A

PROJECT REPORT ON

"Pick & Place Robot"

Submitted by,

Durgesh Kolte	**T196003**
Rutuja Patil	**T199069**
Tarun Shrivastava	**T197045**

Guided by,

Prof. Shridhar Khandekar

A Report submitted to MIT Academy of Engineering, Alandi (D.) submitted in partial fulfilment of the requirement for Sixth Semester of BACHELOR OF TECHNOLOGY in School of Electrical Engineering.

SCHOOL OF ELECTRICAL ENGINEERING

MIT Academy of Engineering

Dehu Phata, Alandi (D.)

Pune - 412105, Maharashtra (India)

2019-20

MIT | Academy of Engineering	**School of Electrical Engineering** **(Accredited by NBA, ISO 9001:2008 Certified)**

This is certify that,

Durgesh Kolte	**T196003**
Rutuja Patil	**T199069**
Tarun Shrivastava	**T197045**

of TY B.Tech. have submitted a Report on,

Pick & Place Robot

The said work is completed by putting the requirement of hours as per prescribed curriculum during the academic year 2019 - 20. The report is submitted in the partial fulfilment of the requirements for the **Mini Project** in the Sixth Semester of Degree of Engineering in –School of Electrical Engineering, MIT Academy of Engineering.

SCHOOL OF ELECTRICAL ENGINEERING

MIT ACADEMY OF ENGINEERING, ALANDI(D), PUNE-412105

MAHARASHTRA (INDIA)

MAY, 2020

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

It is hereby certified that the work which is being presented in the TY B.Tech. Mini Project Report entitled **"*Pick & Place Robot*",** in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Electronics Engineering** and submitted to the **School of Electrical Engineering of MIT Academy of Engineering, Alandi(D), Pune, Affiliated to Savitribai Phule Pune University (SPPU), Pune** is an authentic record of work carried out during an Academic Year 2019-2020, under the supervision of Prof. **Shridhar Khandekar, School of Electrical Engineering.**

Durgesh Kolte	**0120170036**	**T196003**
Rutuja Patil	**0120170323**	**T199069**
Tarun Shrivastava	**0120170267**	**T197045**

Date: 27 / 04 / 2020

Signature of Project Advisor
Project Adviser
School of Electrical Engineering,
MIT Academy of Engineering, Alandi(D), Pune

Signature of Dean
Dean
School of Electrical Engineering,
MIT Academy of Engineering, Alandi(D), Pune

(STAMP/SEAL)

Signature of Internal examiner/s

Name………………………………

Affiliation…………………………

Signature of External examiner/s

Name………………………………

Affiliation…………………………

CONTENTS

ACKNOWLEDGEMENTS

We take this opportunity to record our profound gratitude and indebtedness to Prof. Shridhar Khandekar, School of Electrical Engineering for his inspiring guidance, valuable advises, constant encouragement and untiring supervision throughout our project work.

Finally, we would like to acknowledge and express our special thanks to our family, friends and classmates for their patience, encouragement, support they have made during the period of this work.

Durgesh Kolte	**T196003**
Rutuja Patil	**T199069**
Tarun Shrivastava	**T197045**

ABSTRACT

Picking and placing of goods is one of the most repetitive task in industries. Along with repetitiveness, this task requires certain amount of precision and also adaptability as many times the destination of goods might change. For tackling such a task a Pick and place robots are designed which are extensively used in many medium scale as well as large scale industries. But this technology is limited to assembly line only, we can use the same technology with the help of few major changes to make it useful in various aspects such as delivering goods from the assembly line to storage room, displacing goods from storage room to shipping bay and so on. We did a project to realize this and the following document is the project report of the same. **Pick & Place Robot (PPR)** is a project based on computer technology and sensor integration being used in a variety of applications that identifies numerous objects using digital images or external Cameras. It has various applications such as Pick & Place, Facial Recognition, security etc.

The **PPR** is capable of identifying or verifying goods carrier from a digital image or a video frame from a video source, then process it to compute required actions respectively. With the help of Absolute Odometry it was made possible to loco mote the robot into the industry from one place to another. There are multiple methods in which this system could be done like manually or autonomously, but in general, they work manually and this needs to be replaced as autonomous systems are more accurate as compared to manual and less costly as well as require less human interaction. In this project we have modelled our system to work as in it can do its work without human interference.

LIST OF FIGURES

1. INTRODUCTION

1.1 Motivation:

There are three major key aspects that are looked into by industries nowadays which are mainly cost, time and accuracy.

So, whenever we talk about automation and robotics the main purpose of the company is to make maximum profit in less amount of time with minimum workforce and errors.

So, this system of picking and placing is designed such that it can work manually or autonomously in industries with minimum to no human interference for transportation of goods carriers within the industry campus and in less time and errors.

1.2 Objective:

1. To design a system that can be used in industries for picking and placing goods (coloured balls) carrier.
2. To design a Mechanical CAD of the structure of Robot using SolidWorks.
3. To mould an Algorithm or Flowchart required for the implementation using Absolute Odometry.
4. Identify and locate the coloured balls in a confined place using NI Vision.

1.3 Problem Statement:

To Design a system that can be used for picking and placing goods carrier in industries with the help of Computer Vision and Absolute Odometry.

2. LITERATURE SURVEY

There are three major key aspects that are looked into by industries nowadays which are mainly cost, time and accuracy. Whenever we talk about automation and robotics the main purpose of the company is to make maximum profit in less amount of time with minimum workforce and errors.

So, this system of picking and placing is designed such that it can work manually or autonomously in industries with minimum to no human interference for transportation of goods carriers within the industry campus and in less time and errors. In a IEEE conference in 2012, Mr. Sajjad Hussain presented a paper on Automation of mobile pick and place robots during which he stated that after going through all the data he gathered from his research on best/optimum motors for such a product, he came to a conclusion that using PMDC motors would be efficient economically and power-wise as well as it will be easier to control [1].

Designing the manipulator such that we have DOF that we require not less which will act as a constraint or more which will just be difficult to control. The more simplistic the design the better. Also, the links and joints should be made up of material which is safe enough for working in industrial environment.

"Different manipulator configurations are available as Rectangular Cylindrical, Spherical, SCARA, Revolute and Horizontal Jointed. The robot Arm determines the position of the wrist in 3D space [2]. So having a design based on and planed according to the workspace required for the problem makes it easier to achieve your goal." Stated Mr. Yogesh Ugale while presenting his paper "Design and Analysis of pick and place robot" which was presented in an IJSRSET conference held in 2018.

Also, the design of the manipulator cannot be a universal thing, it will wary from application to application. The manipulator can be manually controlled or autonomous. Considering the possibility of controlling the manipulator and pick and place functions using LabVIEW will make the computations easier [3]. Both the ways of controlling have their merits over the other since manual control will be easy to implement the autonomous control will be highly precise and would not require an operator for the application but also removes the final decision-making chance from human.

During the presentation of his paper "Review on Industrial pick and place robots" at an IJIERT conference in 2018, Mr. Pawan Shinde explained the same point by taking an example of a manipulator built for doing spot welding.

In a IJSRSET conference held in 2017 Ms. Pratiksha Andhare presented her paper "Pick and Place industrial robot using computer vision", during which she stated that extracting useful information from an image is a tedious task and hence it should be divide into stages such as scene analysis which includes identifying the object and the space where the object is along with the position of object and other stage would be planning process of gripper controller accordingly. Only after successfully doing both we can say that we have achieved a computer vision based system.

3. SYSTEM ARCHITECTURE

3.1 Block Diagram:

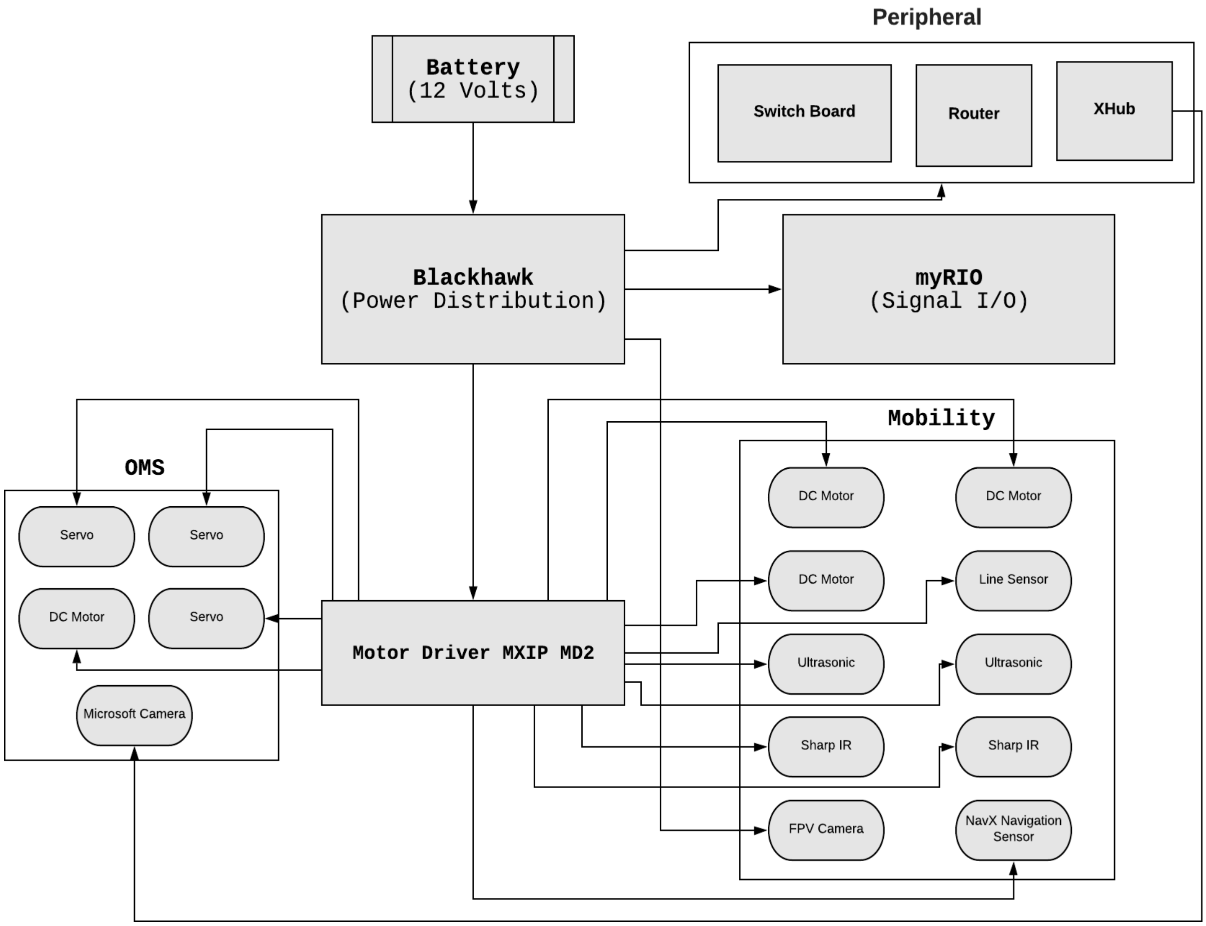

Fig. No. 3.1 Block Diagram

As soon as the power is switched on, the controller will start taking input from sensors to identify its position which it will home as per the readings of sensors. If the manipulator is not in home position which it will identify by the help of servo driver, controller will send command to servo driver to set home position of all servos controlling the arm.

Now after the controller has set itself in home position, it will send signal to the motor drivers so as to move motors and start its task, throughout the task the controller will keep tracking the readings it will get from encoders coupled with the motor to make sure it is in control over the number of rotations of the shaft of motor so that it can control the position of the robot. Also during this motion, the controller will read all the essential sensors which are coded to keep track of the distance of robot with any obstacle it faces as collision might damage the robot and hence the sensors will be set on a threshold value below which the controller will consider as danger and stop its motors to stop and avoid collisions.

After reaching the point of loading, the controller will read the image taken by the camera and extract the position of the balls that it is supposed to pick. Now as per the co-ordinates, the controller will command the motor driver unit to adjust the motors to an appropriate position to pick the balls easily. Again the camera will take image, which controller will read to take feedback whether the ball was picked successfully or not.

3.2 Mechanical Drawing:

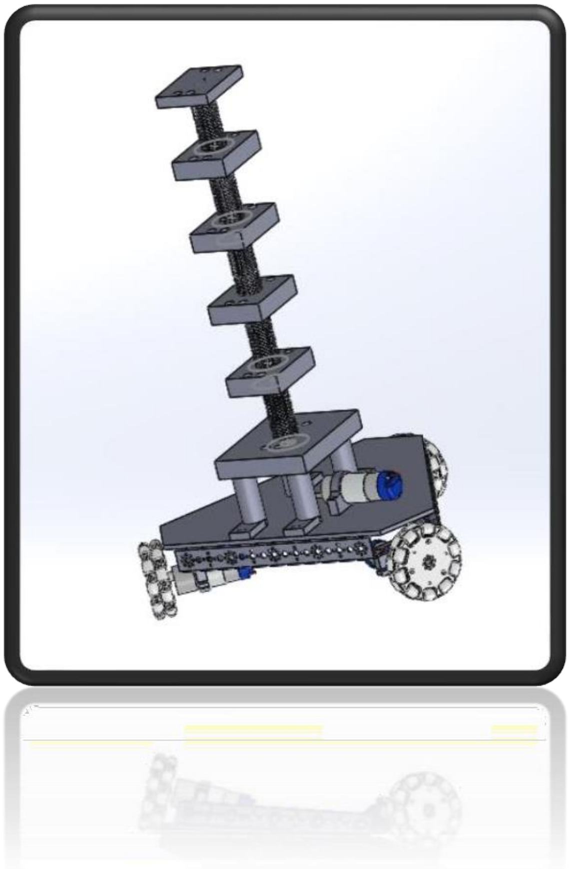

Fig. No. 3.2 CAD Model of Gripper

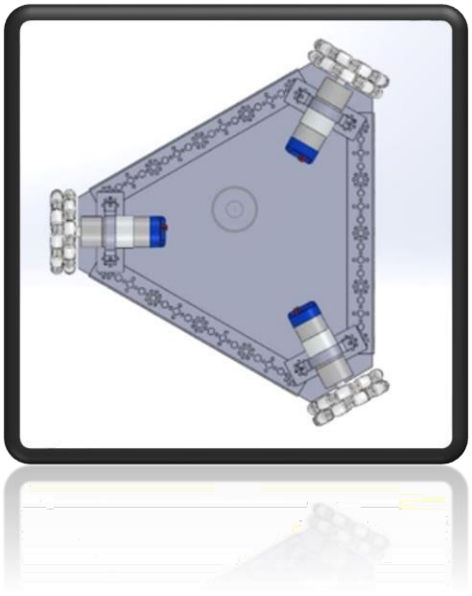

Fig. No. 3.3 Base CAD(Three-Wheel)

3.3 Sequence Diagram:

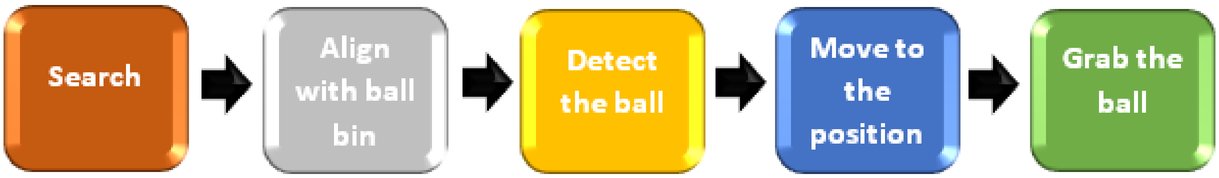

Fig. No. 3.4 Task Sequence Diagram

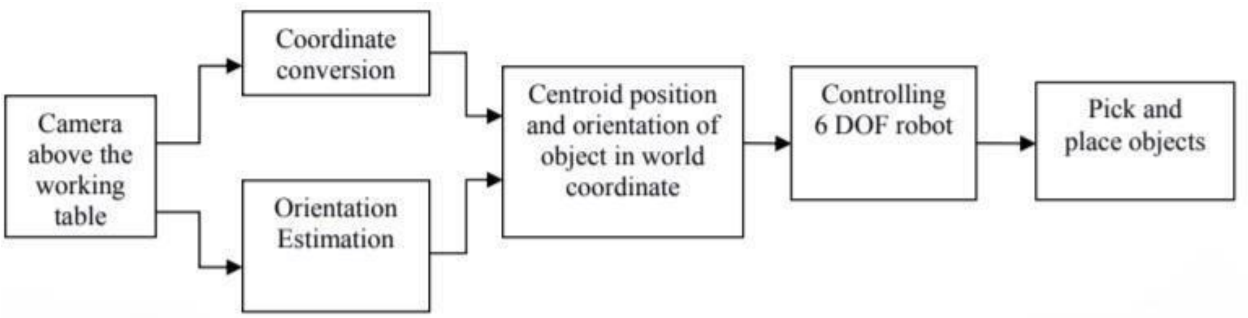

Fig. No. 3.5 Execution Sequence Diagram

3.4 Activity Diagram:

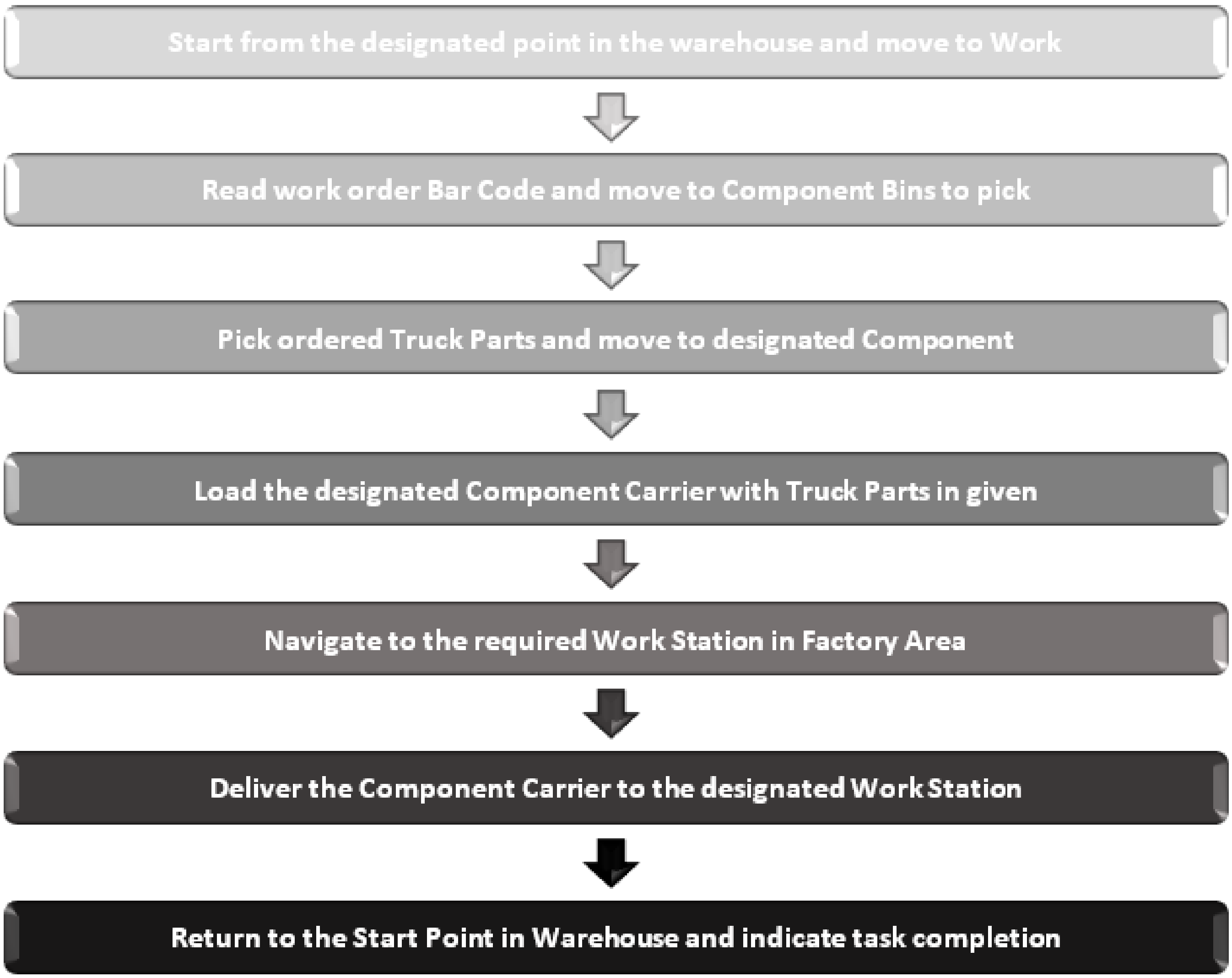

Fig. No. 3.6 Activity Diagram

3.5 Related Mathematical Modelling:

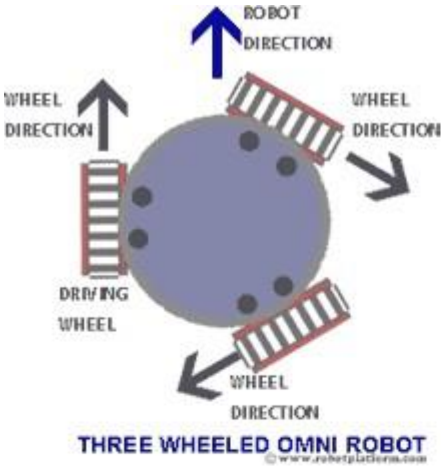

Fig. No. 3.7 Three-Wheel Omni Base

The **Kinematic Model of Base** is as follows:

Each motor speed is computed as follows:

$$\mathbf{M_1 = -0.33 * L_X + 0.58 * L_Y + 0.33 * R_Z}$$

$$\mathbf{M_2 = -0.33 * L_X - 0.58 * L_Y + 0.33 * R_Z}$$

$$\mathbf{M_3 = +0 * L_X + 0.67 * L_Y + 0.33 * R_Z}$$

Where,

M_i: Final Drive Motor PWM

L_X: Linear Speed in X-axis

L_Y: Linear Speed in Y-axis

R_Z: Velocity of Rotation about Z-axis

Note: If the motor **PWM** (M1, M2, M3, M4) is evaluated to be negative, it denotes a change in direction of rotation of the motor shaft at the computed magnitude.

3.6 Hardware and Software Requirements:

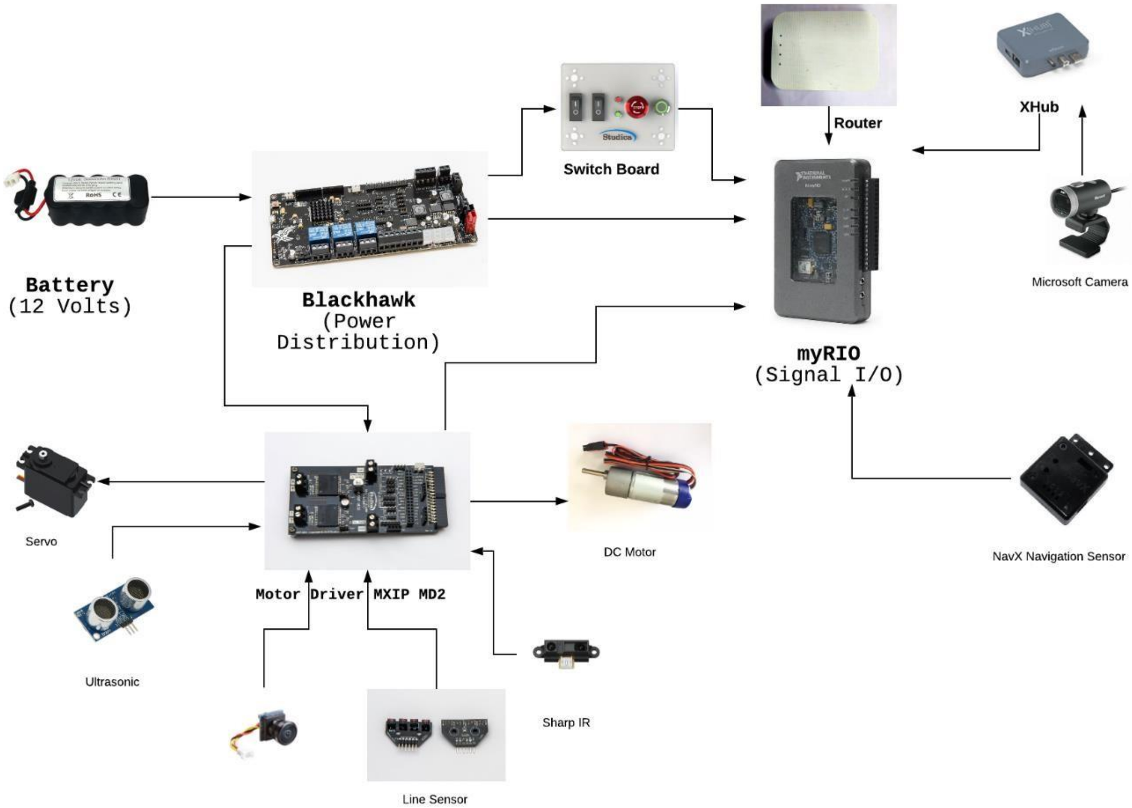

Fig. No. 3.8 Hardware Diagram

Software Required:

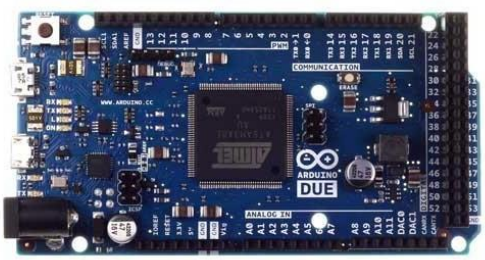

Fig. No. 3.9 LabVIEW **Fig. No. 3.10 Arduino Due** **Fig. No. 3.11 Arduino IDE**

4. IMPLEMENTATION

4.1 Algorithm:

Coordinate Detection Algorithm:

Step 1: Open NI Vision Assistant.

Step 2: Add sample images to be processed through Open Image option.

Step 3: Use Method 1 or Method 2 to get Coordinates of Balls.

Method 1: Circle Detection

- Select image
- Use Masking Block under Image column to mask the unwanted part of image
- Under Color column use Plane Extraction Block
- From Histogram Graph in Image column add the Threshold Block
- In Binary column select Advanced Morphology to Fill minute Holes
- And finally, from same column select Circle Detection

Method 2: Template Imaging

- Select image
- Select Color Detection under Color column
- Select Create template if you want to create template from current image
- Otherwise if already template is available then select Load from File option
- Set the Saturation Threshold and Done.

Step 4: From the above methods Centres of Balls available in image will be displayed

Step 5: With setting respective Radius Size or Matching Score select the desired Ball

Fig. No. 4.1 Template Images

Robot Locomotion Algorithm:

Coordinate Extraction

Step 1: From the Vision Assistant extract the coordinates of the Balls

Step 2: Map the pixels range of coordinates using Operators Block

Colour Palette Display

Step 3: Add three colour palettes in an array

Step 4: Make a switch case corresponding to Barcode Correspondence figure

Rotation

Step 5: Taking difference of the Current angular position and Desired rotation

Holonomic Drive

Step 6: Using Force Equations to find pwm coefficients on each wheel

Step 7: Writing these equations in the Numerical Node Block

Servo Positioning

Step 8: Create switch case which has various height for placing

Step 9: These cases are connected through Select Case Block and Boolean buttons

Indicators

Step 10: Finally, all these outputs from above blocks are connected to Indicators

For localization, One encoder in X and Y axes each are hinged freely for getting its coordinates in the Cartesian plane. For getting those coordinates, **Optical Incremental Quadrature Encoders** of 600 PPR, coupled with Omni wheels of diameter (d) 58 mm, rolling freely on the ground. Hence, to cover **'D' mm** distance, the encoder counts required would be:

X = 182.12 / (2400 * D)

Where, circumference of the wheel is: **3.14 * d = 182.12 mm** and **2400 encoder counts** are recorded per rotation. The velocity of the robots would be manipulated in the **PID Control** block with ‘X’ encoder counts as input.

To avoid the minor offset due to slippage or localization error, a pair of **limit switches** on either side of the robot is mounted to align at certain checkpoints for **Absolute Odometry.**

4.2 Flowchart:

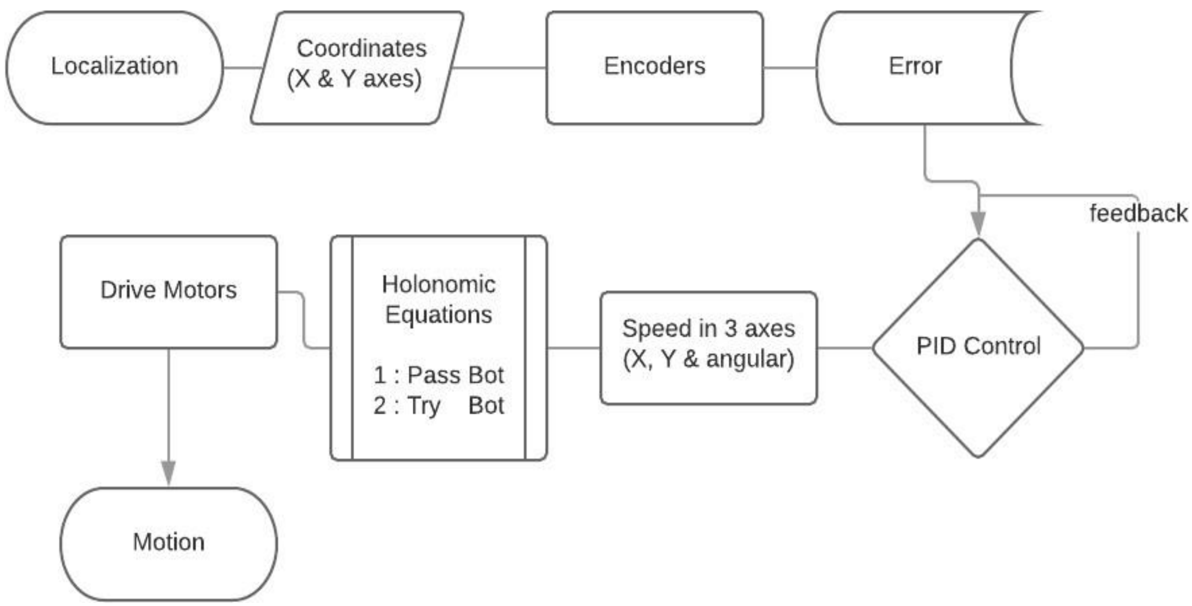

Fig. No. 4.2 Flowchart

4.3 Results:

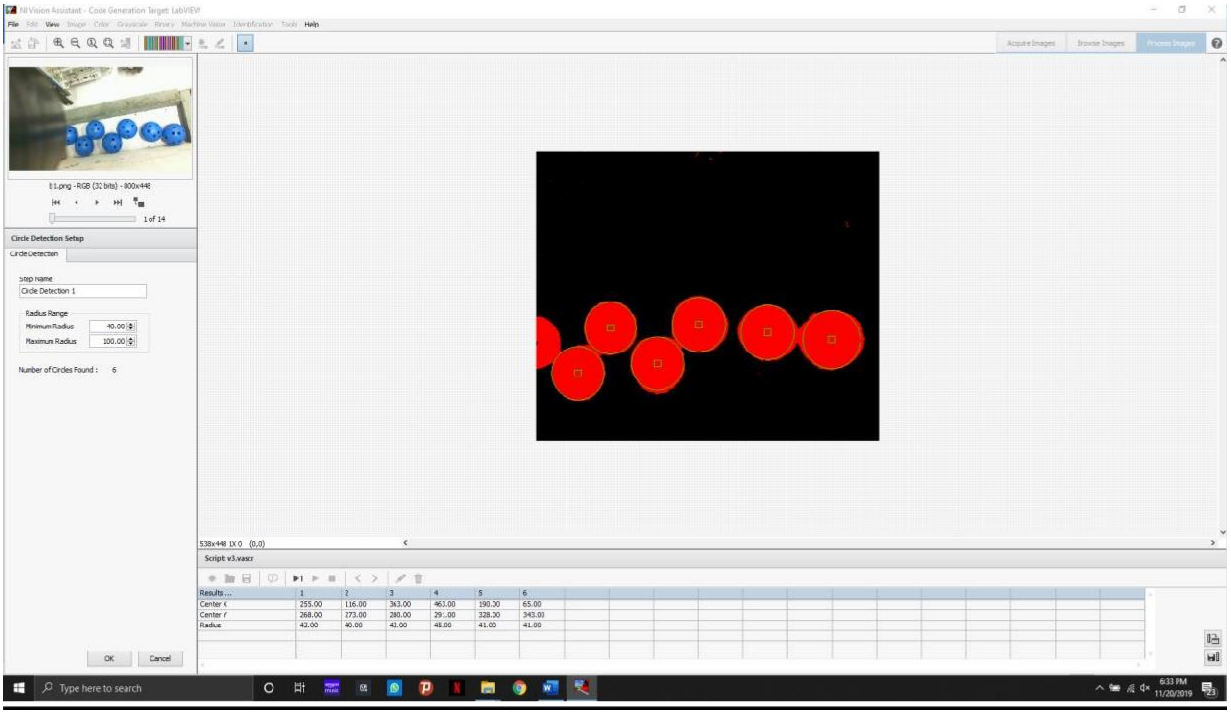

Fig. No. 4.3 Coordinates by Method 1

This is the final output of position extraction of balls from the image taken by the camera mounted on top of the gripper. Here we can see the image was first masked to remove the unwanted area of image using the Masking block. Then using the plane extraction block we got the colour of the balls. Then we used histogram block to add a threshold so that we do read any other objects such as small pebbles. Next we used advanced morphology block to fill any holes below threshold levels as presence of such holes may cause the program to think that there are two balls separated by a hole or it might think that the hole is a third ball which will cause error. Finally circle detect block is used to identify the positions of all the balls present.

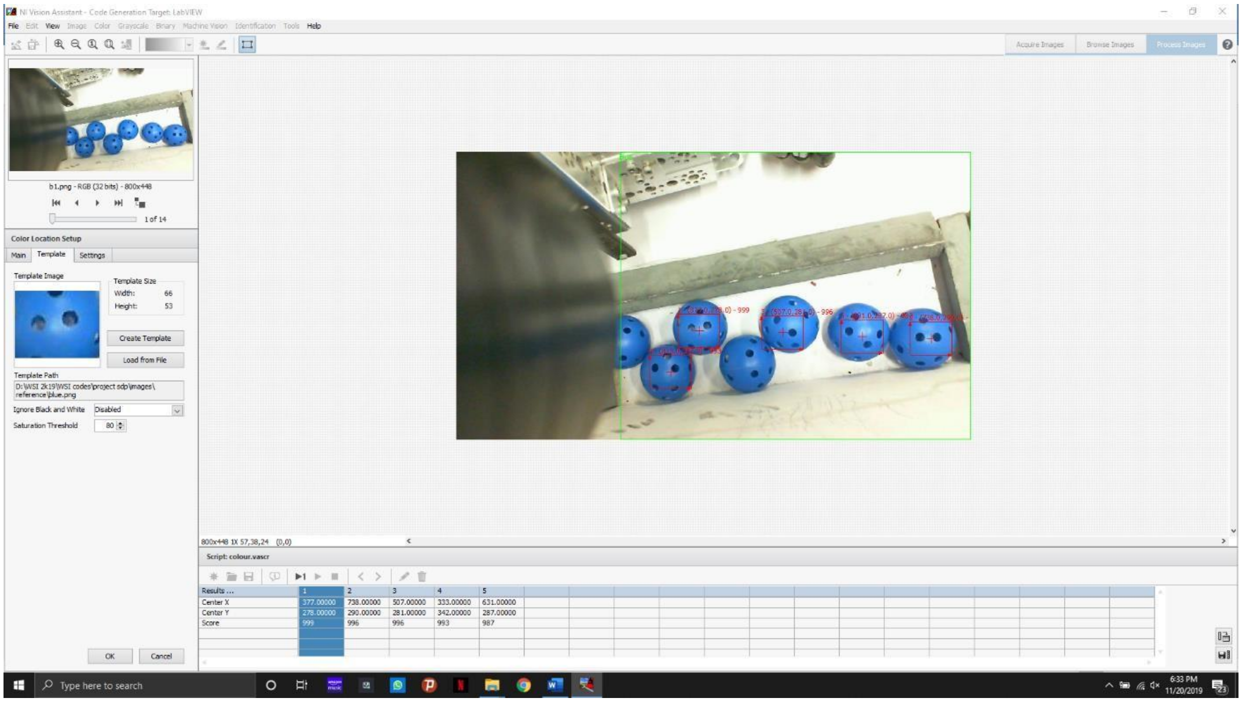

Fig. No. 4.4 Coordinates by Method 2

This is the results obtained by using the second method for obtaining the coordinates of the balls. First we took colour detection block which helps us identify the colour of the ball. Then we use the create template block from the template pallet to create a template of the colour of ball we choose. Now we use the saturation threshold block for identifying all objects of the same colour as our template, hence we get the position of all the balls of that particular colour that we made the template of. The coordinates will be w.r.t camera.

5. FUTURE SCOPE

Marching into the world of automated industry, we were able to device a system that could be efficiently used in the fast growing industry. Our project helps remove most of the drawbacks which were faced earlier such as a need of a manual interference for motion and working of the robot. There are still some drawbacks that can be eliminated. There is a need to have a preprogrammed command set for this robot, if we are able to control it with a voice command it will like talking to a person who has minimal movements and can bring you balls. Now, due to the construction of a minimalistic manipulator, the robot has to load the balls on itself and then unload it using a carrier. If we develop a higher DOF manipulator, the time required to complete the task will reduce to half because we will directly be able to load and unload balls without needing it to go in a carrier and then moving the carrier. This will be a significant increase in both efficiency and complexity but this will also unlock many possibilities such as personal domestic assistant. We could have a robot which will bring things we ask of it from other rooms to us on a single command.

Also there has to be changes in the algorithm of movement of the robot. Currently the code is working serially which means the controller has to process the complete code for checking all the instructions before executing a function. If somehow we are able to achieve a more efficient way to process so that the controller can read the readings of encoder as well as the proximity sensors parallely, this would increase speed by great proportions.

6. CONCLUSION

Computer Vision:

- USB Camera which is mounted over the gripper of Pick and Place Robot
- Real time image is captured
- This image is then processed using NIs Vision Assistant
- Vision Assistant gives the exact coordinates of the object which in our case are various coloured balls (Red, Green, Blue and Yellow)

Absolute Odometry:

- Odometry is the use of data from motion sensors to estimate change in position over time.
- It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location.
- This method is sensitive to errors due to the integration of velocity measurements over time to give position estimates.
- Rapid and accurate data collection, instrument calibration, and processing are required in most cases for odometry to be used effectively.
- **Advantages** of having automated system**:**
  - **High Accuracy**
  - **Cost Effective**
  - **Space Efficient**
  - **Increased Safety**

PSEUDO CODE

```
void single_align(float Dd, float baseSpeed, int pattern) {
  float offset = 15.0;
  offset = (offset * 2400.0) / circ;
  Dd = (Dd * 2400.0) / circ;
  float kp = 0.0;

  if (pattern == 3) {
    s11 = LF;
    s12 = LB;
  }
  else if (pattern == 4) {
    s11 = BL;
    s12 = BR;
  }
  else if (pattern == 5) {
    s11 = RB;
    s12 = RF;
  }

  if (pattern == 3 || pattern == 5) {
    kp = 1 / abs(Dd - encoderY.encoderTicks());
    while (abs(Dd - encoderY.encoderTicks()) > offset) {
      if (pattern == 3) {
        Lx = -ppwm;
      }
      else if (pattern == 5) {
        Lx = ppwm;
      }

      Ly = kp * baseSpeed * (Dd - encoderY.encoderTicks());

      Rx = -algn(s11) + algn(s12);

      if (Ly > 0) {
        Ly = constrain(Ly, 1.5 * ppwm, baseSpeed);
      }
      else {
        Ly = constrain(Ly, -baseSpeed, -1.5 * ppwm);
      }

      OmniX.Cross(Lx, Ly, Rx);
      //      print_speed();
    }
  }
  else if (pattern == 4) {
```

```
    kp = 1 / abs(Dd - encoderX.encoderTicks());
    while (abs(Dd - encoderX.encoderTicks()) > offset) {

      Lx = kp * baseSpeed * (Dd - encoderX.encoderTicks());

      Ly = -ppwm;
      Rx = algn(s12) - algn(s11);

      if (Lx > 0) {
        Lx = constrain(Lx, 1.5 * ppwm, baseSpeed);
      }
      else {
        Lx = constrain(Lx, -baseSpeed, -1.5 * ppwm);
      }

      OmniX.Cross(Lx, Ly, Rx);
      //          print_speed();
    }
  }

  OmniX.Cross(0, 0, 0);
}

void dum_enc(float Xx, float Yy, float bX, float bY, int offset, float kpX, float kpY) {

  encX = encoderX.encoderTicks();
  encY = encoderY.encoderTicks();

  if (abs(Xx - encoderX.encoderTicks()) > offset) {
    Lx = kpX * bX * (Xx - encX);

    if (Lx > 0) {
      Lx = constrain(Lx, 1.5 * ppwm, bX);
    }
    else {
      Lx = constrain(Lx, -bX, -1.5 * ppwm);
    }
  }
  else {
    Lx = 0;
  }

  if (abs(Yy - encoderY.encoderTicks()) > offset) {
    Ly = kpY * bY * (Yy - encY);

    if (Ly > 0) {
      Ly = constrain(Ly, 1.5 * ppwm, bY);
    }
    else {
      Ly = constrain(Ly, -bY, -1.5 * ppwm);
```

```
    }
  }
  else {
    Ly = 0;
  }

  OmniX.Cross(Lx, Ly, Rx);
  //    print_speed();
}

void traverse(float Xx, float Yy, float baseSpeed, int offset) {
  offset = (offset * 2400.0) / circ;
  Xx = (Xx * 2400.0) / circ;
  Yy = (Yy * 2400.0) / circ;
  float kpX = 1 / abs(Xx - encoderX.encoderTicks());
  float kpY = 1 / abs(Yy - encoderY.encoderTicks());
  float bX = 0.0, bY = 0.0;
  if (kpX > kpY) {
    bX = (kpY * baseSpeed) / kpX;
    bY = baseSpeed;
  }
  else {
    bX = baseSpeed;
    bY = (kpX * baseSpeed) / kpY;
  }

  while (abs(Xx - encoderX.encoderTicks()) > offset || abs(Yy - encoderY.encoderTicks()) > offset) {
    dum_enc(Xx, Yy, bX, bY, offset, kpX, kpY);
  }
  OmniX.Cross(0, 0, 0);
}

void dum_limit(int pattern) {
  Lx = 0, Ly = 0, Rx = 0;

  if (pattern == 1) {
    Lx = - algn(s11) - algn(s12);
    Ly = - algn(s21) - algn(s22);
    Rx = - algn(s11) + algn(s12) - algn(s21) + algn(s22);
  }
  else if (pattern == 2) {
    Lx = algn(s11) + algn(s12);
    Ly = - algn(s21) - algn(s22);
    Rx = algn(s11) - algn(s12) - algn(s21) + algn(s22);
  }

  OmniX.Cross(Lx, Ly, Rx);
  // print_speed();
}
```

```
void limit_align(int pattern) {
  if (pattern == 1) {
    s11 = LF;
    s12 = LB;
    s21 = BL;
    s22 = BR;
  }
  else if (pattern = 2) {
    s11 = RF;
    s12 = RB;
    s21 = BL;
    s22 = BR;
  }

  while ((digitalRead(s11) && digitalRead(s12)) || (digitalRead(s21) && digitalRead(s22))) {
    dum_limit(pattern);
  }

  int now = millis();
  while (millis() - now < 500) {
    if ((digitalRead(s11) && digitalRead(s12)) || (digitalRead(s21) && digitalRead(s22))) {
      dum_limit(pattern);
    }
  }

  OmniX.Cross(0, 0, 0);
  encoderX.encoderZero();
  encoderY.encoderZero();
  delay(250);
}

int algn(int LS) {
  if (digitalRead(LS)) {
    return ppwm;
  }
  else {
    return 0;
  }
}
```

REFERENCES

1. Mir Sajjad Hussain Talpur, Murtaza Hussain Shaikh. Automation of Mobile Pick and Place Robotic System for Small food Industry. 978-1-4577-1139- 8/12/$31.00 ©2012 IEEE
2. Yogesh G.Ugale, Gopal R. Ugale, Akash B. Dhote, Sagar S.Dawange, Nikhil G.Vase , Shrutika R.Ghodke, Prof.Pramod S.Wankhede. Design & Analysis of Pick and Place Robot. © 2018 IJSRSET

   | Volume 4 | Issue 4 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099Themed Section : Engineering and Technology
3. Mr.Pawan Shivaji Shinde, Mr.Aaditya Manoj Sonawane, Mr. Kumar Sunjay Gaikwad, Mr. Omkar Vijay Pawar, Prof .Suhas Gudhate. Review paper on industrial pick & place robotic arm. International Journal of Innovations in Engineering Research and Technology [IJIERT], ISSN: 2394-3696 Conference Proceedings of i - Mechanical Engineering Students Conference 2018 (i – MESCON 18) 28th December, 2018.
4. Pratiksha Andhare, Sayali Rawat. Pick and Place Industrial Robot Controller with Computer Vision.
5. D Scaramuzza, F Fraundorfer . Visual Odometry.IEEE robotics & automation magazine, 2011.Cited by 1106 Related articles All 18 versions
6. J. Zhang and S. Singh. Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast. IEEE Intl. Conf. on Robotics and Automation (ICRA). Seattle, WA, May 2015. (PDF) (Video)
7. http://download.ni.com/pub/devzone/epd/mobile_robotics_experiments.pdf