ISSN: 2594 - 2905

MODULAR EXPERIMENTATION OF A MODULAR MANIPULATION ROBOT WITH A VA

EXPERIMENTATION OF A MODULAR MANIPULATION ROBOT WITH A VA

Franco Gabriel Caballero Julián[1], Eric Mario Silva Cruz[2], César Hernández Sánchez[3], Anayansi Cristina Hernández Abrego[4], Martín Vidal Reyes[5]

https://doi.org/10.61117/ipsumtec.v7i2.317

[1]Master of Science. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Department of Electronic Engineering. E-mail: franco.caballero@itoaxaca.edu.mx951 1346062. Address: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, https://orcid.org/0000-0002-5924-7759

[2]Doctor of Science. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Department of Electronic Engineering. E-mail: eric.cruz@itoaxaca.edu.mx, telephone:9511065627. Address: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, https://orcid.org/0000-0002-0496-9682

[3]Master of Science. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Department of Electronic Engineering. E-mail: cesar.hernandez@itoaxaca.edu.mx, phone: 9511959974. Address: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, https://orcid.org/0009-0002-2890-8099

[4]Master of Science. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Department of Systems and Computing. E-mail: anayansi.hernandez@itoaxaca.edu.mx, phone: 9515690697. Address: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, https://orcid.org/0000-0002-8882-4842

[5]Master in Education. Tecnológico Nacional de México, Instituto Tecnológico de Oaxaca, Department of Electronic Engineering. E-mail: martin.vidal@itoaxaca.tecnm.mx, 951 2291428telephone: 951 2291428. Address: Av. Ing. Víctor Bravo Ahuja 125, esq. Calz. Tecnológico; Oaxaca, Oax., CP 68000, https://orcid.org/0009-0009-1361-0645

Abstract - - This paper is a second delivery in the magazine *IPSUMTEC* of the *TecNM/Instituto Tecnológico de Milpa Alta*, of the partial results achieved in the project of a *Manipulation Robot*.
[1] with *Machine Vision (VA)* for applications with fruits and seeds in the agricultural and food industry. Here is presented the schematic and wiring diagram between the *Arduino Mega* board and the robot structure, its servomotors and potentiometers that by means of gear structures, read the position of each of the servomotors that emulate the links of the trunk or waist, shoulder and elbow, in a behavior like the *Cincinnati Milacron T^3* robot and the *PUMA* robot of *Unimation*[2].
In the operation, we start from an image in jpg format from which a circular figure of red, green or blue color is extracted, because the background is colored white; from that figure we obtain the color and its position in a plane divided into three rows by three columns or 9 sectors. In the master-slave operation, the computer passes to the *Arduino-Mega* board the color and the sector in which the figure is identified.
With the color and position data received, the digital slave system executes the positioning routines of a light pointer that points in a plane towards the sector in which the figure was found.
What follows from this project, future work; is the development of simple applications by means of motions with *inverse kinematics* [3,4].

Keywords: Arduino Mega, control, Python, robot, Machine Vision.

Abstract - - In this issue, a second installment is made in the *IPSUMTEC* magazine of the *TecNM/Technological Institute of Milpa Alta*, of the partial results achieved in the project of a *Manipulator Robot* with *Artificial Vision(VA)* for applications with fruits and seeds in the agricultural industry and of food. Here is the schematic and wiring diagram between the *Arduino Mega* card and the structure of the robot, its servomotors and the potentiometers that, through gear structures, read the position of each of the servomotors that emulate the links of the trunk or waist, of the shoulder and elbow, in a behavior like the *Cincinnati Milacron T^3* robot and the PUMA robot from *Unimation*.
In the operation, we start from an image in jpg format from which a circular figure of red, green or blue is extracted, because the background is colored white; from that figure we obtain the color and its position in a plane divided into three lines by three columns or 9 sectors. In master-slave operation, the computer passes to the *Arduino-Mega* card the color and sector in which the figure is identified.
With the color and position data received, the slave digital system executes the positioning routines of a light marker that points in a plane towards the sector in which the figure was found.
What follows from this project, future work; is the development of simple applications through movements with inverse kinematics.

Key words: Arduino Mega, control, Python, robot, Artificial vision.

INTRODUCTION

This article is a second installment of the results of the research and experimental work on the development of a *Manipulation Robot* with *Artificial Vision* in the area of Electronic Engineering of the *TecNM/Instituto Tecnológico de Oaxaca*, whose antecedent is in the publication number 6, volume 6, special edition of the magazine *IPSUMTEC* year 2023.

The project consists of a prototype robotic arm with the ability to perform gripping, manipulation, and positioning tasks [5]. In the prototype we take as a reference the behavior and functionality of the human arm, for the execution of various routine tasks.

In this project of a three-degree-of-freedom Gripper *Robot Manipulator* with *Artificial Vision* [6]divided into two sections is intended:

1) Design and build the robot by appropriate selection of materials and mechanical components, electrical, embedded systems, computer, *encoders* and programming software.
2) Environment programming and application design.

In the article of the year 2023, we deliver the results of section 1. To solve section 2 in this paper we present the modular experimentation divided into two parts: image recognition and motion tracing respectively by means of pre-established basic routines; and we still leave for future work, the experimentation and design of applications in the solution of needs in the agricultural sector and in the preparation of food.

Three aspects are presented in this article:

a) image recognition section describes the work for the identification of an object [7] by its dominant color, and its position in the viewing plane.
b) The Mechanics section exposes the execution of harmonic movements in the three motors by means of routines programmed in *Python-Arduino*.
c) The recognition-execution interface in which, depending on the result of the image processing, a drive is executed on the robot.

Programming Environment

The digital system is through communication in a Master-Slave scheme, where the master is a PC with *Windows 10* operating system and the slave is an *Arduino Mega* module. On the PC we installed *Python 3.11* for 64 bits and it is enabled from *Visual Studio Code* to make the development environment more friendly.

The interface between *Python* and *Arduino* allows bidirectional communication between the two, which facilitates the exchange of information and commands between the *Python* control software and the *Arduino-controlled* robotic arm hardware. The *Arduino* installation is more in the public domain among technology specialists, and of interest is the choice of the serial communication port, the development board itself, and the 9600 *baud* data rate.

Python has a wide variety of libraries and modules that can be imported into programs to perform different tasks. These libraries and modules include tools for data analysis, graph creation, image manipulation and graphical user interface construction, among others.

Numpy is a *Python* library that creates a *multidimensional array* object, derived objects and a number of *array* options. Arrays, in a strict sense, are a type of two-dimensional *array*, since they have a precise length and width.

Numpy is based on a data structure called an "array" that is similar to mathematical arrays and vectors. *Numpy* arrays allow you to perform numerical operations efficiently on large data sets, which is often faster than using *for* loops in *Python.*

It can be installed from CMD using the following command.

pip install numpy

The serial library provides an interface for working with serial devices, such as sensors, actuators, *Arduino*, microcontrollers and other devices that connect through a serial port. It allows developers to send and receive data through the serial port using the serial communication protocol.

The serial library also provides functions to configure and control the data rate, number of data bits, parity and number of stop bits. These parameters are set so that the serial device and the software communicate correctly.

It can be installed from CMD using the following command:

pip install serial

The *Open CV2* library

Its main use covers object and face detection, especially in areas such as security, marketing or even photography.

It can be installed from CMD using the following command:

pip install opencv-python

Image recognition

By means of *Open CV*, an *Intel* library, and an external video camera we provide *Python* with tools that make the interaction easier, an image is captured, put in jpg format and there the processing is done. In this release, the plane is divided into 9 quadrants and the test images are filled circles only for the three primary colors: red, blue and green; any other color be associated with these. In the identification, a variable combining color and position is returned.

Mechanical Section

This section organizes the movements of the three joints, for three colors and 9 sectors.

The degrees of freedom in this robotic arm model refer to the different ways in which it can move or position itself. The number of degrees of freedom [8] of this robotic arm may vary depending on its design and configuration, but common degrees of freedom include:

Base degree of freedom: This degree of freedom allows the entire robotic arm to rotate around its base. It is rotational movement on the vertical axis that allows the arm to change its general orientation.

Degree of freedom at the shoulder: This degree of freedom allows the arm to move up and down in a vertical plane.

Elbow degree of freedom: This degree of freedom allows the arm to flex or extend in a vertical plane. It is the flexion or extension movement of the arm and is performed in the horizontal axis.

Gripper degree of freedom: This degree of freedom allows to open or close the gripper or manipulation tool of the robotic arm (generates the movement of gripping or releasing objects).

Mechanical Structure

Mechanical coupling: To know the trajectories, the positioning of a servomotor is measured using 100K potentiometers because they can provide more accurate and real time information about the current position of the servomotor [9].

Position control: Knowing the current position of the servomotor through the potentiometers, it is possible to implement position control algorithms. This allows to set a desired position and use the feedback information to adjust the control signal and get the servomotor to position itself at the desired destination.

The objective of this test is to place the 100K potentiometers on the robot structure, using mechanical, power transmission, tension force, extra couplings, etc. To achieve these couplings, basic tests were made, moving servomotor by servomotor to identify the 0 positioning point. Once the identification was achieved, the following improvements to the model were chosen:

Attachment of a more resistant base: The coupling of the base or platform, possesses the purpose of keeping the robot always standing.

Gear coupling: they are used to transmit energy from a servomotor to a potentiometer, thanks to the fundamental principles of mechanics.

Support coupling: These are essential materials to keep the potentiometer fixed in a joint of the robot[10], so that it remains fixed at the moment of mechanical energy transmission.

Coupling **of terminal blocks:** The 3-space terminal blocks were used in order to keep the potentiometer power supply cables fixed, since they were coming loose from the potentiometers and could cause delays in the development of the prototype.

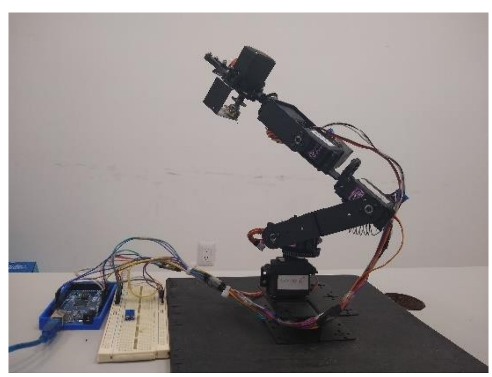

Figure 1. Assembly of the original prototype structure.

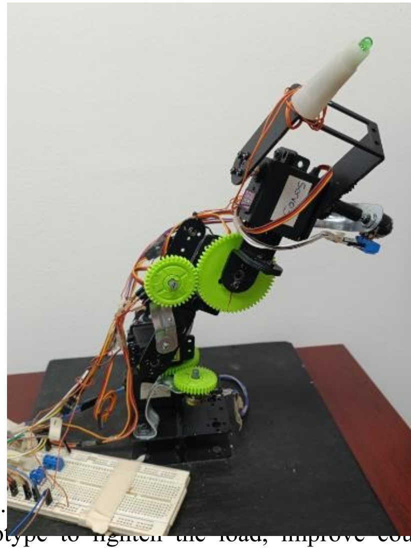

Figure 2. ructure of the prototype to lighten the load, improve coupling and interaction.

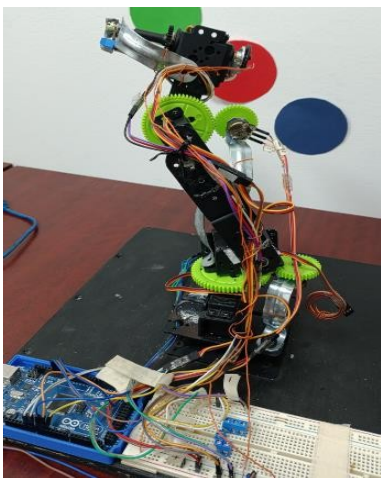

Figure 3. Complete view of the mechanical structure of the prototype with its connections to the Arduino Mega board.

DEVELOPMENT

In the previous section of this publication, the block diagram of the general structure of the *Artificial Vision Manipulation Robot* and the general form of the flow diagram of the projection of its operation were presented. In this section, the electrical diagram of the prototype, the procedure for the identification of figures from an image capture and the three pre-established execution routines are presented.

Electrical diagram

The electrical diagram is presented by means of two products in academic version, the simulation in *TinkerCad* in which an *Arduino UNO* board is used and the simulation in *Proteus* in which an *Arduino Mega* board is used, whose implementation is taken to the physical prototype.

The following assignments are made in the design:

One pair of wires, the 5V power supply and its ground for each of the three servos named *servo1*, *servo2* and *servo3.*
servo1 connected for control to D9 on the card
Arduino.
servo2 connected for control to D10 on the card
Arduino.
servo3 connected for control to D11 on the card
Arduino.

A pair of wires, the 5V power supply and its ground for each of the three position potentiometers named Pot1, Pot2 and Pot3 associated to the *servo1*, *servo2* and *servo3* servo motors.

Pot1 connected to the analog input A0 on the card
Arduino to read the position of *servo1*.
Pot2 connected to the analog input A1 on the card
Arduino to read the position of *servo2.*
Pot3 connected to analog input A2 on the card
Arduino to read *servo position3.*

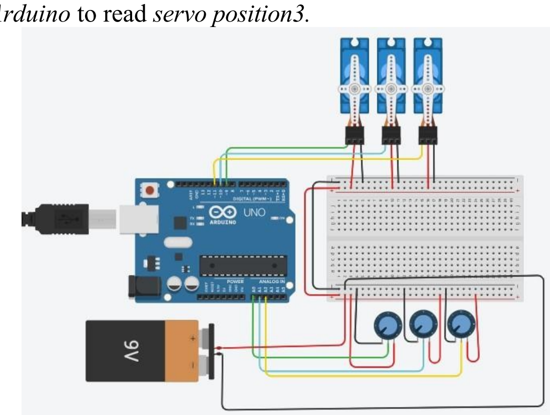

Figure 4. Representation of the electrical diagram for simulation in Arduino UNO.

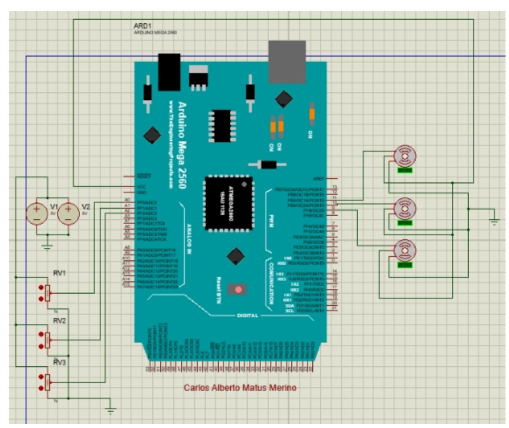

Figure 5. Representation of the electrical diagram for simulation in Arduino MEGA.

Relationship of the mechanisms

In this prototype, the joints of the *roll*, *pitch* and deflection operations, representing the trunk, shoulder and elbow, are available in reference to the *Cincinnati Milacron T^3* and *Unimation*'s *PUMA* robot.

The *servo1* and *servo2* mechanisms, which correspond to the trunk and shoulder, have a motion transmission system by means of gears with a 60:36 tooth ratio; such that in the stroke of only 180 degrees, they transmit almost a complete turn on the potentiometer.

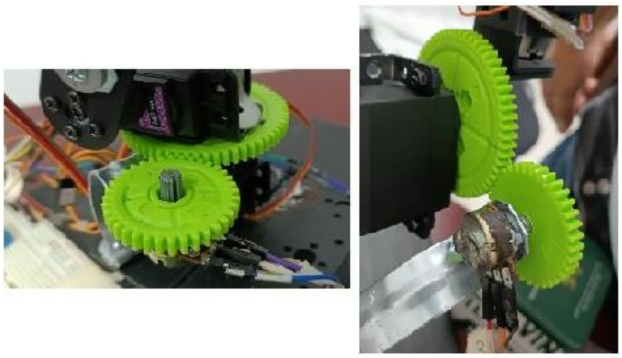

Figure 6. Transmission of trunk and shoulder position by means of gears.

With a resolution of six degrees or 166.6 mV in forward or reverse on the trunk or shoulder per tooth.

For the elbow mechanism, called *servo3* has a system of direct transmission of the movement to the potentiometer 3, without a gear in between.

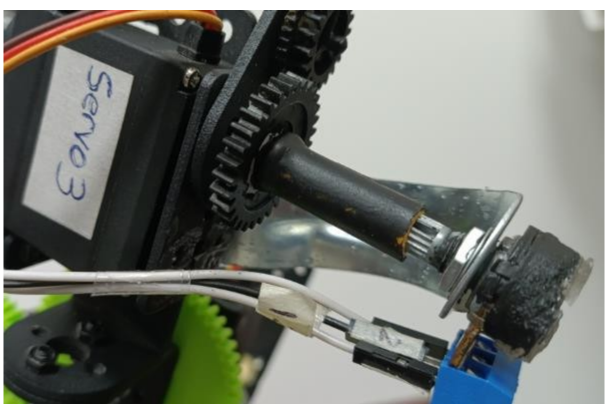

Figure 7. Transmission of the elbow position.

Image extraction

The experimental work[11] is with templates in jpg format created in *processing,* of 250 x 250 pixels of a size close to 10 KB in the understanding that video captures *(frames)* are extracted from the , in each template there is a round figure that can be the color of any of the primary colors: red, green or blue.

The background of each template is gray, like a board and is divided into three rows by three columns; in nine sectors hereafter referred to as *sector n* for n=1,...,9. All possible combinations are in total 3 colors x 9 sectors, in total 27 possible combinations, only one color at a time.

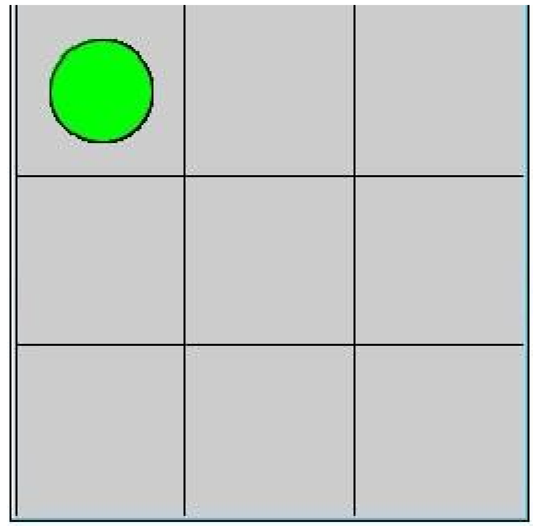

Figure 8. A green figure with position in row 1, column 1, named sector 1; extracted from the video camera for processing.

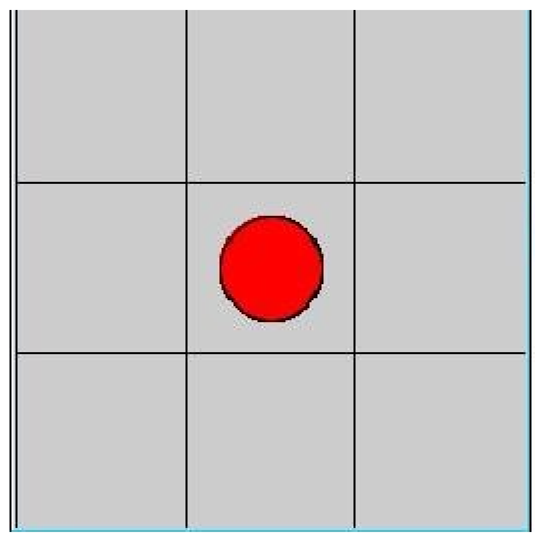

Figure 9. A red figure with position in row 2, column 2, named sector 5; extracted from the video camera for processing.

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
IPSUMTEC
ISSN: 2594 - 2905

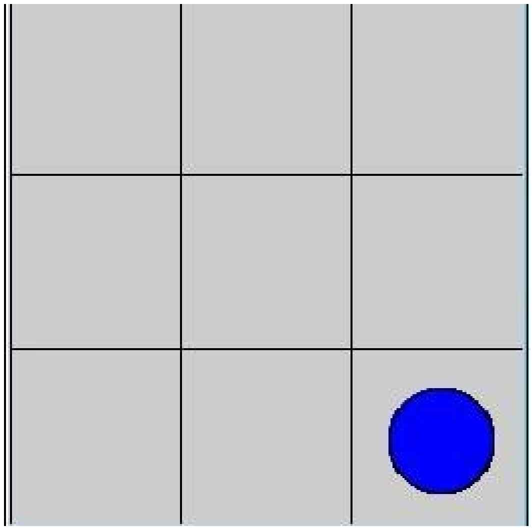

Figure 10. A blue figure with position in row 3, column 3, named sector 9; extracted from the video camera for processing.

Recognition

The recognition stage [12] consists of obtaining an image in jpg format, the source image is in BGR tones, it is converted to RGB, the figure is extracted which in this case is a concentric circle and a filter is applied to leave the figure on a canvas with a white background to eliminate any confusion; the color is identified and average between the maximum and minimum of an array that collects the number of rows and columns in which the figure is obtained its X,Y position. The computer transmits the color and the X,Y position to the *Arduino* board.

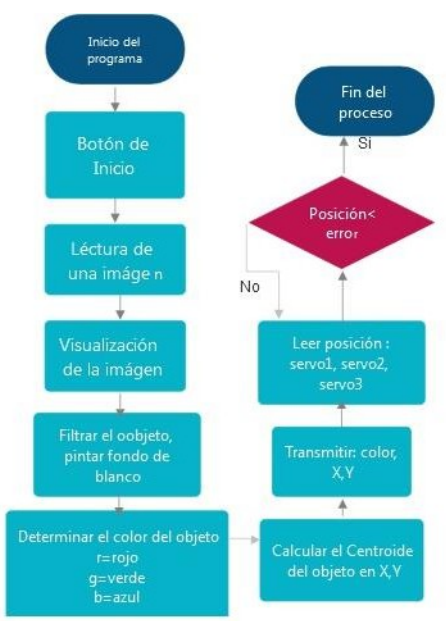

Figure 11. Flowchart of the process of identifying an image, filtering, white background placement, color identification, X,Y position; sector and variable transmission.

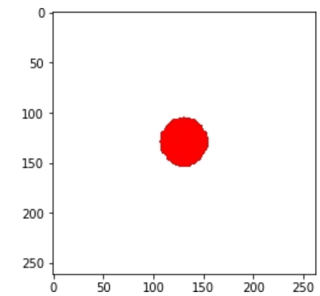

Figure 12. Filtering of the image with white background, you can see its centroid X,Y; and a sector is assigned to it.

```
In [43]: arr2
Out[43]: array([130., 131., 125., ..., 134., 135., 136.])

In [44]: np.amin(arr2)
Out[44]: 106.0

In [45]: np.amax(arr2)
Out[45]: 155.0
```

Figure 13. From an array, the columns in which the figure is painted are extracted, from the maximum and minimum the midpoint is obtained, it assigned to Y. The same is done to obtain X from the rows. A sector is assigned to it.

For the scope of this publication, only three simple routines are available[13], consisting of the regulated movement of the three motors in the sequences indicated.

We define *varTransm*, an integer obtained from the recognition of a color figure and its position.

VarTransm=0 at the start of the program and if a figure is not recognized.
VarTransm=1 if a green figure is recognized in row 1, column 1 of the board.
VarTransm=2 if a red figure is recognized in row 2, column 2 of the board.
VarTransm=3 if a blue figure is recognized in row 3, column 3 of the board.

Robot Movements

The operation of the robot consists in pointing a light pointer, depending on the color and the position of image on the plane; thus, for a green color in sector 1 of the image, it points to sector 1 of the viewing plane. For a red color in sector 5 of the image, it points to sector 5 of its viewing plane. For a blue color sector 9 of the image points to sector 9 of your plane. Note that the plane of vision of the robot is not well defined because it is not a Cartesian robot, but its displacement is in a volume space.

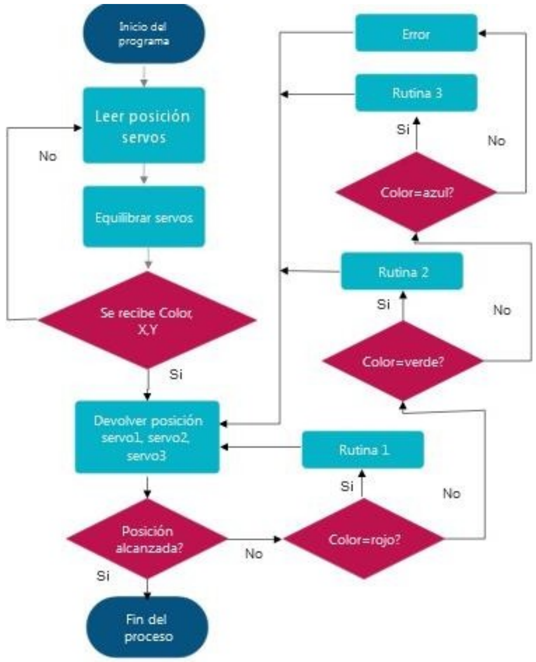

Figure 14. Robot drive. By means of three different routines the pointer is positioned in sector 1, sector 5 or sector 9.

At the moment, depending on the value of the integer variable *VarTransm*, one of the three routines programmed in the sequence indicated is executed.

Routine 1.- It is executed when the green colored figure is identified in any of the three positions of the first line.
Routine 1: Motor1, motor2, motor 3

Routine 2.- It is executed when the red colored figure is identified in any of the three positions of the second line.
Routine 2 : Motor 2, motor 3, motor 1

Routine 3.- It is executed when the blue figure is identified in any of the three positions of the third line.
Routine 3: Motor 3, motor 1, motor 2

The servomotors in this prototype [14] rotate counterclockwise from their reference at zero for values of the desired angle given in positive values, thus Servomotor 1, mounted on the Y axis, rotates in the X plane; it represents the trunk, has a sweep ranging from 30^0 to position itself in column 3, commands 60^0 to position itself in column 2, and commands 90^0 position itself in column 1. Servomotor 2, mounted on the Z-axis, rotates in the Y-plane; it represents the shoulder, has a sweep from 30^0 to position at row 3, commands 60^0 to position at row 2 and commands 90^0 to position at row 1.

Servomotor 3, mounted on the X-axis, rotates in the Z-plane; it represents the elbow, has a spatial adjustment or sweep offset function that goes from 90^0 as an extension in row 3, commands 60^0 for offset in row 2, and commands 30^0 for offset in row 1.

```
void moveRoutine1() {
servo1.write(90);
 Serial.println("Trunk movement 1: ");
 Serial.println("Potentiometer voltage at 90 degrees: ");
 Serial.println(Vout0);
 delay(2000); servo2.write(90);
 Serial.println("Shoulder movement 2: ");
 Serial.println("Potentiometer voltage at 90 degrees: ");
 Serial.println(Vout1);
 delay(2000); servo3.write(30);
 Serial.println("Elbow movement 3: ");
 Serial.println("Potentiometer voltage at 30 degrees: ");
 Serial.println(Vout2);
delay(2000);
}
void moveRoutine2() {
servo2.write(60);
 Serial.println("Shoulder movement 2: ");
 Serial.println("Potentiometer voltage at 60 degrees: ");
 Serial.println(Vout1);
 delay(2000); servo3.write(60);
 Serial.println("Elbow movement3: ");
 Serial.println("Potentiometer voltage at 60 degrees: ");
 Serial.println(Vout2);
 delay(2000); servo1.write(60);
 Serial.println("Movement1 of the trunk: ");
 Serial.println("Potentiometer voltage at 60 degrees: ");
 Serial.println(Vout0);
 delay(2000);
 }
void moveRoutine3() {
servo3.write(90);
 Serial.println("Elbow movement3: ");
 Serial.println("Potentiometer voltage at 30 degrees: ");
 Serial.println(Vout2);
 delay(2000); servo1.write(30);
 Serial.println("Movement1 of the trunk: ");
 Serial.println("Potentiometer voltage at 30 degrees: ");
 Serial.println(Vout0);
 delay(2000); servo2.write(30);
 Serial.println("Arm movement2: ");
 Serial.println("Potentiometer voltage at 60 degrees: ");
 Serial.println(Vout1);
 delay(2000);
}
```

DISCUSSION AND ANALYSIS OF RESULTS.

The structure in use, despite the different configurations, is robust, durable and consistent. The servomotors respond well to instructions, we have made some adjustments to relieve the structure of the extra weight, the gear systems transmit the movement correctly and the potentiometers return the voltage value in the range of 0-5 Volts. Problems are occurring due unforeseen misalignments such as slight displacement in the gearing or unexpected false contacts at the various wiring contact points. In the testing process, both in *Python* and through the *Arduino* serial monitoring port, these position data are read; which as indicated in the flowcharts, allow to ensure the positioning of the servomotors against the possibility of unexpected displacements especially in the X and Z axes.

In the identification and recognition, we chose a very easy example; fruits can be identified by their color and it would not be enough to find their centroid, certainly their contour will also be an important data to improve their characterization. The communication of the identification parameters to the slave system is very easy in *Python*.

On the robot mechanism side, the operation is still fully linear because not even two servomotors can run at the same time; at the moment we only reflect on the fact that the slave module can be solved with a multiprocessing arrangement to improve the performance.

For the example that is tested, we have good precision, but it is not considered a high degree of accuracy; in fact, the pointer points in a very wide range of the nine sectors of a very small plane because not being a Cartesian model, its displacement is spatial, which has to be considered exponentially in the differential subdivision in its displacement volume.

This implies raising the degree of complexity [15] of the prototype, with a more sophisticated analysis in the identification and recognition of the images and in the execution of the applications of interest.

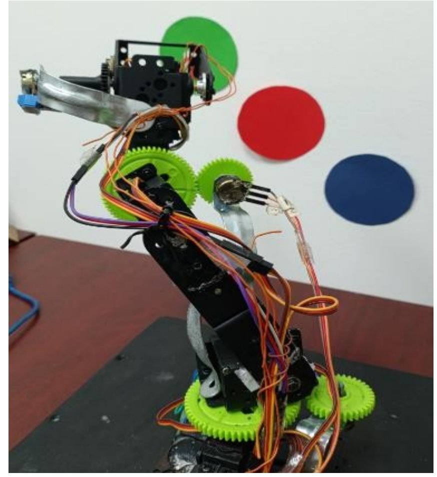

Illustration of the manipulator drive to position the pointer the region where a green object is identified, sector 1.

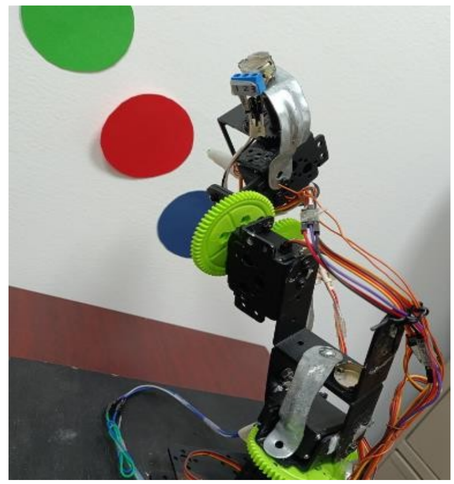

Figure 16. Illustration of the manipulator actuation to position the pointer the region where a red object is identified, sector 5.

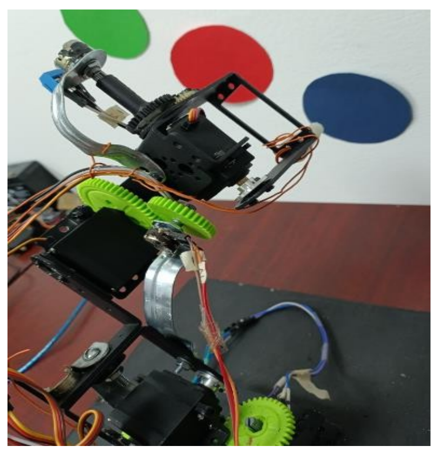

Figure 17. Illustration of the manipulator drive to position the pointer the region where a blue object is identified, sector 9.

This project is based on the kinematic model of a redundant *7 DOF* robotic arm of

ISSN: 2594 - 2905

a humanoid called *Mex-One*, in which it is applied to find the solution of inverse kinematics, in order to position the robotic arm at a specific point in space. This *7 DOF* work has been carried out by the research team for 9 years at *CINVESTAV* Guadalajara unit.

Also referenced in this study, outside the context of theoretical work only, are the experimental results of prototypes made in public education institutions in Mexico as is the case of the Master's thesis mentioned in [16] at the *Center for Innovation and Technological Development in Computing of the IPN* that develops a *Mobile Robot* with *Artificial Intelligence* and also for the elaboration of its mechanical structure we have as reference the one of the *ESIME, Section of Postgraduate Studies and Research of the IPN* in which a *Robotic Manipulator* for the training of personnel for incisive surgery is referred [17].

CONCLUSIONS

The work represents a significant advance in a far-reaching project. The partial results are encouraging, requiring many hours of planning, execution and imagination; hours and days of testing to achieve the results.

We have to work on three aspects; first in a more sophisticated image analysis and processing, the second aspect is the provision of the prototype for use in utility applications and finally we have to consider two lines of research for future work: inverse kinematics techniques that characterizes commercial robots to achieve positioning by means of matrix numerical operations; and Multiprocessing by means of several processors in collaborative environment for the distribution of tasks in order to improve response time.

ACKNOWLEDGMENTS

To the *TecNM/Instituto Tecnológico de Oaxaca* for giving us the opportunity to put into effect our ideas that begin in the mind and are based on the knowledge that is communicated and researched in this educational space, also to the *Professional Development Program (PRODEP)* that has the mechanisms to motivate the creative work of teachers and students.

BIBLIOGRAPHY .

[1] Martinez, C. (2017). Development of a soft and hyper-redundant manipulator robot. Final degree work for obtaining the degree of graduate in engineering of industrial technologies. Higher technical school of industrial engineers. Polytechnic University of Madrid.

[2] Fu, K.S., Gonzalez, R.C. & Lee, C.S. (1988). Robotics: Control, detection, vision and intelligence.

McGrawHill/ Interamericana de España S. A.

ISBN: 84-7615-214-0

[3] Reyes, F. (2011). Robotics. Control of Robot Manipulators. Alfaomega Grupo Editor, S.A. de C.V.

[4] Alvarez, J.G. (1999). Control techniques for manipulator robots with joint flexibility. PhD Thesis. School of Mechanical and Electrical Engineering. Universidad Autónoma de Nuevo León. México.

[5] Carlson, W. B. (2015). TESLA Inventor of the electric era. Ediciones culturales Paidós, S.A. de C.V. Work published in collaboration with Editorial Planeta-España.

[6] Gevarter, W. (1982). An overview of artificial intelligence and robotics Volume II- Robotics. National Aeronautics and space Administration. Headquarters. Washington D.C. 20546. U.S. Department of Commerce.

[7] International federation of Robotics (2022). Artificial Intelligence in Robotics. Position paper. Frankfurt, Germany.

[8] Smith, J, et al (2018). Advances in Manipulator Arms: Industrial Applications and Challenges. Robotics and Automation, 35(2), 45-58.

[9] Jones, A., et al. (2019). Flexible Manipulator Arms: Applications and Challenges. Journal of Robotics Engineering, 42(4), 123-136.

[10] Liu, S., et al. (2020). Impact of Manipulator Arms on Industrial Efficiency: A Case Study. International Journal of Automation and Robotics, 25(3), 87 - 102.

[11] Parzibyte (2021). Recording camera video with Python, Flask and OpenCV - Surveillance camera. Parzibyte's blog. https://parzibyte.me/blog/2021/02/13/grabar-video-camera-python-flask-opencv-camera-surveillance/

[12] Rodríguez, C., Botero, J., Quintero, H.(2007). Application of Robot Manipulator Theory to the biomechanics of the human arm. Iberoamerican Journal of Automatics and Industrial Informatics. ISSN:1697-7912 Vol. 4 pp51-57 http://riai.isa.upv.es

[13] Reyes, C., et al. (2017). Control Strategies for 2GDL Industrial Manipulator Robot for path generation. Academia Journals.com. ISSN 2153-3318. Vol. 6 No. 2 pp29.

[14] Greyrat, R. (2022b, July 5). Real-time multiple color detection using Python-Open CV- Barcelona Geeks. https://barcelonageeks.com/deteccion- de-multiple-colors-in-real-time-using-python-opencv/.

[15] Torres, I.,(2017). Development of a computer vision algorithm for the guidance of a mobile robot. Final degree thesis. Polytechnic University of Valencia.

[16] Orozco, R.(2014) Mobile Robot with Artificial Vision. Master's thesis in Computer Technology. Center for Innovation and Technological Development in Computing. IPN.

[17] Chavez, A., et al. (2009). Design and development of a robotic manipulator for the training of personnel in incisal surgery. Científica Vol.13 No. pp 145-155 ESIME, IPN. ISSN 1665-0654.

CONTRIBUTION ROLES

Role	Author(s)

Conceptualization	Franco Gabriel Caballero Julián
Data Curation	Anayansi Cristina Hernández Abrego
Methodology	César Hernández Sánchez
Project Management	Martin Vidal Reyes
Resources	Martin Vidal Reyes
Software	Anayansi Cristina Hernández Abrego
Supervision	Eric Mario Silva Cruz
Validation	Franco Gabriel Caballero Julián
Visualization	Franco Gabriel Caballero Julián
Original draft	César Hernández Sánchez (Contribution level: equal) Anayansi Cristina Hernández Abrego (Contribution level: equal) Martín Vidal Reyes (Contribution : equal)
Writing, proofreading and editing	Franco Gabriel Caballero Julián (Contribution grade: principal) Eric Mario Silva Cruz (Degree of contribution: supporting)