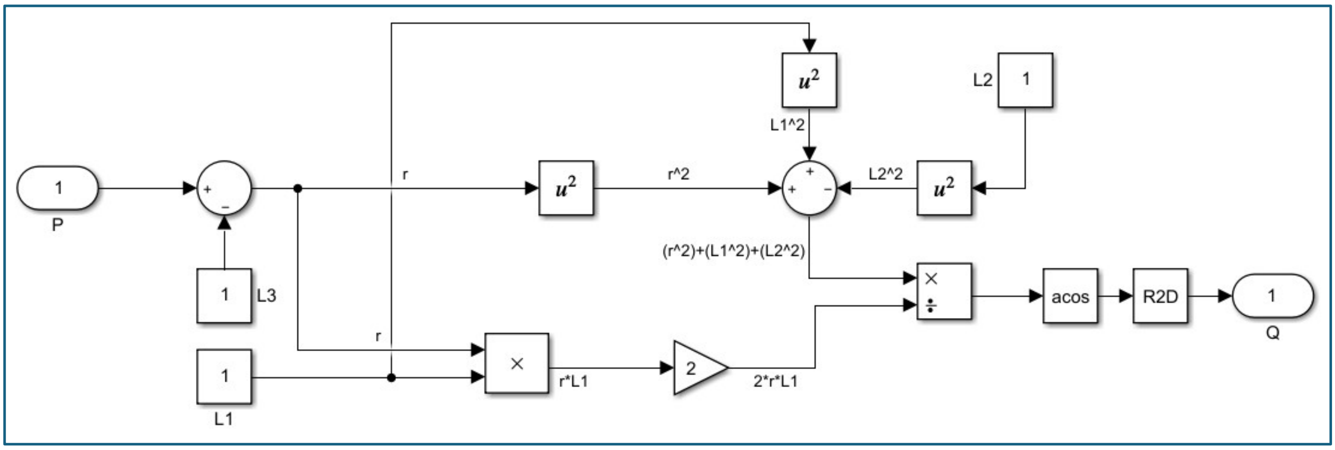
The PID - Proportional Integral Derivative - control consists in the regulation of a system's behaviour based on the difference - error - between input signal and the real output signal achieved by the system. The controller is made up of three terms: the error, taken as function of time e(t), its integral and its derivative forms. The three components are first adjusted by some gain values, respectively Kp, Ki and Kd, and then added together to produce the appropriate control signal.

Once the program is uploaded on the board, the ‘desired angle’ and ‘actual angle’ signals are data exchanged between the microcontroller and the software through a serial communication port. By studying this system, the PID controller is tuned, and the gain values are set as the most suitable for the majority of motor angles and applications.

Robot kinematics is built and tested on Simulink. The block diagrams represent the mathematical equations for the forward and inverse kinematics.



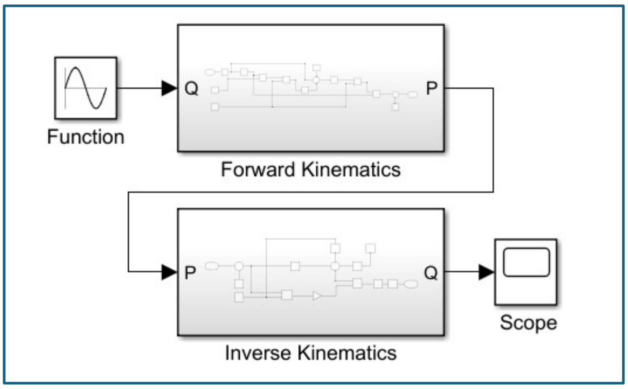

*Simulink block diagram for forward kinematics*



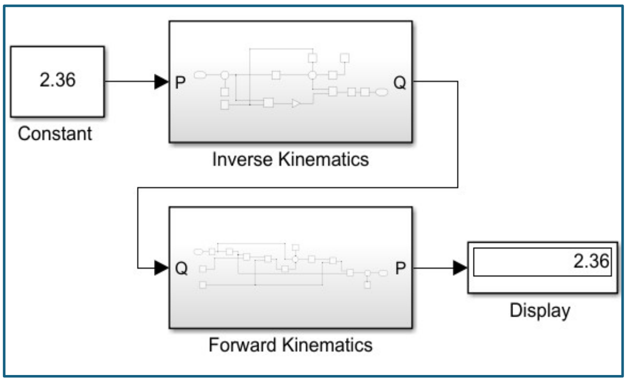

*Simulink block diagram for inverse kinematics*

It is important to consider that for the kinematics calculations on Simulink input angles must be converted from degree to radians, vice versa for output angles. P stays for position of end-effector while Q is the angular displacement of the actuator.

The two block diagrams are referred to as subsystems, and they are inserted in the same Simulink model and connected together to verify that the outcome of the calculations is always correct for both subsystems.

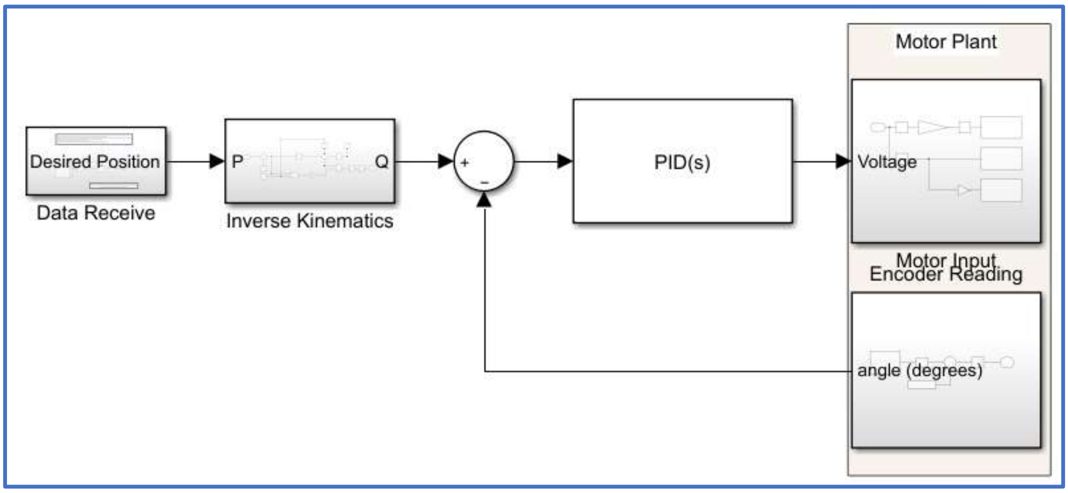



*kinematics results check 1*



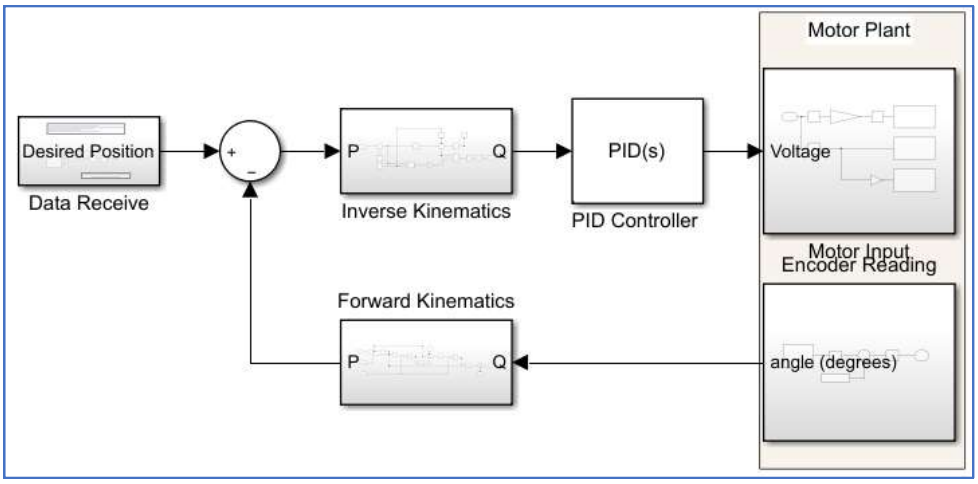

*kinematics results check 2*

Once the kinematic blocks have proven to work correctly, the algorithms are implemented in the Simulink control diagram with the PID controller for the DC motor.




*Simulink block diagram of kinematics PID controller for the robot*

The two kinematics blocks can be interchanged according to the application of the device, or they can be used together, changing the type of error signal:




*Simulink block diagram of kinematics PID controller for the robot 2*