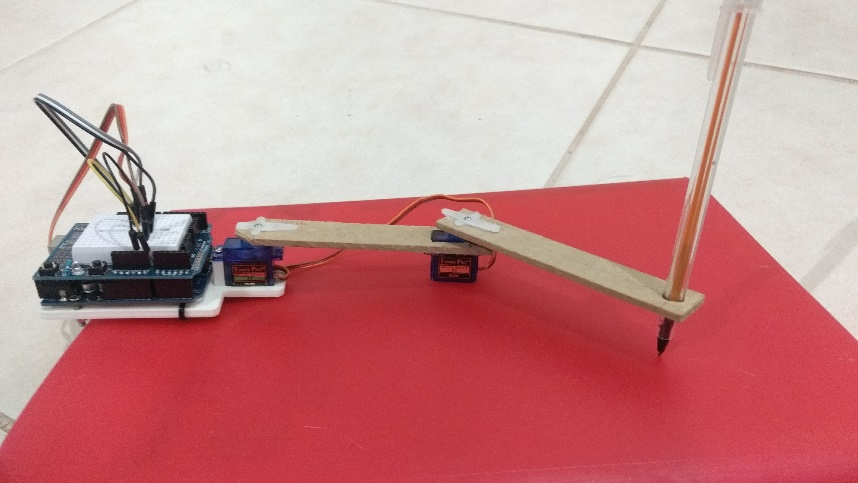
1. Introduction
2. Robot design

2.1            Physical structure

The Scara didactic robot was built with the following components:

* Plate I in Ardu One;
* Servomotor SG90, used in aero modeling;
* L inks of MDF with 120mm . Elo1 99mm between axles and Elo2 100 mm between axles ;
* External power supply (9V alkaline battery) ;
* Clamping attachment base for flat surfaces;
* Terminal organ: ballpoint pen;



2.2  Software

The following code, titled ServoSimples.ino, was shipped to the microcontroller through the official IDE of the Arduí prototyping platform . Its function is basically to zero the robot when it is initialized, sending the terminal effector to the position (0,200) and establish a communication with the serial port to receive values ​​referring to the angles that must be sent to the servo motors.

#include < Servo .h>

  Servo shoulder;

Servo board;

void setup () {

  Serial.begin (9600);

// definition of pins

shoulder. attach (10);

together. attach (11);

initprogram ();

}

// auxiliary variables

int a, pos1 = 0;

int b, pos2 = 0;

int communicates = 0;

void loop () {

comunica = Serial.read ();

delay (40);

if (( Serial.available ()> 1) && (communicate)) {

// getting data

a = Serial.read ();

delay (25);

b = Serial.read ();

delay (25);

shoulder. write (a);

together. write (b);

}

}

void initprogram () {

int au1 = pos1, au2 = pos2;

f or (au1 = pos1; au1! = 95; au1 ++) {

shoulder. write (au1);

delay (10);}

for (au2 = pos2; au2! = 160; au2 ++) {

together. write (au2);

delay (10);}

pos1 = shoulder. read ();

pos2 = joint. read ();

}

1. Kinematic modeling

3.1     Direct Kinematic Modeling (Denavit-Hartemberg)

The direct kinematics of the robot, which provides the Cartesian coordinates of the terminal effector in function of the variables of joints, was obtained, according to the specification of the work, according to the Denavit-Hartemberg algorithm for modeling mechanisms:

STEP1: identification of joints and positioning of coordinate systems for each link



STEP 2 : Table with the basic DH parameters

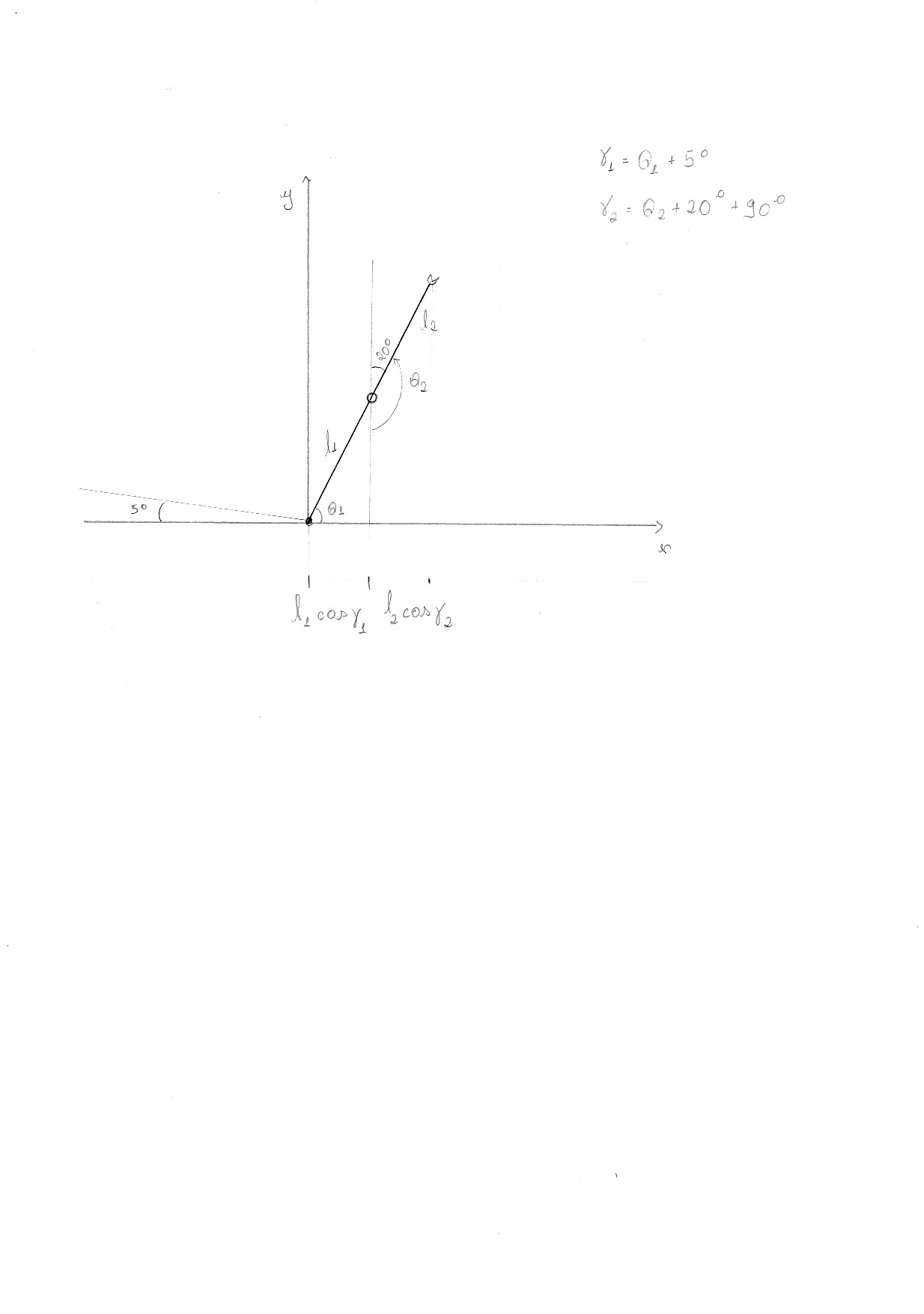
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | The | α | d | (i.e. |
| board1 | 99 | 0 | 0 | ɵ 1 + 5 |
| board2 | 100 | 0 | 0 | ɵ 2 +20 |

STEP 3: homogeneous matrices for each transformation of the coordinate system

 STEP 4: transformation claw basis for the multiplication of transformations for each link

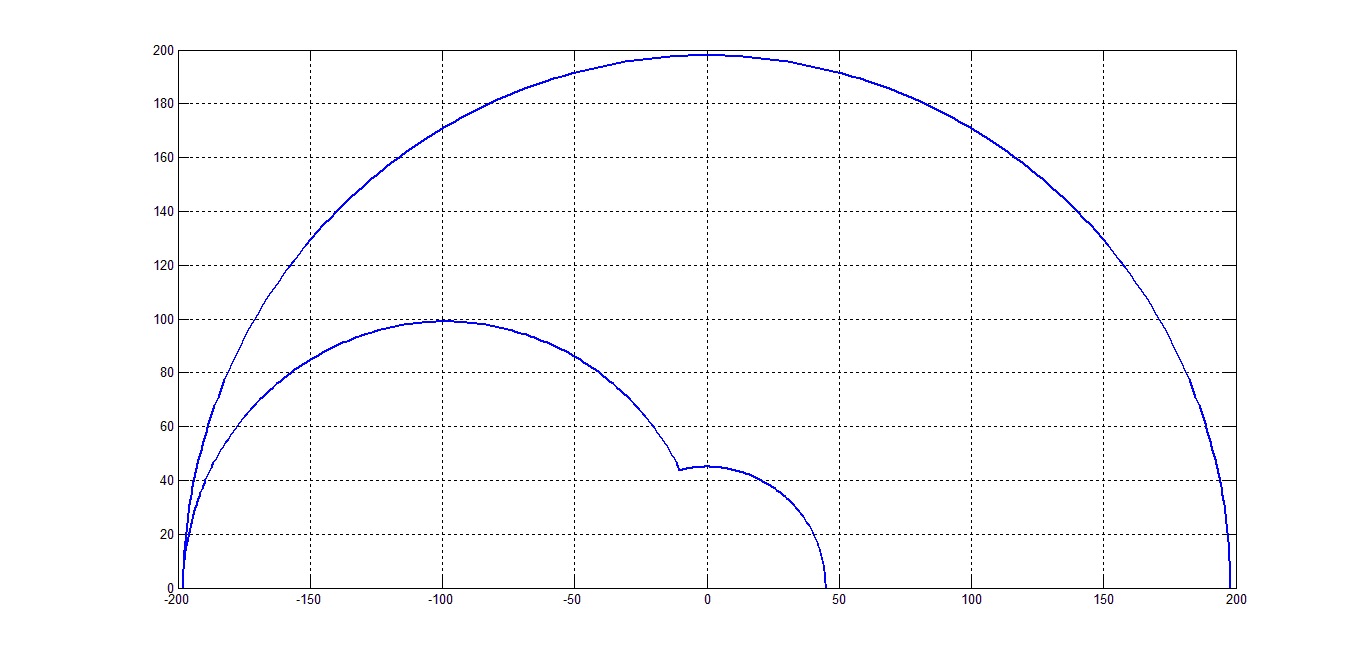
The Cartesian variables correspond to the first two elements of the last column of the final homogeneous matrix, so the direct kinematics is defined by the following equations:

3.2     Inverse Kinematic Modeling (Geometric)

 3.2     Inverse Kinematic Modeling (Geometric)

Therefore the inverse kinematics is defined by the equations:

3.3     Work volume



The modeled area of the workload is limited by three semicircles described:

Semicircle1:

Semicircle2:

Semicircle 3:

Therefore it comprises the points (x, y) such that   and

1. Stages of Work Development

4.1     Phase 1

In this step, the program moves the terminal effector to position (x, y) by linking one link and then another, followed by interpolation in the domain of the joints it generates smooth movements in the Cartesian domain.

It was necessary to create in MATLAB a function to verify if the requested point (x, y) is inside the working volume of the robot. If the point does not meet this condition, the program is aborted. It was also necessary to code the equations that describe the inverse kinematics of the robot. This function receives the coordinates of the point to be reached and the current angles of the joints so that it is possible to calculate the angles that should be imposed on the actuators based on the best path option between the current point and the desired point.

Having defined the joint variables, vectors are created with the intermediate values ​​that must be sent to the servos so that they make smooth movements in order to avoid peaks of acceleration and mechanical shocks. The angles are then sent to the robot through serial communication between MATLAB and arduino.

4.2     Level 2

The program moves the terminal effector to the (x, y) position, moving the two links at the same time with the same speed.

Using the workflow verification and reverse kinematics functions created in the previous phase , this step evaluates the angular distance that each motor will have to describe. A speed is then set for both. Then the intermediate points are found for each actuator, considering this speed.

In order to be able to use the same program previously shipped on the arduino board, the trajectories sent to each motor must be the same size, so the motor that starts the movement closest to its final destination continues to receive the final value of the trajectory until the other motor completes its movement.

4.3     Phase 3

The effector reaches the position (x, y) moving the two links simultaneously, with speeds proportional finishing movement together.

Again involving the items developed in the previous steps, in this step a variable of proportionality, obtained through the module of the difference between the final and initial angle of the shoulder, impose the number of intermediate points for all actuators. In this way, their trajectories will be completed simultaneously.

1. Conclusion
2. Refer and Bibliographic TRENDS