Scientific Journal of Impact Factor(SJIF): 3.134
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 5, May -2015

# Task Programming of Robotic welding using touch sensing principle

Gopi Damle[1], Jagrut Gadit[2]

[1]PG Scholar, ElectricalEngg. Department, MS University, Baroda-390001, India,
[2] Associate Professor, ElectricalEngg. Department, MS University, Baroda-390001, India,

***Abstract: -*** *Robotics is the applied science of motion control for multi-axis manipulators and is a large subset of the field of "mechatronics" which is a more general term that includes robotic arms, positioning systems, sensors and machines that are controlled by electronics and/or software. Any form of physical work that a human being can do can be replicated or performed faster, more accurately, cheaper and more consistently using computer controlled robots and mechanisms. Robotic welding uses robots which completely automate a welding process by both performing the weld and handling the part. The Robotic welding is an automatic and adaptive process using robotics and automatic controls can result in consistency and desired output. It is possible to achieve completely automatic welding cycle by use of Sensors and Servo controls. Use of Touch Sense of robot is low cost, software based, joint location sensing system. Touch sensing is one of the most advance and accurate feedback mechanisms to compensate the plate waviness and to minimize rejections. It works by using the welding electrode, nozzle, or other sensing pointer to make electrical contact with the part. This paper makes the use of touch sensing principle for the programming of seam welding on the work piece using KUKA robot KR 16 L6 and KUKA controller KR C2.*

***Keywords: -****Robotic Welding; Touch Sensing; KUKA; Motion; Programming*

## I. INTRODUCTION

Modern manufacturing faces two main challenges; more quality at lower prices and the need to improve productivity. Those are the requirements to keep manufacturing plants in developed countries, facing competition from the low salary regions of the world. Other very important characteristics of the manufacturing systems are flexibility and agility of the manufacturing process, since companies need to respond to a very dynamic market with products exhibiting very short life-cycles due to fashion tendencies and worldwide competition. Consequently, manufacturing companies need to respond to market requirements efficiently, keeping their products competitive. This requires a very efficient and controlled manufacturing process, where focus is on automation, computers and software. The final objective is to achieve semi-autonomous systems, *i.e.*, highly automated systems that work requiring only minor operator intervention.

Robotic welding is one of the most successful applications of industrial robot manipulators. In fact, a huge number of products require welding operations in their assembly processes. Despite all the interest, industrial robotic welding evolved only slightly and is far from being a solved technological process, at least in a general way. The welding process is complex, difficult to parameterize and to monitor and control effectively. Welding can in most cases impose extremely high temperatures concentrated in small zones. The majority of industrial welding applications benefit from the introduction of robot manipulators, since most of the deficiencies attributed to the human factor is removed with advantages when robots are introduced. This should lead to cheaper products since productivity and quality can be increased, and production costs and manpower can be decreased. Nevertheless, when a robot is added to a welding setup the problems increase in number and in complexity. Robots are still difficult to use and program by regular operators, have limited remote facilities and programming environments, and are controlled using closed systems and limited software interfaces.

In this paper task programming for touch sensing based robotic welding system is shown using KUKA Robot. The system is required to be implementing at Danke Power Ltd., Baroda-Gujarat-India. The touch sensing based system has the advantage over the image processing techniques which used costly camera on the Robot. It works by using the welding electrode, wire, or other sensing pointer to make electrical contact with the part. The robot stores the position data and then makes adjustments automatically to the entire weld path before the arc start. Touch Sensing allows for adjustments in one, two, or three dimensions. The programming of the touch sensing of robotic welding is to be done with KUKA system software (KSS V5.x). It is also required to program the system in a manner that motion of the tank mounted on fixture must fall in tandem at a suitable angle with the motion of the KUKA robot. This means that the robot system has to execute corrections from the program path to ensure desired set point regardless of the disturbance in terms of defects in raw material such as waviness and tacking errors. The touch sense must be adaptive to the variation between program dimension and the actual size of the object.

## II. ROBOTIC WELDING SYSTEM

The components of robotic welding system are as shown in figure. An industrial robot consists of "manipulator" which moves and performs tasks, "controller" which actuates and controls the manipulator, and "programming pendant" which teaches the manipulator movement.

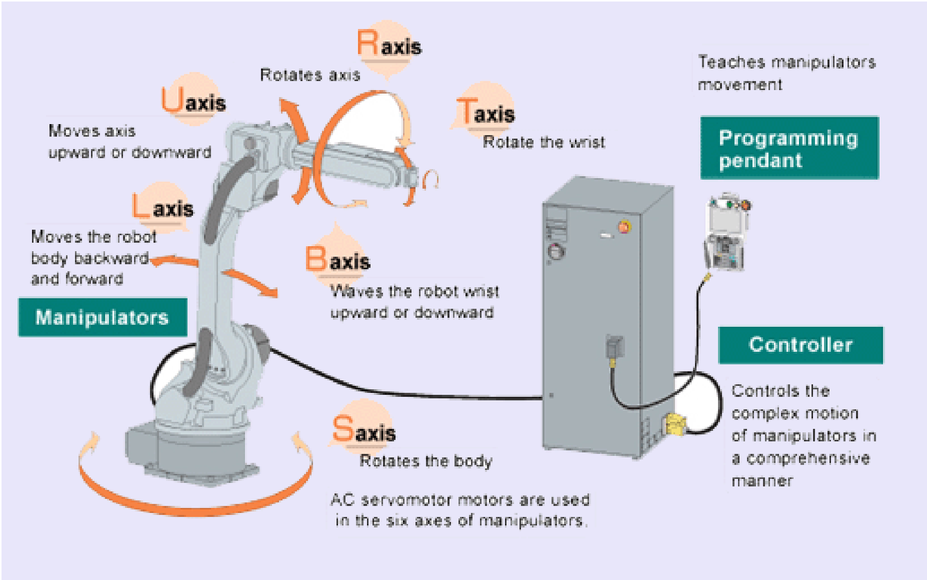



Figure 1: Components of Robotic welding system

A robot is programmed to move the welding torch along the weld path in a given orientation. The robot is typically comprised of a large number of links and linkages, which are interconnected by gears, chains, belts, and/or screws. Most of the high-end robots currently use AC servo motors which have replaced the use of hydraulic actuators & are essentially maintenance free which is very important in industrial applications. In an arc welding robot system, the torch is attached to the wrist of the robot which has two or three axes of motion.In the case of three-axis motion, the motion is composed of yaw, pitch, and roll, similar to the human wrist. The robot with three axis are used to perform the complicated welding operation.

The controller is the brain of the robot arc welding system. This is because the controller stores the robot programming and arc welding data, and performs the necessary computations for robot control, typically by a high-speed microprocessor. The controller provides a signal to the actuators and the motors by programmed data and position, speed, and other information obtained from various sensors. The controller is integrated to govern not only the robot but also any peripheral devices, such as manipulators (fixtures). When the system is required to weld a work piece that has a complicated geometry, the simultaneous coordinated control of the integrated controller is inevitable.

The welding equipment generates power to generate the arc for welding. One of the most important characteristics is stability of power. The arc sensor detects the current value so that the power source can supply the correct amount of power to the wire feeder, which then controls the wire feeding speed.Therefore, a good location for the wire feeder for a robot system is at the end of the upper arm of the robot.All the connections for welding such as electrical power, the wire, and coolant are usually integrated to one cable unit.

The programming pendant is also called as the teach pendent for the robot system. The programming pendant has all the functions required for operating and programming the robot system.

## III. REQUIREMENT OF ROBOT PROGRAMING

The earliest and most widespread method of programming robots involves manually moving the robot to each desired position, and recording the internal joint coordinates corresponding to that position. In addition, operations such as
closing the gripper or activating a welding gun are specified at some of these positions. The resulting “program” is a sequence of vectors of joint coordinates plus activation signals for external equipment. This method of robot programming is usually known as ***teaching by showing.*** The robot programming for the welding requires touch sensing,

knowledge of the coordinate system for realization of several control function, motion specifications and calibration of Tool and Base. This section discusses each of these issues and impact on the robot programming.

**Touch Sensing**

The original position of the work piece is detected using the sensor, a live welding wire. This is done by means of search instructions which are programmed via inline forms. When the welding wire touches the work piece, the circuit gets completed and 24V DC signal from the robot is returned to the robot via welding power source, fast measurement unit and controller. That means current flow is registered and at the same time, the search motion is interrupted by an interrupt signal and the position of the robot when the welding wire touches the work piece is saved.

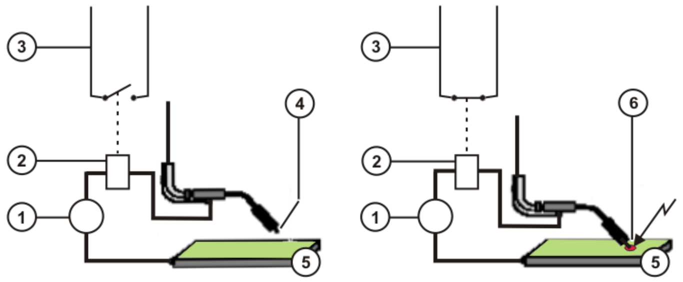
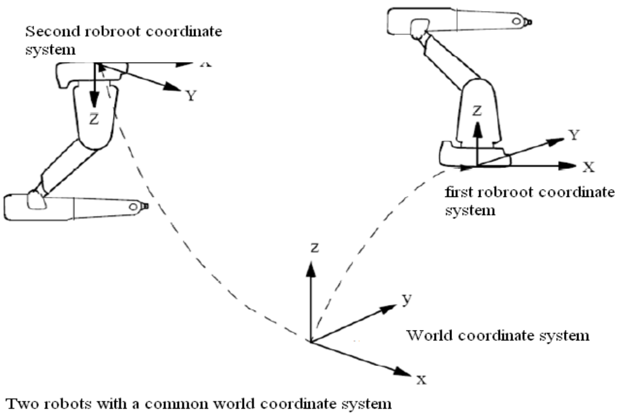



Fig 2: Functional Principle

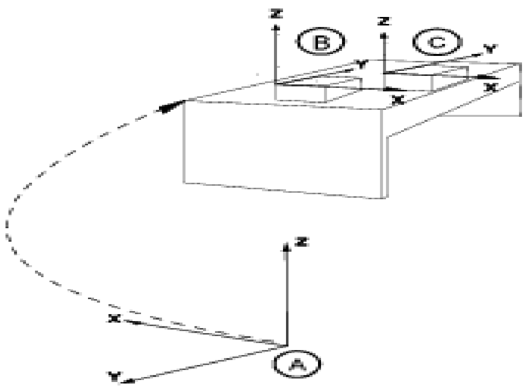
1. Welding power source
2. Relay
3. "Fast measurement" cable
4. Welding wire
5. Work piece
6. Current flow on contact

**Coordination System**

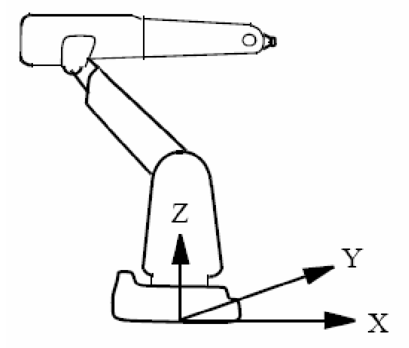
The use of coordinate system are several control functions realization, like off-line programming, adjustment of program, motion coordination of several robots and additional servo drives called as external axis, jogging motion of robot and copy of program from one robot to another.



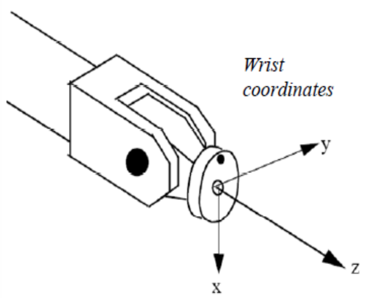

a. World coordinate system




b. Base coordinate system




c. Rob root coordinate system




d. wrist or tool coordinate system

Fig 3: Types of coordinate system

The classification of coordinate system is based on where reference point is to be considered for measurement in programing. For example, The Base coordinate system has its reference point on the work piece whereas tool coordinate system has reference point at the tool center point (TCP). The system in which the base is revolving it is preferable to use base coordinate system for robotic welding programming.

**Motion Specification**

There are three types of motion that can be used for programming trajectory of path for welding. These are point to point motion (PTP), linear motion (LIN) and Circular motion (CIRC). Out of these, the LIN and CIRC motion are termed as CP (Continuous Path) motion.

In PTP motion, the robot guides the TCP along the fastest path to the end point. The fastest path is generally not the shortest path and is thus not a straight line. As the motions of the robot axes are rotational, curved paths can be executed faster than straight paths. The exact path of the motion cannot be predicted.

In LIN motion, the robot guides the TCP at a defined velocity along a straight path to the end point.

In CIRC motion, the robot guides the TCP at a defined velocity along a circular path to the end point. The circular path is defined by a start point, auxiliary point and end point.

**Tool Calibration**

The tool calibration is meant for determination of the center point of the tool with reference to flange position which is termed as TCP. In tool calibration, the user assign Cartesian coordinate system to the tool mounted on the flange. There are total 16 tool data can be saved in the KUKA controller. The data saved are Origin of the tool (X, Y, Z) and orientation of the tool (A, B, C) with reference to the flange Position. The XYZ 4-point method and XYZ reference method are commonly used for finding the origin of the tool and ABC world method is used to find the orientation.

**Base Calibration**

The Base calibration is used for finding the reference point on the work piece from which the other point coordinate data will be saved. In Base coordinate, user mainly assigns Cartesian coordinate system to the Base or work piece which will be used for welding. The most common method for base calibration is 3-point method and Indirect method. There are total 32 base coordinate data can be saved in the robot controller.

## IV. PROGRAMING

Before discussing about the programming of robotic welding first let's define the task. The task is that the robot has to perform the welding on the tank of the distribution transformer. In distribution transformer the tank is like cube. So it has four sides on top, four sides on bottom and two sides at the vertical surface. The entire programming is mainly divided into two parts: Programming for touch sensing and Programming for welding.

### PROGRAMING FOR TOUCH SENSING

**i. Dynamical generation of the tank by teaching**

Out of 32 base coordinate system, we are going to use 28 base coordinates or touch point on the work piece as shown in fig. By using the inline forms of motion programming we can generate the programs for all 28 touch points.

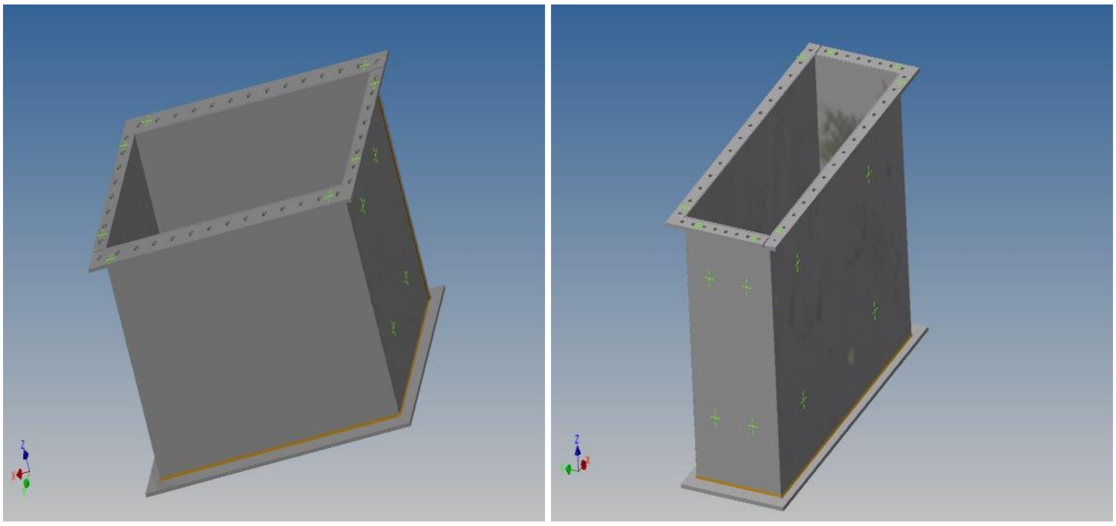


Fig 4: Geometry of the workpiece to be welding and all the touch point

The following fig 5 shows the flowchart of the programs for dynamical generation of tank by teaching the robot so that it will touch all the 28 points on the work piece.

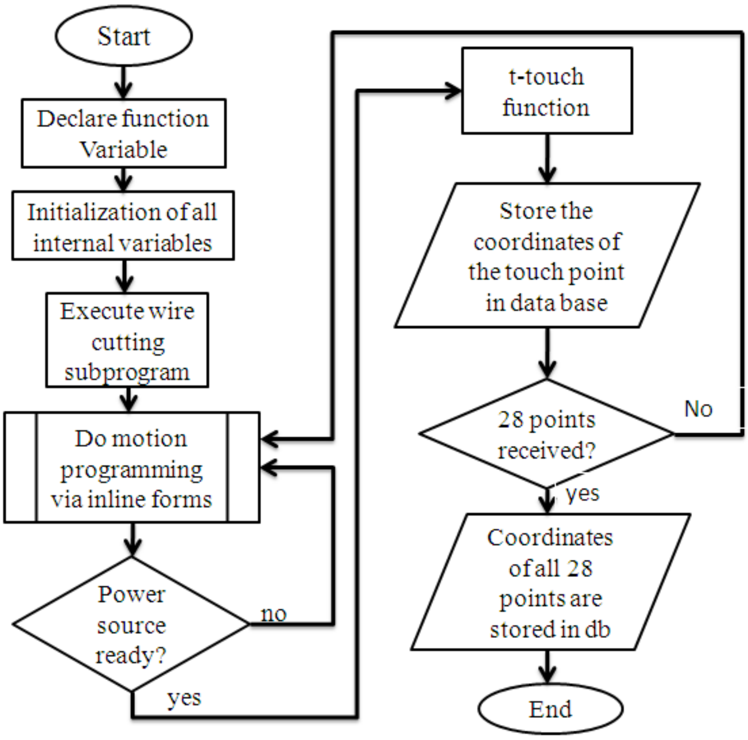



**Fig 5. Flowchart of the program for teaching the robot to touch all the points on the work piece**

**2. Making the data base of all 28 points**

After teaching all the points to the robot the next step is to make the data base which will be helpful in all other programming for dimension updating as per actual size and programming for welding. This will be done by saving all the point coordinates in data file.

**3. Programming for actual welding position by updating starting and end point**

Due to fabrication inaccuracy the actual dimension sometimes different than the programmed one. This is termed as the dimension correction. So every time before actual doing welding, the robot will update its data base. The program flow chart is as shown in fig.6.

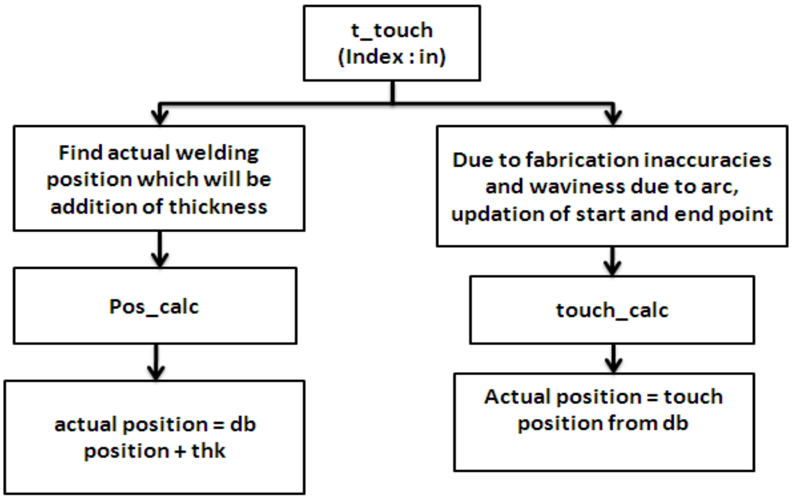
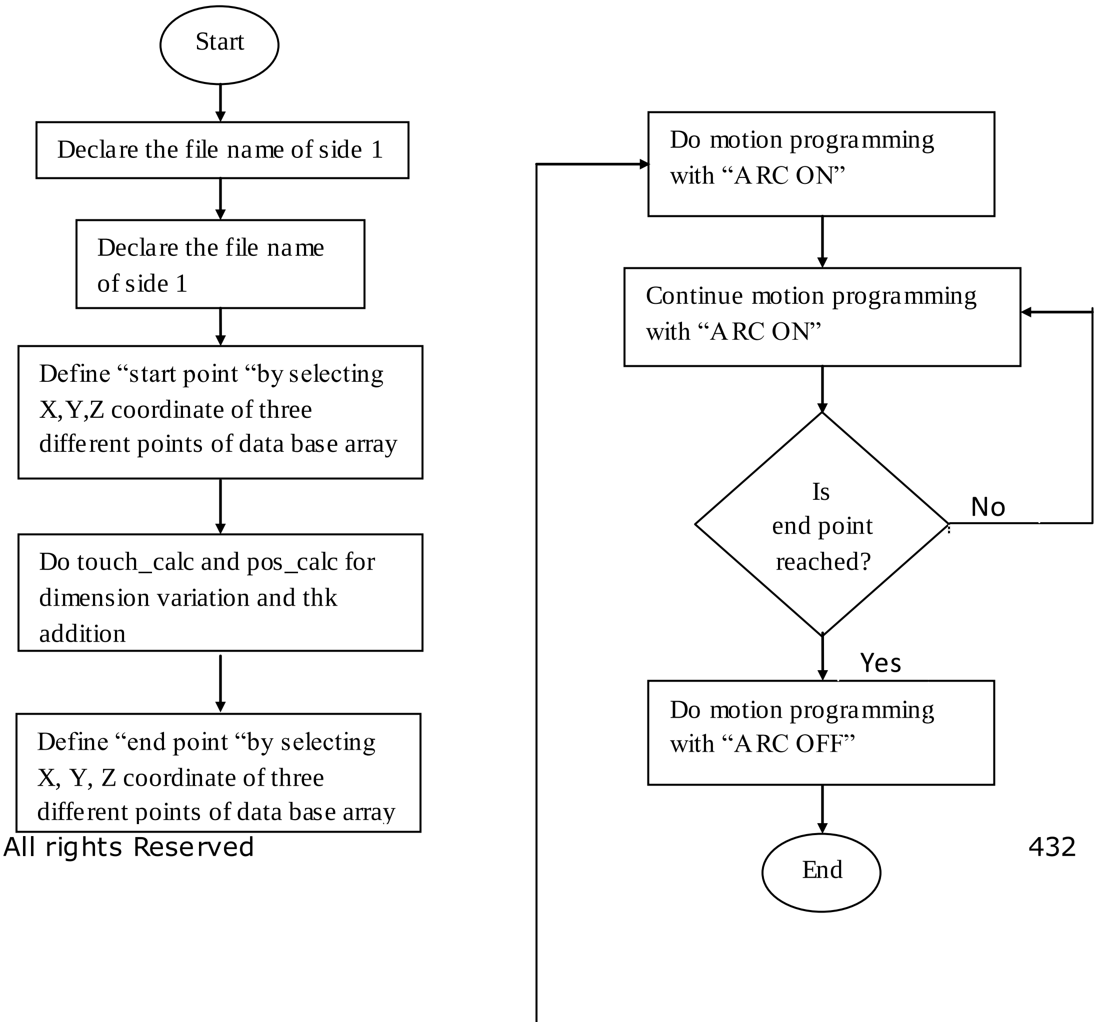



Fig 6: Programming for updating the data base

As shown in fig.6, the name of the program is the t_touch in which the inputs are from the data base array (index: in). The pos_calc is used to find the edge required to be welded which is position of the data base plus thickness of the plate of tank. The touch_calc is used to update the data base with the actual position obtained after touch.

**PROGRAMING FOR WELDING**

Like C programming, in KRL Programming concept like calling of subprogram with a particular program can be implemented. Thus complete welding program contains the subprogram of each 10 sides (4 on top + 4 on bottom + 2 longer sides). In welding program for each side, the the welding starts when Arc is ON and it is stop when Arc is OFF. Fig 7 shows the flowchart of one side of welding program. In this manner, the welding programs for all the side have to be made for complete tank welding.



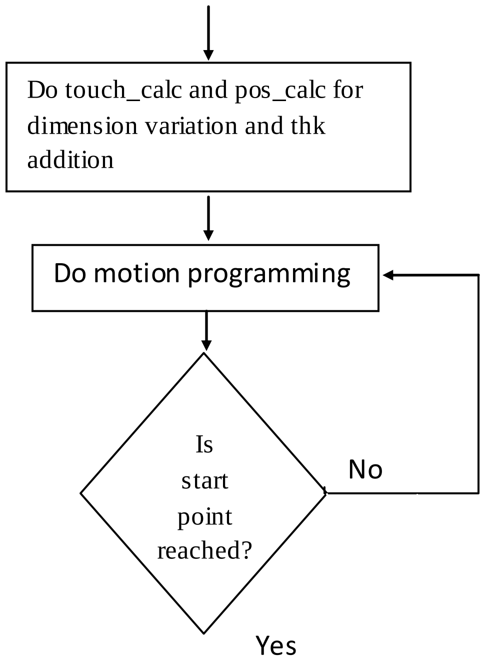



**Fig 7: Flowchart for welding program of one side of tank**

## VI. CONCLUSION

In this paper we are implementing the robotic welding programming to the distribution transformer tank. The robotic welding using touch sensing is very advantageous and approved technology in case of welding of thin plate where the problem of waviness is occurred when it is heated. The touch sensing is very cheap as compared to the camera on the robot arm that costs approximately same as that of robot. The time required for welding is very less if we use robot for welding purpose as compared to manual welding of work piece. The robotic welding with KRL (Kuka robotic language) is very easy to understand and compatible. Once the programming for one work piece is done, for other sizes of work piece, the robot automatically corrects the dimension and does the welding.

## REFERENCES

[1] KUKA System software –"Operating and programming instructions for system integrators "by KUKA, germeny
[2] KUKA.TouchSense 1.2- For KUKA System Software 5.2, 5.3, 5.4 and 5.5, Issued: 31.03.2008 Version: KST TouchSense 1.2
[3] T. Lehtla "Introduction to robotics" TTU, Dept. of Electrical Drives and Power Electronics. Tallinn, 2008
[4] Beom-Sahng Ruhr, Gordon R. Pennock," Arc Welding Robot Automation Systems" at *School of Mechanical Engineering, Purdue University,West Lafayette, Indiana, 47907-1288, USA*
[5] Craig A. Materick and Dr. Lon Shapiro, "research on Mapping robotics movement to three dimensional coordinate system" *at Illinois Wesleyan University,1997*
[6] A. Baer, C. Eastman, and **M.** Henrion, "Geometric modem: A survey," ***Computer Aided Des.,*** voL **11,** no. **5,** pp. **253-272,** Sept. **1979.**
[7] R. Bolles and R. P. Paul, 'The **use** of sensory feedback in aprogrammable assembly system," Artificial Intelligence Laboratory, Stanford University, Rep. AIM **220,** Oct **1973.**
[8] **S.** Bonner and K. G. **Shin,** "A comparative study of robot ***languages,"IEEE Computer,*** pp. **82-96,** Dec. **1982.**
[9] A. P. Ambler, R. J. Popplestone, and K. G. Kempf, "An experiment in the Offline Programming of Robots," in ***Roc. 12th Znt. Symp. on Industrial Robots*** **(Paris,** France, June **1982)** ASEA ''Industrial robot system," ASEA AB, Sweden, Rep.
[10] D. Falek and **M.** Parent, "An evolutive language for an intelligent ***robof"Zndust. Robot,*** pp. **168-171,** Sept. **1980.**
[11] J. Feldman ***et al.,*** "The Stanford Hand-Eye Project," in ***Proc. FirstZJCAZ*** (London, England, Sept. **1971),** pp. **350-358.**
[12] R. A. Finkel, "Constructing and debugging manipulator programs," Artificial Intelligence Lab., Stanford Univ., Rep AM **284,** Aug. **1976**
[13] R. Finkel, R. Taylor, R. Bolles, R. Paul, and J. Feldman, "AL, A programming system for automation," Artificial Intelligence
Lab., Stanford Univ., Rep. **AIM-177,** Nov. **1974**

[14] J. W. Franklin and G. J. Vanderbrug, "Programming vision and robotics systems with RAIL," ***SME Robots VI,*** pp. **392-406,**Mar. **1982.**

[15] General Electric "GE Allegro documentation," General Electric Corp., **1982.**

[16] C. C. Geschke, "A system for programming and controllug sensor-based manipulators," Coordinated Sci. Lab., Univ. of Illinois, Urbana, Rep. **R-837,** Dec. **1978.**

[17] G. Gini, M. Gini, R. Gini and D. Giuse, "Introducing software systems in industrial robots," in ***Proc. 9th Znt. Symp. on industrial Robots*** (Washington DC, Mar. **1979),** pp. **309-321.**

[18] G. Gin4 **M.** Gini, and M. Somalvico, "Determining and nondeterministic programming in robot systems," ***obernen'cs and Systems,*** voL **12,** pp. **345-362, 1981.**