Available online at www.eccomasproceedia.org
Eccomas Proceedia SMART (2023) 1246-1257

ECCOMAS
Proceedia

X ECCOMAS Thematic Conference on Smart Structures and Materials
SMART 2023
D.A. Saravanos, A. Benjeddou, N. Chrysochoidis and T. Theodosiou (Eds)

DEVELOPMENT OF A CARTESIAN ROBOT WITH IMAGE PROCESSING AND GRASP DETECTION

MOHAMMAD Y. SAADEH[*] AND CRIS KOUTSOUGERAS[†]

[*] Department of Industrial and Engineering Technology
Southeastern Louisiana University
Hammond, Louisiana 70402, USA
e-mail: msaadeh@southeastern.edu, www.southeastern.edu

[†] Department of Computer Science
Southeastern Louisiana University
Hammond, Louisiana 70402, USA
e-mail: ck@southeastern.edu, www.southeastern.edu

Abstract. This work presents a Mechatronics system of a Cartesian robot that can detect, pick, and sort items. Industry 4.0 paved the way for the tedious and repetitive tasks to be automated and replaced by robots to ensure consistency, speed, and quality control maintenance. Typical industrial systems are expensive and proprietary in nature which limits their utility for educational and training purposes. In this work, we present a design of a Cartesian Robot which incorporates a variety of standard systems to implement a miniature industrial robotic setup. The systems include machine vision, trajectory planning, gripping mechanism with force limiting sensor, and hardware interfacing and communication between different platforms. The work reported here is a roadmap for building a Cartesian robot that is an alternative to standard proprietary industrial robots and which incorporates trajectory planning, vision, force limitation, and communication. This document describes the design with its major subsystems. The Cartesian robot is capable of planar motion in the X and Y dimensions, and a vertical linear motion to pick and place objects. An overhead camera for object and color detection takes still images and processes them on a computer that is interfaced with the control board. A calibration method for the camera is described using red, green and blue cubes randomly placed on the workspace. Joints follow a Linear Segment with Parabolic Blend (LSPB) trajectory to achieve the joint motions. A remote host computer employs code that detects the center and the corners of each cube to find the exact location and the orientation of each cube. It also maps a suitable route for picking and placing these cubes from their current locations to the destinations (the sorting bins). The end effector consists of two servo motors, one to orientate the end effector and the other to close and open the gripper. To achieve a controlled gripping force on the cubes, a force sensing resistor (which changes resistance according to pressure applied at its surface) is embedded within the gripper fingers. Preliminary results show that the system performed the required task and was able to follow the planned route successfully. The description is written for a wide technical audience but in sufficient detail for reproduction.

Peer-review under responsibility of the organizing committee of SMART 2023.
doi: 10.7712/150123.9872.442489

Key words: Mechatronics, Cartesian Robot, Trajectory Planning, Vision, System, Force Sensing Resistor.

1 INTRODUCTION

In this work, we provide the roadmap to a Cartesian robot that incorporates industrial standard systems such as machine vision, trajectory planning, gripping mechanism with force characterization, and hardware interfacing and communication. This design/roadmap is of particular interest to educators looking for affordable options of Cartesian robots for training labs but it is also of interest to developers looking for affordable customizable solutions of such robots for working environments.

All Cartesian robots are classified as 3P robots or Gantry robots. Each coordinate consists of a prismatic joint that allows it to move along one of the coordinates in the Cartesian space. These robots are used mainly in pick and place tasks due to their simple structure and practicality. Cartesian robots operating in a planar surface have an overhead framework which regulates horizontal motion, and a linear actuator which controls vertical motion. They can be configured to move in x-y or x-y-z directions. Motion on each axis is controlled by a separate motor with timing belts and pulleys. The linear actuator is positioned on the scaffolding and can move vertically to reach in the vertical coordinate.

Trajectory planning for the joints (motors) plays a significant role in delivering a reliable pick and place robotic operation. A well-defined trajectory can eliminate abrupt starting and stopping that give rise to undesired acceleration profile, especially when the operation involves delicate objects or fluids. One of the reliable trajectories in robotics is the Linear Segment with Parabolic Blends (LSPB) trajectory. The challenges with the LSPB is that the profile can be achieved through numerous ways. However, most applications have constraints that are limited by the max speed, acceleration, path selection, and time of travel such that a unique profile can be fully identified. The trajectory planning in both: joint space and Cartesian space with LSPB have been widely used in industry [1]. The LSPB method has been investigated to identify its segments at joint-space planning for a model of 2-link planar robot using neural network [2]. While in [3], they applied inverse kinematic algorithm and LSPB trajectory planning method in Cartesian space to a 6-DOF robotic arm to perform straight flat welding movements.

Some critics of the normal LSPB proposed that it is restricted so that the acceleration must be sufficiently high. The work presented in [4] proposed a modified LSPB that is engaged with Particle Swarm Optimization to generate smooth trajectories that pass through specific path points while satisfying velocity and acceleration constraints of physical mechanical robot manipulators. Another approach to address the shortcoming of the LSPB was to take the manipulator dynamics into consideration [5]. In their proposed method, they performed the dynamic calculation only at certain points on the given path, which is effectively utilized to determine the kinematic constraints while minimizing the computational burden by solving only at dynamic edge points on the given path. The application of the path trajectory on this

work is limited to simple pick and place operations. Thus, the normal LSPB with constraints on the acceleration, velocity, and path travel time is applied.

Several works used the Force Sensing Resistor (FSR) for robotic applications due to its simple structure, ease of interfacing, low cost and small footprint. For instance, in [6] the authors used FSR as a transducer to develop a force moment sensor. The work presented in [7] used a PI control method with FSR as input to control gripping force of a 5 DOF robot. Two FSR matrices have been used to construct a two-layer artificial robotic skin that detect location, value and direction of forces acting on the skin [8]. In [9], they showed how the FSR can be used in several applications such hardness detection and force-position control. They found that the FSR was able to produce satisfactory and reliable results. FSRs have been used in haptic applications such as in [10]. They proposed haptic interaction with force feedback for an exoskeleton and used two FSRs for measuring the force applied at the jaws. A flexi-force sensing resistor was calibrated to be attached to the claws of a two fingered gripper to achieve proper grasping force and prevent slippage [11]. In [12] they used an embedded FSR in a finger-wearable electronic refreshable Braille device to measure force applied at the finger pad. FSRs continue to be utilized in many industries and applications due to their versatility and ease of implementation.

In the following we present a Cartesian robot design which provides a design example which accounts and balances the above considerations and challenges.

2 EXPERIMENTAL SETUP

An experimental model was built using rails, belts and stepper motors mounted on a frame. The conceptual design is shown in Figure 1a. The actual system is shown in Figure 1b. The 3-fingered end-effector is shown in Figure 2.

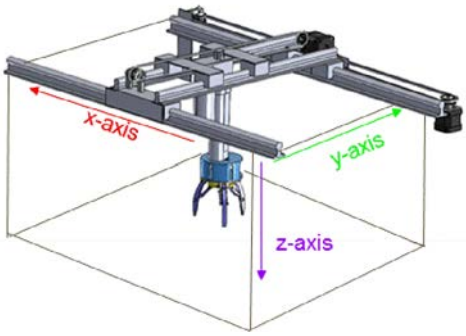

Figure 1a: The model

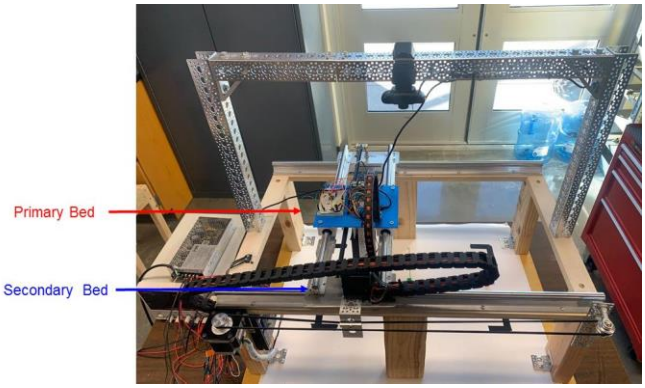

Figure 1b: Experimental setup

The x-axis motor is mounted on one side of the frame opposite to a pulley and bearing, and a timing belt attached to the motor on one end and the pulley on the other; this facilitates the motion on the x-axis. The y-axis works similarly. Attached to the end of a vertically placed linear actuator is a 2-finger gripper for the picking and placing of objects. Objects are detected by a camera that is mounted onto an overhead scaffolding above the Cartesian plane and

analyzed and controlled by a Raspberry Pi board.

The selected end-effector is a 2-finger gripping system for the picking and placing of objects with an embedded FSR. Objects are detected through a camera, then processed on an attached processor for edge and location detection; the processed information and the set of motions are then coded and passed to a Raspberry pi board that controls the actuators of the manipulator

Figure 2: End Effector (gripper) with attached driver motor and linear actuator for vertical motion

2.1 Motors selection

Stepper motors are selected to move the bed in the X and Y directions; a stepper motor offers a good choice for synchronized and incremental motion that allows implementation of precise positioning and speed trajectory. Two stepper motor driver boards are attached to a Raspberry Pi to produce control signals to the motors.

To properly select the suitable motors for the X and Y motion, all forces acting on the motors should be considered. There are two main forces acting on each motor, the static force due to friction, and the dynamic force at the beginning of the motion.

$$F = Ma + F_{st} \tag{1}$$

$$F_{st} = M\ g\ \mu \tag{2}$$

$$T = F \times\ d \tag{3}$$

where M is the mass of all the components that are moving in the coordinate, a is the acceleration required, F_{st} is the static force due to friction, g is the acceleration due to gravity, μ is the friction coefficient, T is the torque of the motor, and d is the pulley's radius.

The friction inside the motor bearings and stator is considered negligible compared to the friction between the moving bed and the rails. This work follows the Linear Segment with Parabolic Blends trajectory to achieve a smooth motor motion. The selected trajectory provides faster transitioning between points with ramping up and down speeds towards the two ends of the trajectory. Furthermore, some constraints can be applied through this trajectory profile such as: total travel time, maximum speed, and the length of the parabolic segments.

The maximum acceleration can also be limited in this trajectory (to properly size the motors and better control its bandwidth). Figure 3 below shows an example trajectory for a 10 inch travel in 5 seconds.

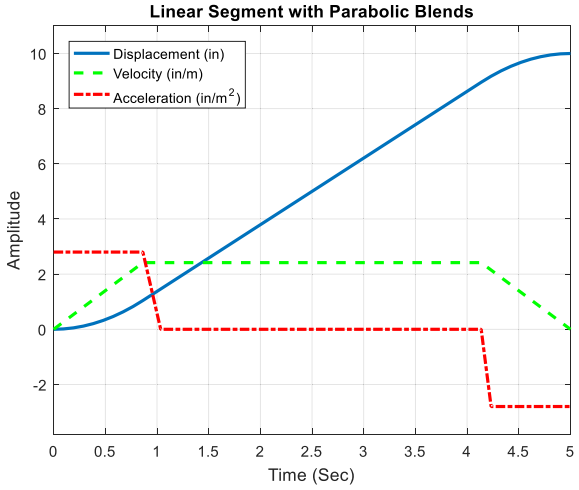

Figure 3: LSPB Trajectory Profile

Table I shows the parameters of the Cartesian robot and the trajectory constraints (motion):

Table I: Robot and Trajectory Constraints

Max speed (v_m)	2.5 in/sec
Max acceleration (a_m)	3 in/s^2
Mass of bed on the secondary rails (y-axis) M_s	2.44 kg
Mass of bed on the major (primary) rails (x-axis) M_p	3.55 kg
Mass of the payload and gripper m_p	0.45 kg
Mass of each block m_b	0.05 kg
Coefficient of friction between rails and the bearings μ_r	0.23
Coefficient of friction between gripper and plastic blocks μ_p	0.34

All the forces that are involved in each motion (in X and Y coordinates) are included in the calculation of the required force for each motor, as follows: the motion in y-coordinate involves moving the secondary bed and the gripper with the payload. The motion in the x-coordinate involves moving the three masses: the primary and secondary beds, and the gripper with the payload. The required torque can then be calculated while applying a 50% safety factor as follows:

$$T_y = 0.26\ N.m$$

$$T_x = 0.57\ N.m$$

In the case of our model and at our scale, the selected motors are a NEMA 14 that has a holding torque of 0.36 N.m for the y-coordinate motor, while the x-coordinate motor is a NEMA 17 that has a holding toque of 0.88 N.m.

2.2 Gripping system

In this project, a linear actuator is needed to allow the gripping system to move in the Z-coordinate plane. The linear actuator is selected to have a fast response to signals of full extension or retraction. A *Mighty Zap* linear servo motor (shown in Figure 2) with a stroke of 53mm and a top speed of 80mm/s was selected. Two other servo motors are installed on the gripper; one to open and close the griping fingers, while the other is to adjust the gripper's orientation. The two finger gripping system allows for an easier pickup of quadrilateral shaped objects.

To prevent these gripping systems from grabbing an object with excessive force (and to prevent slippage), a force sensing resistor FSR that is embedded within one of the gripper fingers is used, Figure 4. The FSR changes the resistance on its terminals as a result of force applied at its surface.

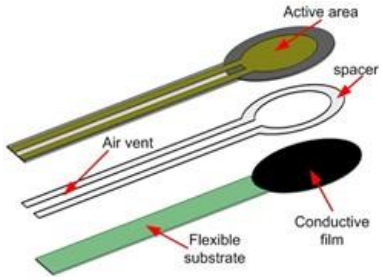

Figure 4: Typical FSR layers arrangement

An FSR works as an open circuit at no load, and when pressure is applied at its active surface, the flexible substrate deforms. This allows the top substrate to be pushed against the bottom substrate, which causes the resistance to drop. If characterized properly, this drop in resistance can be utilized to measure the force applied at the FSR's surface, as shown in Figure 5.

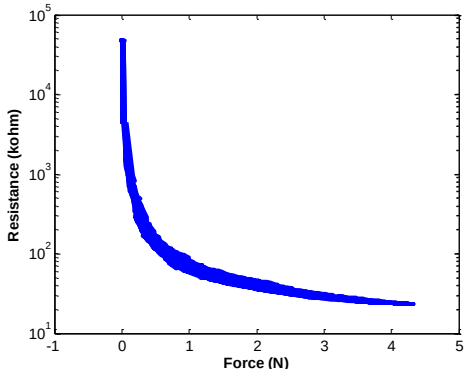

Figure 5: A Typical FSR'S response due to external force

The *FSR 402* (Interlink Electronics) is selected for this application. The FSR exhibits many nonlinearities, the reader is referred to [13] for a detailed analysis of these nonlinearities. However, the FSR has to be calibrated for a specific discrete value for the purpose of determining the appropriate force level. Most of the nonlinearities are associated with the dynamic loading of the FSR; thus, this work will not consider the nonlinear modeling of the FSR.

FSRs are passive resistors that are usually configured in voltage divider circuits for simple resistance-to-voltage conversion, as shown in Figure 6 and Equations 4 and 5.

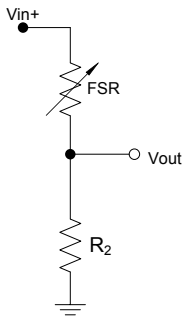

Figure 6: FSR in a voltage divider circuit

To utilize the FSR for force calculation, the output voltage of the voltage divider circuit is first measured then then the resistance of the FSR can be computed, as follows:

$$V_{out} = V_{in}\left(\frac{R_2}{R_2+R_{FSR}}\right) \tag{4}$$

$$R_{FSR} = \frac{V_{in}-V_{out}}{V_{out}} R_2 \tag{5}$$

The FSR is embedded to prevent slippage of the block as well. Figure 7 depicts the free body diagram of the gripper while holding the block.

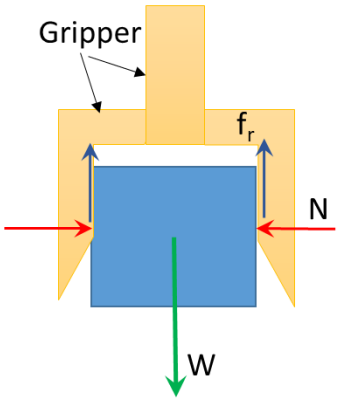

Figure 7: Demonstrating the gripping force of the end effector

To prevent slippage, the friction force (*fr*) should exceed the weight (*W*) of the block, according to the following relation:

$$f_r > W$$

$$N\mu_p > m_b\, g$$

$$N > {}^{m_b\, g}/_{\mu_p}$$

Initially, the voltage divider circuit was constructed with R_2 (100KΩ) resistance in series. A 2N gripping force was applied at the FSR's surface, then the voltage was measured (~3.57V). This voltage will be used as the threshold voltage to guarantee proper gripping force on the objects. The voltage is then compared to a reference (threshold) voltage using an Op-Amp which has its output connected to one of the Raspberry Pi inputs. The system is powered using a 12V, 7A power supply that provides adequate power to run the motors and all sensors. A 12V-5V step down converter is used to power components that require that voltage.

3 OBJECT DETECTION

Before object detection is performed, the camera needs to be calibrated first. The Cartesian plane is labeled with two lines that resemble the X and Y axes. Three blocks of similar shapes and dimensions, but different colors, were prototyped using a 3D printer. The blocks were randomly placed in different locations then a preliminary calculation of the center points of the blocks was established. The camera takes a 2D image that doesn't account for the angle of view, thus the blocks that were farther away from the center of the Cartesian plane have higher errors. To compensate for this error, the calculations to measure the X and Y distance of each block relative to the center point and then adjust the location accordingly. When prompted through a push button that is attached to the Raspberry Pi board, the camera takes an image of the marked Cartesian plane as shown in Figure 8.

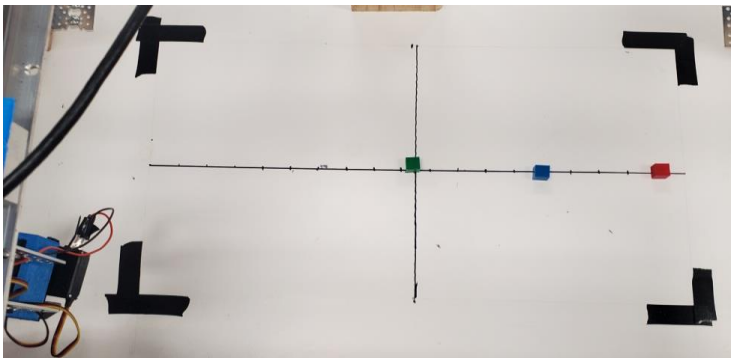

Figure 8: Working space

The image is then transferred to a computer interfaced with the Raspberry Pi to analyze the image and detect the shapes, their location and orientation. A MALAB code (Mathworks. Inc) that uses an image processing toolbox applies three filters to distinguish the three blocks and a

binary image for each block is then produced as shown in Figure 9. These binary images have been processed through the following three filters:

1. A red RGB threshold that leaves only green objects visible as white pixels.
2. A green RGB threshold that leaves only blue objects visible as white pixels.
3. A blue RGB threshold that leaves only red objects visible as white pixels.

This creates three separate binary images isolating the Red, Green, and Blue cubes.

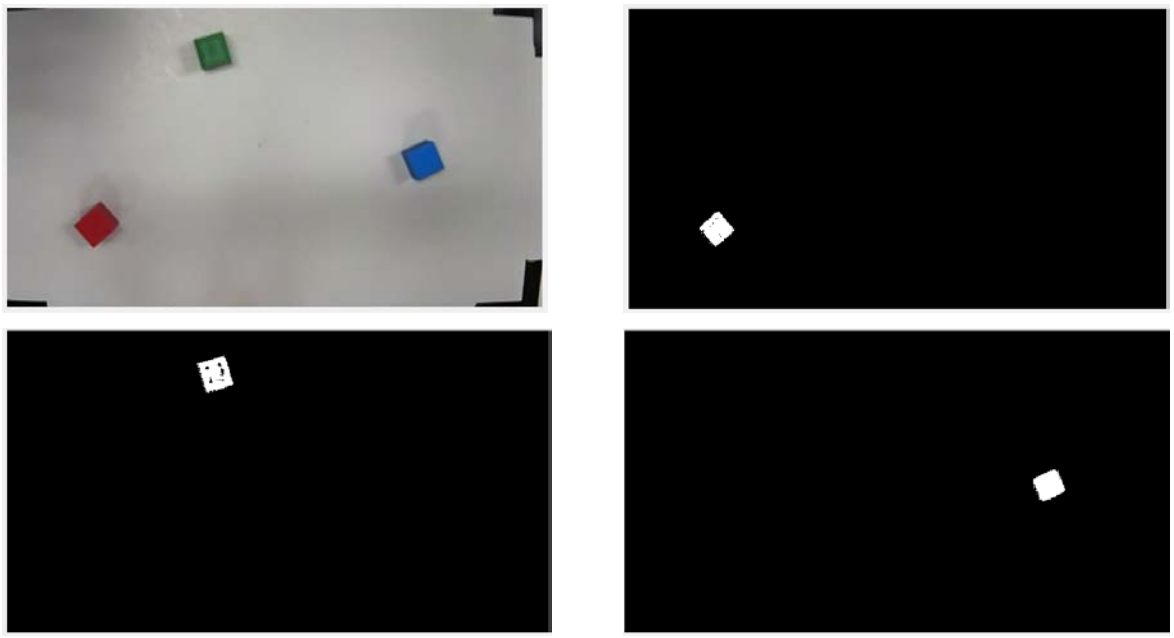

Figure 9: Sample image and RGB filters

Any two diagonal corners can be used to find the center point of the cube. In this case, the coordinates of the first and last non-zero pixels in the horizontal or the vertical planes can be detected. Assuming the first non-zero pixel is at (x_1,y_1) and the final pixel is at (x_2,y_2), then the center point can be calculated as follows:

$$C = \left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right) \tag{6}$$

The rotation angle of the cube can then be calculated as follows:

$$\theta = \tan^{-1}\left(\frac{y_2-y_1}{x_2-x_1}\right) \tag{7}$$

The gripper is attached to a servo-motor that can rotate freely between 0^o and 180^o. Thus, the gripping angle can be adjusted accordingly.

4 PROGRAMMING ALGORITHM

This project uses a Raspberry pi as a microcontroller, sending and receiving signals to and from the Cartesian robot. This is done by constructing specific Python codes that are called by

a master MATLAB code running on a remote host and via a remote connection. A second MATLAB code is then deployed to find the relative location and orientation of each block (the center point of the block and the edge orientation to properly grip the block normal to its edge) while applying the adjustments that were identified earlier according to the c99urrent center points of the blocks. The code also identifies the route that the end effector will take to properly grip each block and place it in its destination. During this step, the trajectory profiles for the X and Y motors are assigned and a vector of directions and their relative timings are passed back to the Raspberry Pi to regulate the motion of the stepper motors through the stepper motor drives to achieve these motions. After each route is performed, the moving bed moves to a home position where two proximity sensors are installed to reset the two axes.

A motion between points A to B prompts the MATLAB code to construct a trajectory that involves both motors. Due to the reliance of the stepper drive on time delays to establish the precise motion for each motor, the motions on X and Y are performed asynchronously. This limitation can be overcome using PLC in future applications.

Typically, a route would appear as follows:

1. Home position
2. Move to Red block, orientate end effector and lower it, then pick it up
3. Move to Red block sorting location, lower end effector and release it to drop the block
4. Move to Green block, orientate end effector and lower it, then pick it up
5. Move to Green block sorting location, lower end effector and release it to drop the block
6. Move to Blue block, orientate end effector and lower it, then pick it up
7. Move to Blue block sorting location, lower end effector and release it to drop the block
8. Move to home position until each proximity sensor is activated to reset the joints

The overall system appears as follows.

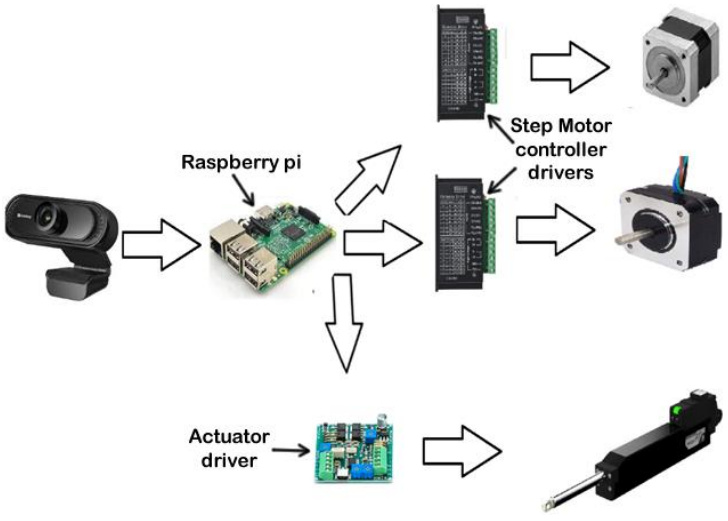

Figure 10: Major Components of the Cartesian Robot

5 CONCLUSION

This work offers a roadmap for the construction of a low cost Cartesian robot for pick and place operation that uses LSPB for joint motion. The stepper motor selection allows the robot to achieve incremental motions that can be computed and correlated to incremental displacements. The motors' motion is achieved through motor drives that receive signals from on-board microcontroller (Raspberry pi). Initially, an overhead camera takes a still image that is passed wirelessly to an interfaced computer host equipped with image processing tools (it utilizes the MATLAB image processing toolbox). The image is then decomposed using RGB filters to create binary images for the different blocks' colors. Each binary image is then analyzed to calculate the center point of the blocks and their orientations. The sorting location for each block is known to the computer. Thus, the computer uses the computed locations of the blocks, the orientation of each block, and the stored sorting locations to calculate each joint's segmented trajectory profiles to achieve these travels. Experimental testing of the robot confirms that these routes are performed successfully and that the accuracy of the picking and placing has been achieved. For the purpose of training students on variety of industrial systems, the proposed setup offers an adequate substitution for the industrial robotic arms at a significantly lower cost with the option of modifying parameters for educational and instructional purposes. Industrial robotic arms move according to a pre-determined path trajectory that follows standard path planning. In addition, they incorporate visions systems and force detection for pick and place applications. The proposed setup adopts these systems and integrates them simultaneously to achieve the required motion. The success of the proposed robotic setup motivates the researchers to replicate it in other robot configurations, such as Cylindrical, Spherical, and SCARA.

ACKNOWLEDGMENT

The authors would like to thank Sidney Bishop, Jeremiah Riewerts, Garrett Parks, and Tyler Huor for their help in conducting the experiments.

REFERENCES

[1] J. J. Craig, *Introduction to Robotics*. New York: Addison-Wesley, 1989.

[2] W. E. Abdul-Lateef A. F. Huayier and N. H. Farhood, "Design of planning trajectory for the planar robot manipulator using linear segments method with parabolic blends (LSPB)," Journal *of Mechanical Engineering Research and Developments*, vol. 44, no. 3, p. 159-171

[3] M. A. Nur Huda, S. H. Susilo and P. M. Adhi, "Implementation of inverse kinematic and trajectory planning on 6-dof robotic arm for straight-flat welding movement," *Journal of Engineering Design and Technology*, vol. 22 no. 1 March 2022; p. 51 – 61

[4] S. Z. Al-khayyt, "Creating through points in linear function with parabolic blends path by optimization method", *Al-Khwarizmi Engineering Journal*, vol. 14, no. 1, March, (2018) p. 77-89

[5] J. Kim, D. Kim and S. Kim, "On-line minimum-time trajectory planning for industrial manipulators," In Proc. 2007 International Conference on Control, Automation and Systems, Seoul, 2007, pp. 36-40, doi: 10.1109/ICCAS.2007.4406875.
[6] M. H. Choi and W. W. Lee, "A force/moment sensor for intuitive robot teaching application," In Proc 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, Korea (South), 2001, pp. 4011-4016 vol.4, doi: 10.1109/ROBOT.2001.933244.
[7] N. J. Ramadhan, N. Lilansa, A. F. Rifa'I and H D. Nguyen, "Pattern recognition based movement control and gripping forces control system on arm robot model using LabVIEW," *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, 13 (2022) p. 1-14.
[8] J. Klimaszewski, D. Janczak and P. Piorun, "Tactile robotic skin with pressure direction detection", *Sensors*, 2019, 19, 4697; doi:10.3390/s19214697
[9] A. S. Sadun, J. Jalani, J. A. Sukor, "Force Sensing Resistor (FSR): a brief overview and the low-cost sensor for active compliance control," In Proc. First International Workshop on Pattern Recognition; 1001112 (2016) https://doi.org/10.1117/12.2242950
[10] A. Saboukhi, M. R. Gorji, E. Amirpour, M. Savabi, R. Fesharakifard, H. Ghafarirad and S. M. Rezaei, "Design -and experimental analysis of a force sensitive gripper for safe robot applications," In Proc. 2019 7th International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 2019, pp. 345-351, doi: 10.1109/ICRoM48714.2019.9071887.
[11] R. Kumar, U. Mehta and P. Chand, "A low cost linear force feedback control system for a two-fingered parallel configuration gripper", *Procedia Computer Science*, vol. 105, 2017, p. 264-269.
[12] M. Y. Saadeh, M. B. Trabia, "Identification of a force-sensing resistor for tactile applications", *Journal of Intelligent Material Systems and Structures*. 2013; vol. 24, no. (7), p. 813-827. doi:10.1177/1045389X12463462
[13] M. Y. Saadeh, T. D. Carambat and A. M. Arrieta, "Evaluating and modeling force sensing resistors for low force applications", In Proc. ASME 2017 Conference on Smart Materials, Adaptive Structures and Intelligent Systems, No. 3703, Snowbird, Utah, USA. September 18–20, 2017.