International Journal of Swarm Intelligence and Evolutionary Computation

Patil, Int J Swarm Intel Evol Comput 2017, 6:1
DOI: 10.4172/2090-4908.1000154

Research Article

Open Access

Robot Manipulator Control Using PLC with Position Based and Image Based Algorithm

Madhav D Patil*

Department of Electrical and Computer Engineering, University of Bridgeport, USA

Abstract

Programmable Logic Controller is making big role in automation and robotics. This paper described about the design and implementation of Artificial intelligence algorithms for controlling the robotic arm or manipulator hardware platform using Programmable Logic Controller (PLC). The controlled robot is 5 degrees of freedom (DOF) manipulator with a closed kinematic chain, designed for high-performance pick and place applications. The control software is fully developed on a commercial PLC system, using its standard programming tools and the multi-tasking features of its operating system. In particular, the paper analyse in detail the drawbacks and the advantages related to the choice of standard PLCs in this kind of applications, compared to the much common choice of specialized hardware or industrial personal computers, with particular emphasis on the computational performances obtained with the proposed control architecture.

Keywords: Artificial intelligence; Vision system; Programmable logic controller; Robot manipulator/Arm; Industrial automation

Introduction

Robot manipulators are largely used in packaging plants, especially for pick and place operations and box filling. In particular, packaging of small food products, like cookies or candies, requires high performance robots with short cycle times and precise motion control, even if their workspace is relatively small [1]. Such performances can be obtained by means of lightweight parallel-drive or delta-like kinematics, whose advantages in terms of reduced moving masses and inertias are well-known. The robots include a vision system, in order to identify and localize products transported in unordered manner and in a large number on the conveyors. The robot controller is actually a quite closed system that customers are not able to extend without specific support from the manufacturer. Since the integration of a robotic system with additional, application-specific, tools and features often requires the development of software or hardware closely matched with the basic robot motion control system, customers often search for openness of robot controllers. In fact, several open architectures for robot control can be mentioned, either developed within academic projects or from commercial robot manufacturer [2-4]. In most cases, these architectures take advantage of hardware platforms based on Personal Computers (PC). On one hand, the use of PCs for robot control reduces software development costs, thanks to high level languages and well-designed Integrated Development Environments (IDE). On the other hand, standard PCs do not have the reliability required by harsh industrial environments, so that the use of more protective cases and more robust electronic components actually increase the final cost of PC based robot controllers.

Programmable Logic Controllers (PLCs) are instead a kind of control devices playing a dominant role in industrial automation, thanks to their robustness, reasonable cost and ease of use [1]. Electronic components inside PLCs are selected for their reliability and solid presence on the market, rather than for their performance. For this reasons, PLCs are commonly considered as low-level systems, whose main purpose is to control simple Boolean signals (i.e. discrete control and sequencing) and to supervise the safety and integrity of plants and operators. However, modern PLCs have most of the times sufficient computational bandwidth to execute also complex mathematical calculations and their programming languages (i.e. IEC 61131-3) allow to easily implement any kind of algorithm that can be used in robot control [5].

Moreover, even if most PLCs support only their proprietary IDE, with that specific IDE it is always possible to upload the user software from the controller and modify it, which is not possible on PC-based systems whose software, is a compiled executable [6]. Therefore, PLC programs can be updated and extended by end-users even without the original source code, which makes PLCs “open” systems. Despite this feature, PLC-based control systems for robot manipulators are quite rare and, in general, limited to simple Cartesian or gantry-like structures [7].

A robot arm is the combination of links and joints in the form of a chain with one end is fixed while the other end is free. The joints are either prismatic or revolute, driven by actuators. In order to move the free end, also called the end effector, along a certain path, most, if not all, of the joints are to be moved [8]. In doing so it is necessary to solve the inverse kinematics equation. In the case of redundant manipulator, inverse kinematics is much more difficult when compared to a non-redundant manipulator whose kinematics is not so complicated [7].

There are traditional methods such as algebraic solutions, geometric solutions and iterative solutions in order to solve the inverse kinematics problem. However, these methods are time-consuming and suffer from numerical problems. Recently, the particle swarm optimization (PSO) has been successfully applied to various optimization problems. This new optimization algorithm combines social psychology principles in socio-cognition human agents and evolutionary computations [9].

In this paper, we aim to show that the capabilities of a commercial

***Corresponding author:** Patil MD, Department of Electrical and Computer Engineering, University of Bridgeport, Bridgeport, CT, USA, Tel: 203-572-4551; E-mail: mpatil@my.bridgeport.edu

Received April 07, 2017; **Accepted** April 19, 2017; **Published** April 26, 2017

Citation: Patil MD (2017) Robot Manipulator Control Using PLC with Position Based and Image Based Algorithm. Int J Swarm Intel Evol Comput 6: 154. doi: 10.4172/2090-4908.1000154

PLC and its multi-tasking operating system are sufficiently powerful to implement a robot control system including inverse kinematics, online trajectory planning. The integration of vision systems with Robot manipulator and PLC used for implementing Imaged Based Intelligence algorithm also to find out the object or goal position for Position Based algorithm. The particle swarm optimization is used for Position based algorithm as it is characterized as a simple concept, which is both easy to implement and computationally efficient [9,10]. The robotic platform described in the paper is designed and developed in Programmable Logic Controller laboratory at University of Bridgeport, CT, USA. The rest of the paper is organized as follows: Sec. II describes the problem definition; Sec. III gives some details about the solution, while Sec. IV describes the conclusion and Sec. V gives the future scope.

Problem definition

A wave of an industrial automation given the profound impact on technologies and electronics leads to the revolution. Due to this industrial revolution, technological advances now allow the manufacturer to invent and create new ideas, product in mass production with enhanced quality [11]. The most of industrial processes uses manipulator arms for picking and placing the objects or products in close proximity of previously placed objects. The task assigned to the robot arms can be performed repeatedly with high accuracy and with precise movements of joint angles [12]. The common industrial manipulator is often referred as robot arm with joints and angles as shown in figure 1. The robot arm used in our application is an articulated arm consists of all revolute joints. The articulated robot arm has maximum flexibility and can reach over and under the objects. As all joints are revolute these robots can cover large workspace and joints are easy to seal. The robot manipulators are assigned to accomplish the specific task in unstructured environment with minimal joint movements and best shortest path (Figure 1).

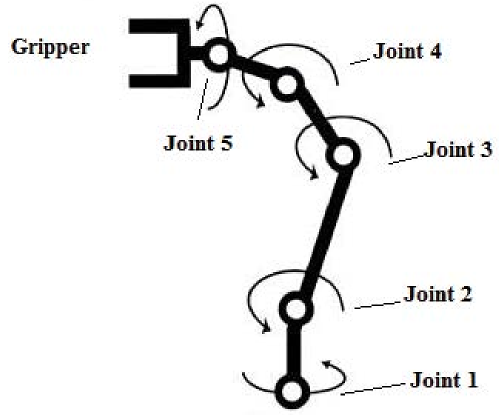


Figure 1: Robot arm with joint angle.

The robot motion path planning has been studied more than two decades and there are so many contributions made on this problem [13]. Deriving the best possible inverse kinematic solution for end effector is challenging and difficult. Some of the challenging aspects in designing reverse kinematic solution are:

A. States Space

As the manipulator is design for pick and place application, the manipulator pick up the part and place it in relative positions such as good part position or bad part or rejection area according to the sensory data. So the states finding for the robot manipulator is finite. The arm has its work space and it can reach to those positions by different paths.

B. Initial State

The initial state for the robot manipulator could be any state depends on the signal send by the sensors to the arm. But at the start of the program the manipulator always go to the home or nesting position.

C. Action

The manipulator action is depend on the perceived signal of the sensors (camera, part detect sensor, etc.). The manipulator takes action after the signal send to the controller by sensors and controller decides the movement of arm. So manipulator moves toward the destination area by avoiding the obstacles in the path as well as within the time limit and with respective speed. To reach to the goal or object manipulator has to find out the best possible path with minimal joint angle movement within its work space. The end effector will try to reach the object as soon as x and y coordinates are calculated. There are also some other actions such as for open and close the gripper, take snap shot of parts, start/stop the conveyor, start/stop the motor etc.

D. Transition Model

The transition model for the manipulator is depending on the action taken. At the start the manipulator returns to the homing position. So we will consider the initial state as the homing position, but not all the time. The camera mounted on the end effector will continuously feed the current position coordinates of the robot manipulator to the controller and the algorithm controls the motion of robot arm. The complete transition model is shown in figure (Figure 2).

E. Goal Test

The goal test of this model is to pick and place the objects/parts with respective area for further processing by detecting metal and non-metallic parts and avoiding the obstacle or collision of the robot arm with any other part of the system.

F. Path Cost

The path cost of this application is varies according to the product cost, which company parts you are using for the system assembling, and other factors.

The complete flow chart of the application designed and implemented as given in figure (Figure 3).

To achieve the goal of the application, the position trajectories calculation of robot manipulator is the most difficult task, as it has to avoid the obstacle and reach to the product picking [14]. As stated in above problem transition state, for this robot manipulator there are lots of possible path to reach object. Among those paths, the best path will have the minimal joint movements and shortest distance within its workspace. There are lots of algorithms and techniques have been developed for finding the inverse kinematics of the manipulator. The complexity of finding the kinematics solution increases with number of joints or Degree Of Freedom (DOF). The robot manipulator position, path planning and motion control in 3 dimensional workspace become a key factor for control system design engineers and robot manufacturer. The robot arm should be self-proficient, flexible, low power consumptions, fully efficient. The typical task for robot arm in industry is to move its end effector from initial position to some desired position in working environment with least residual vibration, minimal torque, obstacle avoidance and collision free kinematics, shortest time interval and/or distance of a desired path [15].

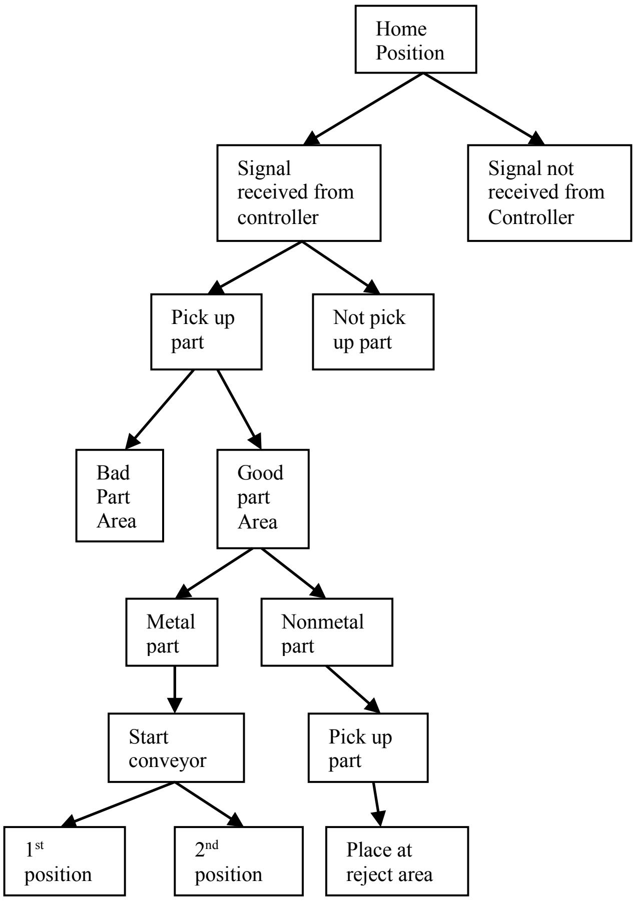


Figure 2: Transition model of the system.

Solution methodology

Dealing with complex, higher level control system with continuous interactive subsystem in dynamic environment is difficult and requires sophisticated and intellectual controller with continuously process optimization. We proposed the solution for above stated problem in this section. The proposed solution has been applied and tested on robot manipulator with 5 DOF. The experimental set up is explained in next section. In this experiment we are using position based algorithm in combination with Image based algorithm to finding the best possible solution for our stated problem. As we are implementing Artificial Intelligence in the robot arm control system based on PLC, we have to use search algorithms, inference engine etc. The system is completely knowledge based as uses the sensors, camera and applies the action on the robot arm and actuators. Solving the inverse kinematics becomes complex and nonlinear as the number of degrees of freedom of the manipulator increases. The position based search algorithm uses the visual data provided by the imaged based algorithm and calculates the joint velocity and angles to form the inverse kinematic solution in 3 dimensional workspace [16]. By adding vision or imaged based algorithm, the robotic control system is more flexible, adaptable and increases the accuracy in the joints movement (Figure 4).

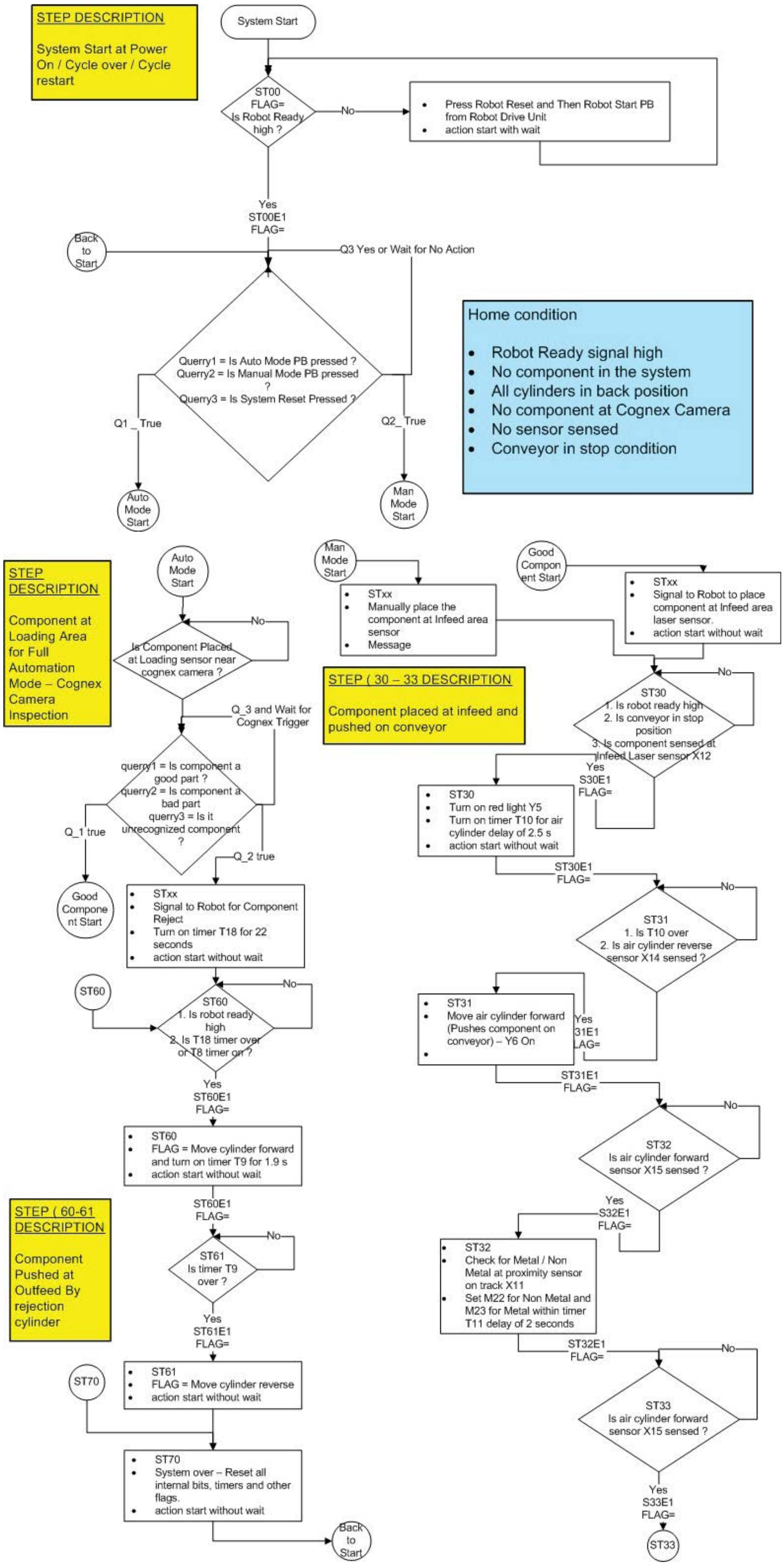
The principal advantage of using position-based control is the chance of defining tasks in a standard Cartesian frame. On the contrary, the image-based control is less sensitive to calibration errors; however, it is required the online calculation of the image Jacobian, that is, a quantity depending on the distance between the target and the camera which is difficult to evaluate. A control in the image plane results also to be strongly nonlinear and coupled when mapped on the joint space of the robot and may cause problems when crossing points which are singular for the kinematics of the manipulator.

The pseudo code for the position based algorithm is as given below,

```
For each position
{
   Initialize position
 }
    Do until maximum iterations or minimum error criteria
      {
        For_each position
        {
          Calculate joint angle kinematics
           If_ Joint1 better than Joint2
           {
               Set pBest = current Joint angle
           }
             If pBest is better than gBest
           {
              Set gBest = pBest
           }
        }
      }
   For each position
   {
      Calculate joint Velocity
      Use gBest and Velocity to update position Data
   }
```

The main purpose or goal of this study is to use the visual feedback from images captured by the camera and the Cartesian space co-ordinates of the target object which ultimately controls the motion of the robot to perform a task. The starting position and coordinate frame boundary is taught to the robot arm. From the calculated co-ordinates of the target object, the surface model in 3 dimensional co-ordinate systems will be constructed for robot manipulator. The sensor data will be used for knowledge base and will be used to avoid collision as well as can be used to find optimal shortest possible path or trajectory by checking each point in its workspace. The sensors can give a signal when contact is made with obstacles, detect metal or non-metal objects, or measure a force being applied. Due to this knowledge provided by sensory units to the system, robot path can be planned before its execution to the target position.

Firstly, a camera is mounted on a manipulator end effector and it catches a 2-D image, a true potential can be exactly calculated. We assign the two dimensional coordinate system with the x-axis and


STEP DESCRIPTION
System Start at Power On / Cycle over / Cycle restart
System Start
ST00 FLAG= Is Robot Ready high ?
No
Press Robot Reset and Then Robot Start PB from Robot Drive Unit
action start with wait
Yes ST00E1 FLAG=
Back to Start
Q3 Yes or Wait for No Action
Querry1 = Is Auto Mode PB pressed ?
Querry2 = Is Manual Mode PB pressed ?
Querry3 = Is System Reset Pressed ?
Q1 _ True
Q2_ True
Auto Mode Start
Man Mode Start
Home condition
Robot Ready signal high
No component in the system
All cylinders in back position
No component at Cognex Camera
No sensor sensed
Conveyor in stop condition
STEP DESCRIPTION
Component at Loading Area for Full Automation Mode – Cognex Camera Inspection
Auto Mode Start
Is Component Placed at Loading sensor near cognex camera ?
No
Q_3 and Wait for Cognex Trigger
querry1 = Is component a good part ?
querry2 = Is component a bad part
querry3 = Is it unrecognized component ?
Q_1 true
Q_2 true
Good Component Start
STxx
Signal to Robot for Component Reject
Turn on timer T18 for 22 seconds
action start without wait
ST60
ST60 1. Is robot ready high 2. Is T18 timer over or T8 timer on ?
No
Yes ST60E1 FLAG=
ST60
FLAG = Move cylinder forward and turn on timer T9 for 1.9 s
action start without wait
ST60E1 FLAG=
STEP (60-61 DESCRIPTION
Component Pushed at Outfeed By rejection cylinder
ST61 Is timer T9 over ?
No
Yes ST61E1 FLAG=
ST70
ST61
FLAG = Move cylinder reverse
action start without wait
ST70
System over – Reset all internal bits, timers and other flags.
action start without wait
Back to Start
Man Mode Start
STxx
Manually place the component at Infeed area sensor
Message
Good Component Start
STxx
Signal to Robot to place component at Infeed area laser sensor.
action start without wait
STEP (30 – 33 DESCRIPTION
Component placed at infeed and pushed on conveyor
ST30 1. Is robot ready high 2. Is conveyor in stop position 3. Is component sensed at Infeed Laser sensor X12
No
Yes S30E1 FLAG=
ST30
Turn on red light Y5
Turn on timer T10 for air cylinder delay of 2.5 s
action start without wait
ST30E1 FLAG=
ST31 1. Is T10 over 2. Is air cylinder reverse sensor X14 sensed ?
No
ST31
Move air cylinder forward (Pushes component on conveyor) – Y6 On
Yes 31E1 LAG=
ST31E1 FLAG=
ST32 Is air cylinder forward sensor X15 sensed ?
No
Yes S32E1 FLAG=
ST32
Check for Metal / Non Metal at proximity sensor on track X11
Set M22 for Non Metal and M23 for Metal within timer T11 delay of 2 seconds
ST32E1 FLAG=
ST33 Is air cylinder forward sensor X15 sensed ?
No
Yes S33E1 FLAG=
ST33

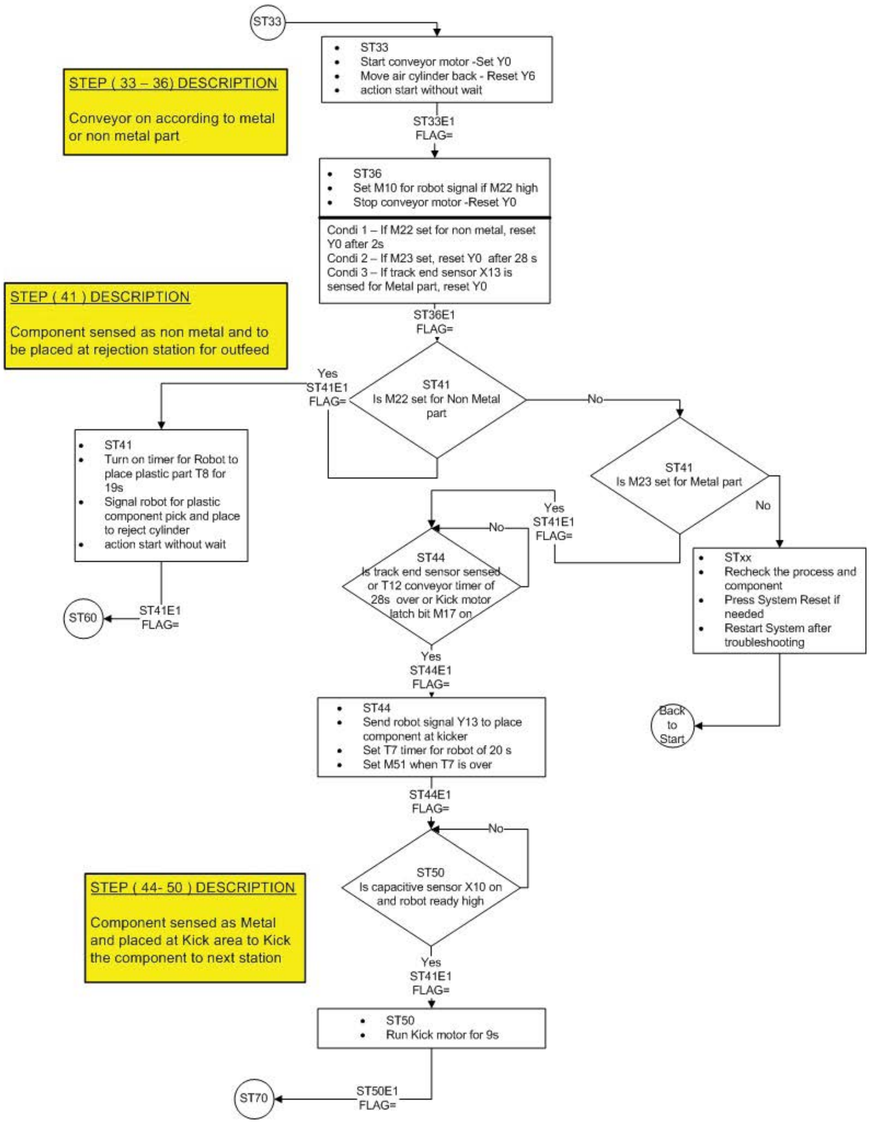

ST33
ST33
Start conveyor motor -Set Y0
Move air cylinder back - Reset Y6
action start without wait
STEP (33 – 36) DESCRIPTION
Conveyor on according to metal or non metal part
ST33E1 FLAG=
ST36
Set M10 for robot signal if M22 high
Stop conveyor motor -Reset Y0
Condi 1 – If M22 set for non metal, reset Y0 after 2s
Condi 2 – If M23 set, reset Y0 after 28 s
Condi 3 – If track end sensor X13 is sensed for Metal part, reset Y0
STEP (41) DESCRIPTION
Component sensed as non metal and to be placed at rejection station for outfeed
ST36E1 FLAG=
Yes ST41E1 FLAG=
ST41 Is M22 set for Non Metal part
No
ST41 Is M23 set for Metal part
No
ST41
Turn on timer for Robot to place plastic part T8 for 19s
Signal robot for plastic component pick and place to reject cylinder
action start without wait
Yes ST41E1 FLAG=
No
ST44 Is track end sensor sensed or T12 conveyor timer of 28s over or Kick motor latch bit M17 on
STxx
Recheck the process and component
Press System Reset if needed
Restart System after troubleshooting
ST60
ST41E1 FLAG=
Yes ST44E1 FLAG=
ST44
Send robot signal Y13 to place component at kicker
Set T7 timer for robot of 20 s
Set M51 when T7 is over
Back to Start
ST44E1 FLAG=
No
STEP (44- 50) DESCRIPTION
Component sensed as Metal and placed at Kick area to Kick the component to next station
ST50 Is capacitive sensor X10 on and robot ready high
Yes ST41E1 FLAG=
ST50
Run Kick motor for 9s
ST70
ST50E1 FLAG=

Figure 3: Flow chart of the complete process.

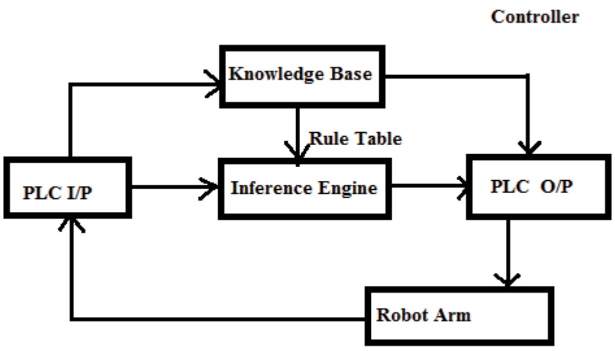


Figure 4: Block diagram of PLC based robot arm control system.

y-axis forming a basis for the image plane and the z-axis perpendicular to the image plane. The origin located at distance ʎ behind the image plane as shown in figure (Figure 5).

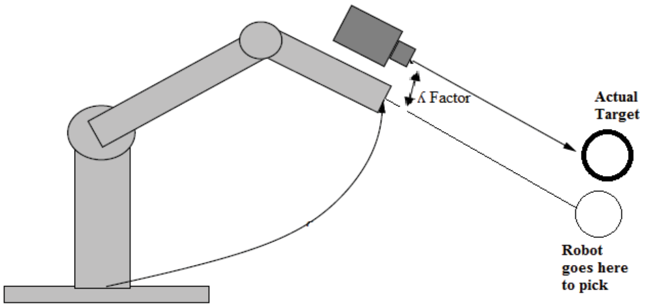


Figure 5: Camera mount position.

In order to determine the position of the target object in the image plane, camera will capture the images and through which only the central point (only a single pixel) have to be identified, without knowing this point's co-ordinates in the attached Cartesian reference system. This coordinates will be feed to the Position Based Algorithm for calculating the joints angle and trajectory. The position finding algorithm outputs the x-coordinate and y-coordinate in the image frame along with a scaling factor (ʎ). The scaling factor is related to the dimensions of the gripper and is used to get an idea of the elevation of the gripper. This helps in making the gripper co-planar with the target object.

Hardware setup

We implemented our proposed algorithms and optimal path planning schemes on a PLC based Robot arm control system for sorting, pick and place application as shown in Figure (Figure 6).

Following are the main components of the project:

1. Mitsubishi Fx3G Programmable Logic Controller.
2. Mitsubishi RV-M1 Mitsubishi Robot along with Movemaster Drive Unit.
3. Mitsubishi touch screen E-1061 HMI (Human Machine Interface).
4. VGA Camera.
5. Cognex Vision Camera unit.
6. A servomotor for controlling belt conveyor and another servomotor for displacing a component as a kicker
7. Sensor system comprising of laser sensors, capacitive sensor, track sensor and proximity.
8. Two pneumatic cylinders which act as component pusher.

The Mitsubishi RV-M1 robot is driven with robot drive unit Movemaster EX from which also provides interface with Mitsubishi PLC with help of 50 pin connector and another interface for robot teach pendant. This drive unit stores the code for the robot. The nomenclature of the robot arm is shown in figure (Figure 7).

The specification of Movemaster robot arm is as given in figure (Figure 8).

The Programmable Logic Controller used in this experimental set up is Mitsubishi Fx3G and its basic specification are given below,

Power Supply: 100-240 V AC/ 12-24 V DC, 50/60 Hz

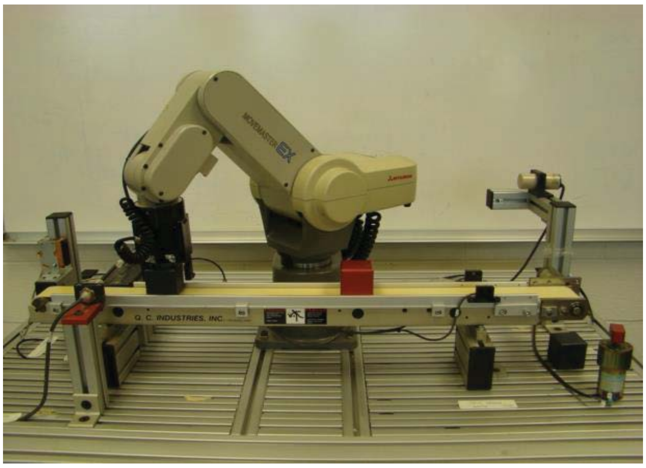
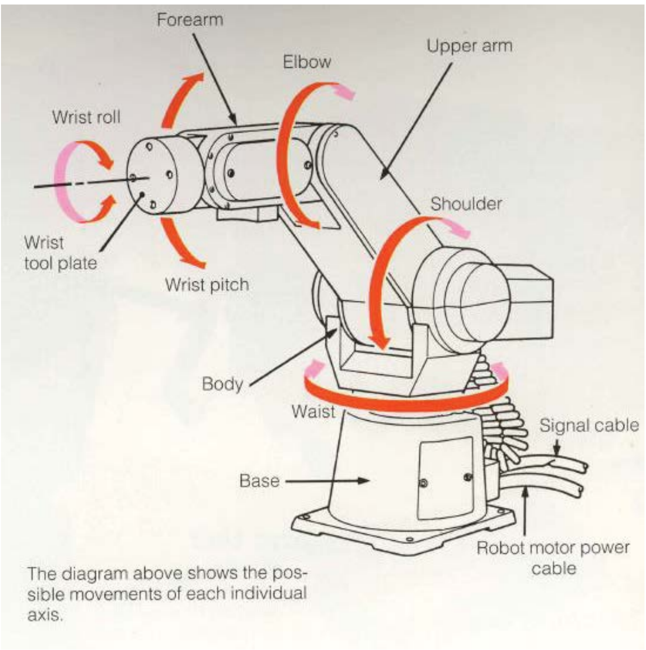
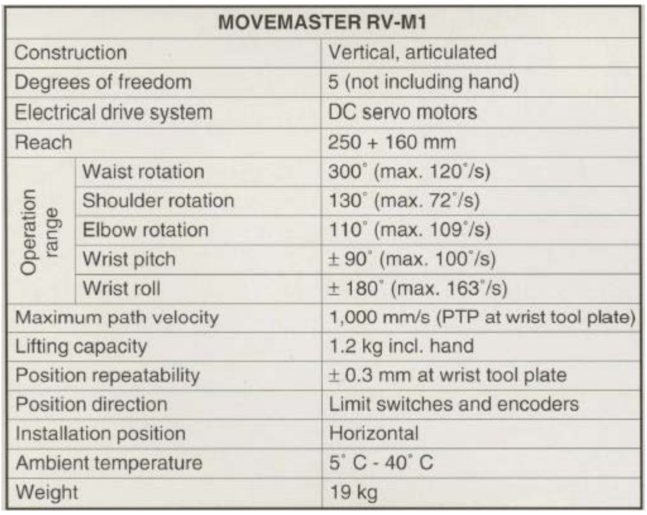


Figure 6: PLC system setup.



The diagram above shows the possible movements of each individual axis.

Figure 7: Nomenclature of robot arm.



MOVEMASTER RV-M1		
Construction		Vertical, articulated
Degrees of freedom		5 (not including hand)
Electrical drive system		DC servo motors
Reach		250 + 160 mm
Operation range	Waist rotation	300° (max. 120°/s)
	Shoulder rotation	130° (max. 72°/s)
	Elbow rotation	110° (max. 109°/s)
	Wrist pitch	± 90° (max. 100°/s)
	Wrist roll	± 180° (max. 163°/s)
Maximum path velocity		1,000 mm/s (PTP at wrist tool plate)
Lifting capacity		1.2 kg incl. hand
Position repeatability		± 0.3 mm at wrist tool plate
Position direction		Limit switches and encoders
Installation position		Horizontal
Ambient temperature		5° C - 40° C
Weight		19 kg

Figure 8: Specification of robot arm.

No. of Outputs: 16.

No. of Inputs: 24.

Input form: sink / source with photocoupler isolation

Input / Output response: Approx 10 ms

Digital Output: Relay Transistor.

Program Cycle Period: 0.55-1 us

Memory: 32K steps EEPROM.

Software: GX Developer.

USB Port, Profibus or Ethernet Connection

Results

The application we used for testing these two algorithm with PLC is Pick and Place application with Quality Inspection. As an initial step we localize the object using a traditional Haar classifier. We trained multiple models for recognition various objects in the scene. At time of training process each Haar object detection model will be fed with respective positive samples or in class samples and out of class samples. We then evaluate the performance of the trained model using test data. The processing is performed on a windows machine using Matlab Software. We chose this method (Haar classifier) as it is one of the successful and simple object detection method used widely in machine vision technologies. Some of the sample training images are shown in figures (Figure 9-12).

The pseudo code for Experimental Setup is as given below-

1. Initialize:

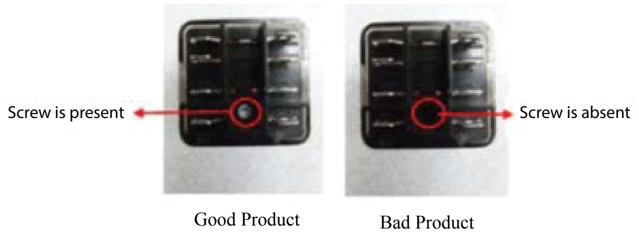


Figure 9: Job 1 test product images.

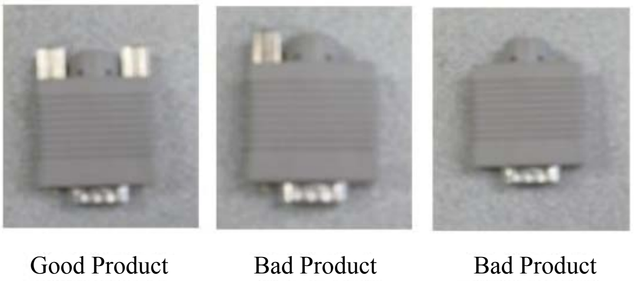


Figure 10: Job 2 test product images.

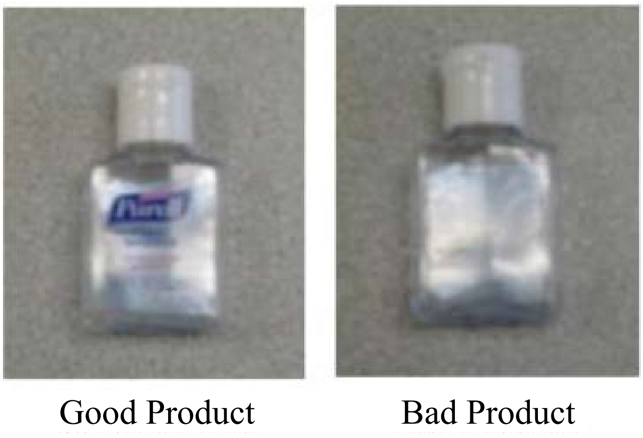


Figure 11: Job 3 test product images.

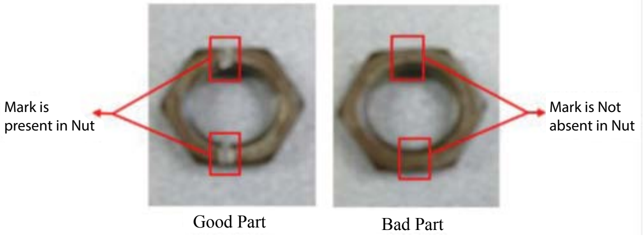


Figure 12: Job 4 test product images.

Start node Stack = empty

End node Stack = empty

All other variables initialized.

Load the mode into Start node Stack equivalent to sNODE out degree each time incrementing the start index. Start index to point to sNODE the Root, on top of stack. Now pop the top of stack and assign to the Depth First Search (DFS) proceed to 2

2. **Begin** (At start node Get the unvisited adjacent)

while (there are bNODEs proceed with PBA and IBA algorithm and generate Step code for the branch till you reach a tNODE or xNODE)

if (xNODE proceed to 3)

else if (tNODE go to 7)

else go to 10

3. At xNODE check node status;

if (we are meeting this xNODE for the first time) go to 4

else (top of end index points to this xNODE pop End node Stack and go to 5)

New Optimal Guidance for Mountaineer Lifesaving Mission

4. Make this xNODE as an end node, if bad part is detected on Cognex vision system Stack equivalent to its in degree each time incrementing the end index. Then go to 5

5. Process this xNODE and increment the in degree processing count for this node.

If (the in degree processing count equal to the xNODE in degree) go to 6

else if (Start node Stack not empty) get the start node ID from stack by popping Start node Stack assign ID to DFS then go to 2

else go to10.

6. Make this node a start node, put its ID on Start node Stack equivalent to its out degree

if (Start node Stack not empty,) pop the Start node Stack assign ID to DFS and then go to 2

else go to 10.

7. Process this tNODE then

if (Start node Stack is empty) proceed to 8

else pop the Start node Stack assign to DFS and then go to 2

8. **if** (End node Stack empty and statistic check correct) proceed to 9

else go to 10

9. Good metal product successfully processed to packaging/loading area

exit (Normal)

10. Bad non-metal product. Encountered an error, **send to rejection area**.

Exit

Following are the test images that were used as a target image to evaluate proposed methodology. We assigned 4 different jobs. Each job has different products with unique features to distinguish whether it is good or bad product.

JOB 1

We used relay for job 1, it will check whether screw in the socket is present or not. If cognex checker detects screw on the right position then it will consider it as good part and send it to accept position else it will send it to reject position (Figure 9).

JOB 2

We used holder for job 2, checker will check whether holding pins are assembled in holder during manufacturing process or not. If all holding pins are present in holder then checker will consider it as good product else it will consider it as bad product and send the good product to accept position and bad product to reject position (Figure 10).

JOB 3

Purell hand sanitizer is used in job 3. In this job we check the label of Purell brand name in the hand sanitizer container. Container with Purell brand name is considered as a good product and without Purell brand name is considered as a bad product (Figure 11).

JOB 4

We used nut for job 4. Nut with two marks is considered as good product and nut without marks will be considered as a bad part (Figure 12).

Conclusion

In this paper we proposed an algorithms and implementation method for calculating the inverse kinematic solution and trajectory planning for industrial robot manipulator using the Position based algorithm in combination with Image based algorithm. The implementation is carried out on Movemaster Robot Arm with PLC for sorting metal and non-metal objects, pick and place application. The proposed combination algorithm reduces the computation time and positioning error for finding the target in real time.

References

1. Patil M, Sheth A, Toporovsky PJ (2011) Implementation of a robotic conveyor assembly line with vision camera using two different coding techniques and comparison of their performance parameters. In proc 2011 ASEE Northeast Section Annual Conference.
2. AB AAT (2006) IRB 340 datasheet.
3. The orocos project: Open robot control software.
4. Macchelli A, Melchiorri C (2002) A real-time control system for industrial robots and control applications based on real-time Linux. In 15th IFAC world congress, Barcelona, Spain.
5. Asada H, Youcef-Toumi K (1983) Analysis and design of semidirect drive robot arms. In: Proc Amer C Conf, San Francisco, CA.
6. Kao I, Gong C (1997) Robot based computer integrated manufacturing as applied in manufacturing automation. Robot and Comput Int Manuf 13: 157-167.
7. Bonfe M, Vignali M, Fiorini M (2009) PLC-based control of a robot manipulator with closed kinematic chain. In Proc IEEE Int Conf Robotics and Automation.
8. Whitney DE, Lozinski CA, Rourke JM (1986) Industrial robot forward calibration method and results. J Dyn Sys Meas Control 108: 1-8.
9. Rini DP, Shamsuddin SM, Yuhaniz SS (2011) Particle swarm optimization: Technique, system and challenges. Int J Comput Appl 14: 19.
10. Hashimoto K, Kimoto T, Ebine T, Kimura H (1991) Manipulator control with image-based visual servo. In Proc IEEE Int Conf Robotics and Automation.
11. Patil M, Toporovsky Y, Lawrence H, Sobh T (2012) Integration of vision system, intelligent Robo Actuator, HMI and PLC to design a universal quality inspection or control machine. I-managers J Mech Eng 2:17.
12. Khatib O, Quinlan S, Williams D (1997) Robot planning and control. Robot Auton Syst 21: 249-261.
13. Glavina B (1990) Solving findpath by combination of goal-directed and randomized search. In Proc IEEE Int Conf Robotics and Automation. Pp: 1718-1723.
14. Feng-Yih H, Li-Chen F (1995) A new design of adaptive fuzzy hybrid force/position controller for robot manipulators. In Proc IEEE Int Conf Robotics and Automation 1:863-868.
15. Ogunlowore O (2013) Realtime motion planning for manipulator robots under dynamic environments: An optimal control approach. In mechanical engineering, University of Waterloo, Canada.
16. Palmieri G, Palpacelli M, Battistelli M, Callegar M (2012) A comparison between position-based and image-based dynamic visual servoings in the control of a translating parallel manipulator. J Robotics.