Scientific Journal of Impact Factor (SJIF): 3.134

ISSN (Online): 2348-4470
ISSN (Print) : 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2, Issue 4, April -2015

A Methodology Of Programming For TAL TR-2 Articulated Robot

Suraj D. Golhar[1], Prashant B. Vavhal[2], Amit P. Jagnade[3], Prof. Vidyasagar Bajaj[4], Prof. Nitin Jadhao[5]

[1,2,3] *Mechanical Engineering, Alard College of Engineering & Management, Pune*
[4] *Head of Department Mechanical Engineering, Alard College of Engineering and Management, Pune*
[5] *Professor Department of Mechanical Engineering, Alard College of Engineering and Management, Pune*

Abstract — *This paper is concerned with learning the technology and programming of industrial robots. A TAL TR-2 articulated robot is used in this paper. In this paper we are studying operation and programming of the robot manipulator and controller hardware by carrying out trials on TAL TR2 robot. The programming for this is been done with the help of teach pad in RobWin7 software. A set of trials are produced for pick and place application. This paper also contains study the mechanism of software control of the robot and various types of programming viz. manual, lead through programming.*

Keywords- *Articulated Robot, Manipulator, Controller, Teach pad, RobWin7, Manual programming, Lead through programming.*

I. INTRODUCTION

A motion task given to the machine must ultimately be represented as a reference signal, which is sent to the control system. The control system acts to make the machine track the reference signal by activating the appropriate actuators. If the reference signal changes too quickly, given the dynamic limitations of the machine, the tracking of the reference signal will be poor, regardless of the control system design. Computer algorithms are designed to calculate an appropriate reference signal based on the desired task path and time-related limits (such as speed and acceleration). This reference signal is the trajectory, and can be defined as a locus of points in operational or joint space on which a time-law has been specified.

Motion planning involves generating the path and its time law, providing the controller's reference signal. Motion planning is often done off-line, typically when the trajectory generation algorithms are computationally intensive. However, it is often desirable to generate trajectories online so that changes can easily be made to the machine's trajectory, increasing the system's overall robustness and adaptability. For example, a manipulator may require the ability to recompute its trajectory on-line in order to avoid an unexpected obstacle that lies along a path on which it is currently moving.

In industrial applications, it is common to use simple PID control laws, which do not take into account the system's nonlinearities, to track a reference signal. To compensate for tracking errors introduced by the system's nonlinearities, a more complicated controller must be used.

The main importance of robot functionality is its flexibility and ability to rearrange the new production, and its movement range. The flexibility of the robot depends on presupposed effective programming. Principally, the robot programming can take place in two different ways: on-line or off-line. In on-line programming, the use of the robot and equipment is required, whereas off-line programming is based on computer models of the production equipment. Both these methods have advantages and disadvantages. In this section we will look at how the two methods can be combined.

1.1 On-Line Programming

Currently, the operation of robot programming is through either on-line programming or off-line programming. The definition of on-line programming is a technique to generate a robot program using real robot systems. An on-line programming robot may be suitable to implement robot use by repeating a monotonous motion. The advantage of on-line programming is that it is easy to access. Its most significant advantage is that the robot is programmed in concordance with the actual position of the equipment and pieces as shown in figure 1.1.

However, the most significant disadvantage of on-line programming is the slow movement of the robot, the program implementation, program logic and calculations being hard to programme, the suspension of production whilst programming, poor documentation and costs equivalent to production value. The differentiation between on-line and off-line programming is shown clearly in the picture below.

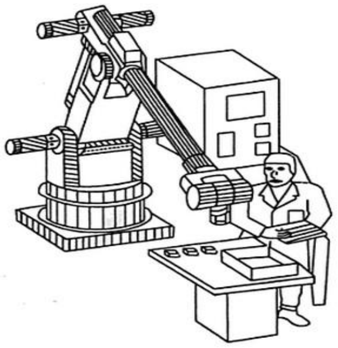


Figure 1 On-line robot programming

1.2 Off-Line Programming

The definition of off-line programming is that it is the technique used to generate a robot program without using a real machine. There are several advantages to this resulting in a reduction in robot downtime, programs being prepared without interruptions to robot operation, and removal of the human from the potentially dangerous environment. There are several types of programming language used in off-line robotic programming, but the most popular off-line programming software was built using the JAVA language.

Off-line programming using JAVA language become famous because it was easy to use, could be integrated with UML, supported C/C++ languages, had architecture independence and had an advanced network support. As shown in figure 1.2 the main advantage of off-line programming is that it does not occupy production equipment, and in this way, production can continue during the programming process. But on-line programming is the largest proportion of robot programming today due to the fact that off-line programming has had a very high burden rate and has demanded the need of expert users.

Advanced off-line programming tools contain facilities for debugging and these assist in effective programming. The programming tools support the utilization of supporting tools for the programming process, for instance optimization of the welding process.

However, off-line programming also has its disadvantages such as the fact that it demands investing in an off-line programming system. Most simulation tools/simulators are implemented with four characteristic: workspace visualization, trajectory planning, and communication with robot control and system navigation.

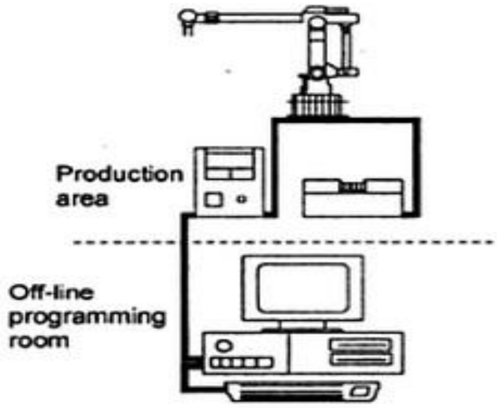


Figure 2 Off-line Robot programming

II. TYPES OF PROGRAMMING

2.1 Powered Lead Through or Tech Pendant Lead Through Programming

- It is the most common method used for robot programming.
- It makes use of tech pendant for controlling the various joint motor. Tech Pendant is a small hand held device, sometimes called as control box that contains toggle switches, dial and buttons to regulate the

physical movement and programming capabilities of robot. Teach pendant is simple to handle and learn and suitable to program may task in industry. It doesn't need a skilled operator. The programme cannot be entered into the pendant while the robot is offline.

- It result in movement of robot arm and wrist through a series of point in working space where in each point is recorded there into the memory for further playback during the work cycle, and during playback testing, the robot moves through the sequence of positions under the own power.
- This is limited to point to point motion of the manipulator, which includes transfer of parts from one point to another point, machine loading and unloading through a particular sequence and spot welding when welding gun moves from one point to another point.

2.2 Manual Lead Through or Walk-Through Programming

- It is suitable for playback robots continuous path.
- Also it fulfills the irregular motion pattern, roughly locating the tool centre point.
- Application of walk-through programming.

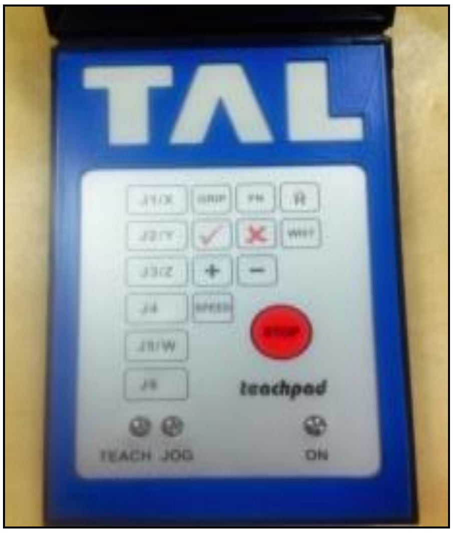
  It includes:
  - ➢ Spray Painting.
  - ➢ Arc welding.
  - ➢ Grinding.
  - ➢ De-burring and polishing.
- Requirement:
  - ➢ It requires the operator to physical grasp the end effector and manually moves it through the motion sequence.
  - ➢ It also requires the recording of the path into memory. The motion are recorded as a series of closely spaced point and during actual playback the actual playback the actual robot arm goes through the same sequence of points.
  - ➢ If the robot arm is heavy and unable to move on itself, an actual robot is replaced by a special device. This device has similar joint configuration to the robot. It equipped with control switch that is activated by the operator.
  - ➢ It requires a highly skill operator to perform the précised motion tasks if the robot cannot be programmed offline.

2.3 **Software used for Tal Robot**

Rob Win © Sands Technology International 1996-2012; this is a Windows based application which runs in the computer. It communicates with the robot controller and also saves robot software on disk. It keeps an identical copy of the controller data in memory while you use it.

2.4 MOVING ROBOT WITH THE TEACH PAD

Press the button. (Red teach LED lights – green already on) The computer asks for a teach speed which defaults to 16. Don't type or click anything, pad is ready to use.



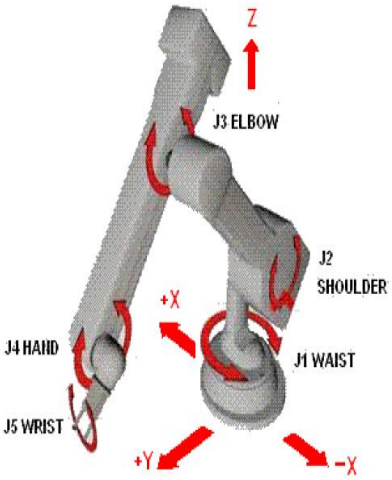


Figure 3 Teach Pad for TAL Robot ***Figure 4 Joint Representation for Robot***

NOTE: Press J1, J2, J3, J4 or J5 to select which joint is to move

2.5 TEACH PAD TEACHING

For TAL TR2 Robot J1 is waist, J2 is shoulder, J3 elbow, J4 hand (pitch), J5 wrist, J6 is track. Release the + key stops movement. Press – and hold Moves the selected joint in reverse. Release Stops movement. To increase the speed press SPEED then +; to decrease press SPEED then. This is confirmed on screen. Press the joint key again then + or – to move at the new speed. Press the WH? Key displays current position of the robot. Press the H key Moves all the joints to their zero positions, Same as HOME. Press GRIP selects the gripper. Press [+] Closes the gripper. Press [-] Opens the gripper.

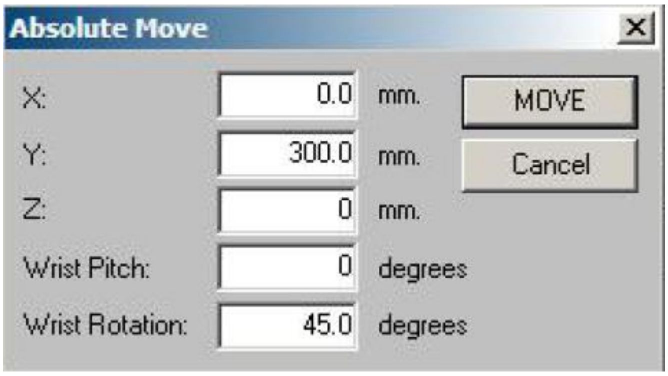
2.6 PROGRAMING TYPE OF ROBWIN 7.0

- ***CARTESIAN COORDINATES***

Type HOME or click button to make sure the robot is at HOME
Now click the Cartesian button. Command sent to controller is CARTESIAN but ROBWIN also
knows about it. Now on the menu bar click Robot Cartesian position. This displays the Cartesian position of the robot at home. Or you could have typed WHERE in the communication window which reports the current position of the robot in Cartesian coordinates. There are 6 possible values:
X Distance along X axis from the origin in mm.
Y Distance along Y axis from the origin in mm.
Z Distance along Z axis from the origin in mm.
Wrist Pitch: The angle of the hand to horizontal.

Controller variable PITCH Wrist Rotation (roll): The roll angle of the wrist. Controller variable W. There is also another variable, TOOL-LENGTH not accessible by ROBWIN except through the communications window - see main manual later for full explanation. (5-axis versions only)

On an TAL TR2 the origin (0,0,0) is at the intersection of the shoulder and waist axes. The X-Y-Z position described is the position of the intersection of the wrist rotates and hand (wrist pitch) axes. Click robot, Cartesian abs. Move This displays the current position of the robot and permits you to enter new values as required. When you click MOVE or OK the robot moves to that position.



Absolute Move

X: 0.0 mm.
Y: 300.0 mm.
Z: 0 mm.
Wrist Pitch: 0 degrees
Wrist Rotation: 45.0 degrees

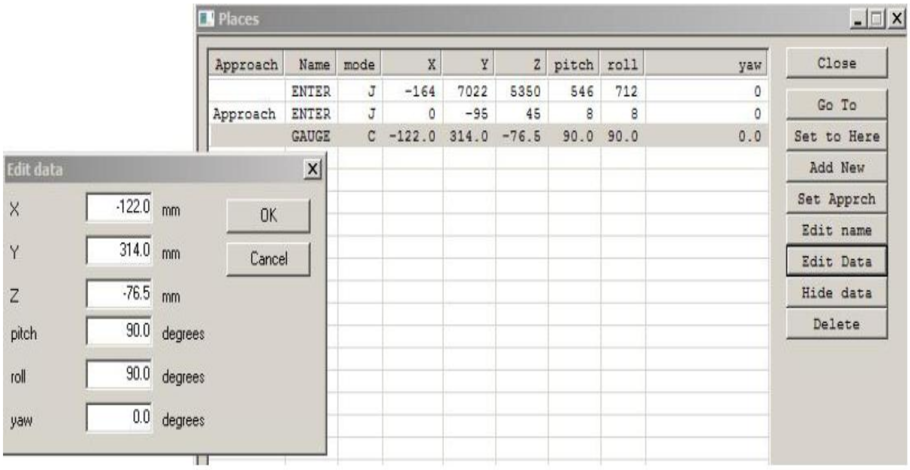
Figure 5 Cartesian Absolute Move

This will move the robot to zero on the X axis, 300mm from center on the Y axis and zero on the Z axis (which for TR2 means level with the shoulder pivot). Wrist rotates to 45 degrees.

NOTE: TAL TR-2 will not assume a low Z value with the hand pointing up (-90.0 deg). This is because there is a software trap to prevent the hand colliding with the covers.

2.7 MOVING ROBOT WITH THE TEACH PAD IN CARTESIAN MODE

Press the button (Jog). (‘Jog’ LED lights) The computer asks for an increment which defaults to10mm. Don’t type or click anything, pad is ready to use.(You can enter a new increment first if you wish but keep the default for now. The largest values you can enter are 25.5 and remember that 25.4 is one inch). On the keypad select X Y or Z. Then press + or – one time (not press and hold as before). On each press the robot moves in that axis by the set increment. To increase the increment press SPEED then +; to decrease press SPEED then – Then press the X Y or Z key again and + or – to move a different axis.J4 selects the hand – an increment of 10.0 means 10 degrees.J5 selects wrist – an increment of 10.0 means 10 degrees. It is best to use the dialog box first to get the robot roughly where you want it before using JOG. To operate the gripper press GRIP then + to close or – to open. To exit the teach pad click the escape button on the PC. Teaching the robot is a process of moving to a position and learning that position. There are two devices for learning robot positions: a PLACE - this is a named single position to which the robot will go when required. The user must choose which is most suitable for the task and must also choose a name which describes what the route or place means in the context of the application. A task can involve more than one route or place or a mixture. All the routes and places will be saved CARTESIAN mode. If you click Show data the places window opens up to show the actual coordinates learned for each PLACE.



Approach	Name	mode	X	Y	Z	pitch	roll	yaw
	ENTER	J	-164	7022	5350	546	712	0
Approach	ENTER	J	0	-95	45	8	8	0
	GAUGE	C	-122.0	314.0	-76.5	90.0	90.0	0.0

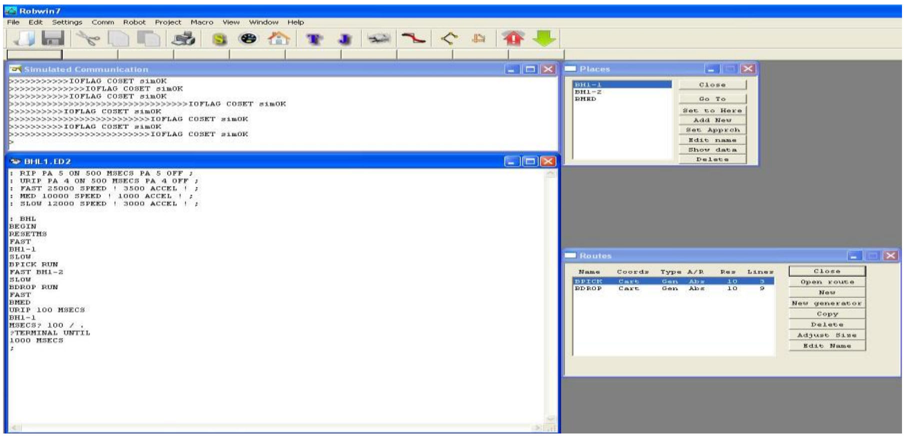
X	-122.0	mm
Y	314.0	mm
Z	-76.5	mm
pitch	90.0	degrees
roll	90.0	degrees
yaw	0.0	degrees

Figure 6 Places from Teach Pad

You can now edit the data manually if you wish. Click the name of the PLACE. That line is highlighted and the header line at the top changes to show the coordinates names (Joint or Cartesian). Click edit data to edit a line. A new box appears and you can change the value in any box. Click OK to save the new value(s). You can also delete or rename a PLACE. However this may cause the project to be reloaded to get the words in the right order.

2.8 PROGRAM FOR PICK AND PLACE APPLICATION IN ROBWIN7

```
: RIP PA 5 ON 500 MSECS PA 5 OFF ;
: URIP PA 4 ON 500 MSECS PA 4 OFF ;
: FAST 25000 SPEED ! 3500 ACCEL ! ;
: MED 10000 SPEED ! 1000 ACCEL ! ;
: SLOW 12000 SPEED ! 3000 ACCEL ! ;
: BHL
BEGIN
RESETMS
FAST
BH1-1
SLOW
BPICK RUN
FAST BH1-2
SLOW
BDROP RUN
FAST
BMED
URIP 100 MSECS
BH1-1
MSECS? 100 / .
?TERMINAL UNTIL
1000 MSECS;
```



```
>>>>>>>>>>>IOFLAG COSET simOK
>>>>>>>>>>>>>>IOFLAG COSET simOK
>>>>>>>>>>>IOFLAG COSET simOK
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>IOFLAG COSET simOK
>>>>>>>>>>IOFLAG COSET simOK
>>>>>>>>>>>>>>>>>>>>>>>>>IOFLAG COSET simOK
>>>>>>>>>>IOFLAG COSET simOK
>>>>>>>>>>>>>>>>>>>>>>>>>IOFLAG COSET simOK
>
```

```
: RIP PA 5 ON 500 MSECS PA 5 OFF ;
: URIP PA 4 ON 500 MSECS PA 4 OFF ;
: FAST 25000 SPEED ! 3500 ACCEL ! ;
: MED 10000 SPEED ! 1000 ACCEL ! ;
: SLOW 12000 SPEED ! 3000 ACCEL ! ;

: BHL
BEGIN
RESETMS
FAST
BH1-1
SLOW
BPICK RUN
FAST BH1-2
SLOW
BDROP RUN
FAST
BMED
URIP 100 MSECS
BH1-1
MSECS? 100 / .
?TERMINAL UNTIL
1000 MSECS
;
```

```
BH1-1
BH1-2
BMED
```

Name	Coords	Type	A/R	Res	Lines
BPICK	Cart	Gen	Abs	10	3
BDROP	Cart	Gen	Abs	10	9

Figure 7 Programming for Pick & Place Robot

III. CONCLUSION

This paper represents the various types of programming. Robot programming is substantially different from traditional programming. We can identify several considerations which must be handled by any robot programming method. The objects to be manipulated by a robot are three-dimensional objects which have a variety of physical properties; robots operate in a spatially complex environment; the description and representation of three-dimensional objects in a computer are imprecise; and sensory information has to be monitored, manipulated, and properly utilized.

REFERENCES

[1]. John J. Craig, “Introduction to Robotics Mechanics and Control”; Pearson Education International (Third Edition), ISBN 0-13-123629-6.

[2]. ONO Kazuya, HAYASHI Toshihiro, FUJII Masakazu, SHIBASAKI Nobuhiro & SONEHARA Mitsuharu, “Development For Industrial Robotics Applications”; IHI Engineering Review, Vol. 42 No.2, 2009.

[3]. Mohd Firdaus Bin Rusli, “Development Of Servo Motor Pick And Place Robot With Pic16f877a Microcontroller”; April 2008.

[4]. Shimon Y. Nof , “Handbook Of Industrial Robotics”; John Wiley Co, 2001

[5]. A. B. Humbe, P. A. Deshmukh, M. S. Kadam “The Review Of Articulated R12 Robot And Its Industrial Applications”, International Journal Of Research In Engineering & Technology IJRET, Vol. 2, Issue 2, Feb 2014, 113-118.

[6]. A.K. GUPTA, S.K. ARORA, “Industrial Automation And Robotics”, S. CHAND Publication