Optimization Of Robotic Arc Welding Process By Eliminating PLC To Reduce The Overall Cost

Naveen. P [1], B. Yogesha [2] and Manish Jain[3]

1. M.Tech, Industrial Automation and Robotics, Malnad College of Engineering, Hassan-Corresponding author
2. Professor, Dept., of Mechanical Engineering, Malnad College of Engineering, Hassan
3. Head of sales and operation dept., ABB Robotics Ltd., Bangalore

Abstract

Robotic arc welding systems have become increasingly popular in industries that require speed and repeatability to maintain a high volume of productivity. When multiple fixtures are to be welded by a single robot and are controlled by multiple operator panels, the communication between the multiple operator panels and the robot controller will be generally done with the help of Programmable Logic Controller (PLC) which acts as an intermediate communication device between the robot controller and operator panel. This method leads to the additional cost investment on PLC and also the utilization of controller's main computer. The present work shows standard way for establishing communication between operator panels and main computer controller without the use of PLC. Operator panels can be directly interfaced through digital and combi I/O cards. Robot controller's CPU is programmed in RAPID programming language to handle task of PLC which leads to reduction of arc welding system cost and maximizing utilization of controller's main computer.

1. Introduction

Robot welding is performed and controlled by robotic equipment. In general equipment for automatic arc welding is designed differently from that used for manual arc welding. Robotic arc welding normally involves high duty cycles, and the welding equipment must be able to operate under those conditions. In addition, the equipment components must have the necessary features and controls to interface with the main control system. Robotic arc welding systems delivers speed; accurate air movements that helps to reduce cycle times. Robotic welding systems can repeatedly and precisely position the welding torch with an accuracy of a few thousand of an inch. In doing so, these systems deliver precise arc-on performance that exceeds the capabilities of a human welder.

Usually in automobile industries, foundries, automotive parts manufacturing industries it is regarded to reduce the total manufacturing cycle time of production robotics is employed specially for arc welding applications. To maximize the utilization of robot in welding application multiple fixtures will be loaded by the operator so that robot need not wait till operator replaces the finished fixture with new one. The process block diagram of Robot arc welding system is as shown in the Fig.1

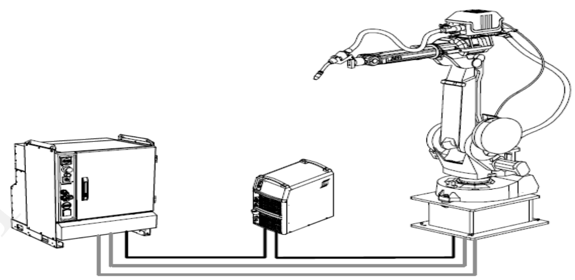


Fig.1: Process of Robot arc welding system.

The control cabinet and the robot manipulator are connected by two cables; one is thick cable that is the *power cable* that carries three phase power for driving the robot motor. The second cable is the thin cable called *SMB (Serial Measurement Board) cable* or *resolver* cable that is used for the communication between the robot and the controller, the feedback of robots angular position and velocity obtained from resolver will be sent back to controller through this cable.

In the arc welding process a special purpose power supply is used to establish the arc for welding process and is connected as shown in the Fig.1. The welding voltage and current can be set by the flex pendent to produce a perfect weld. The arc welding process is a standardized procedure in field of robotics that is implemented by all robot manufacturers in their own way, only programming the robot differs based on the application. Fig.2 shows the simple block diagram robot arc welding system that is used for welding a single fixture.

Robot arc welding system consists of a robot manipulator, robot controller, operator panel, flex pendent and external welding power source that is powered by 3 phase power supply. Flex pendent acts as the interfacing unit between user and system. The 3 phase, 415V AC power supply is connected to the controller via connectors; from their power distribution board in robot controller distributes the power to different parts of the controller and manipulator.

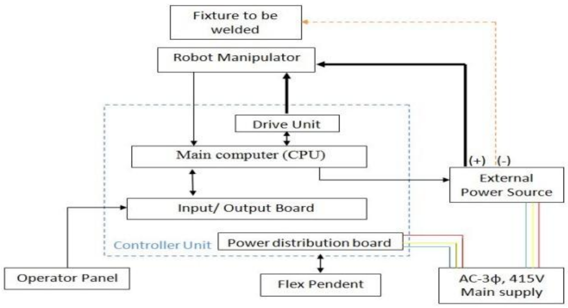


Fig.2: Block diagram of a robot arc welding system for welding single fixture.

Single operator panel is interfaced with the controller's main computer for the communication via I/O board. External power source is connected to the main computer via device net for communicating between power source and computer. Drive unit drives the motors in the robot manipulator which is controlled by main computer of the controller. To establish the arc during welding process the positive supply of the external power source is connected to the welding torch and negative to the work piece.

In present situation, when there are multiple operator panels consisting of more number of inputs and outputs signals that has to interfaced with controller, PLC plays a major role in robot arc welding system by handling most of the I/O signals, only the welding signals and motion commands are directly interfaced with the I/O boards to controller's main computer. In such systems, there are two virtual brain that are handling the system I/O signals i.e. PLC, which handles signals from different operator panel and controller main computer, which handles welding signals. The additional PLC device and programming the PLC leads to increased cost of the system.

Figure.3 shows the block diagram of a robot arc welding system with 4 operator panels interfaced with help of PLC that is being used as current system.

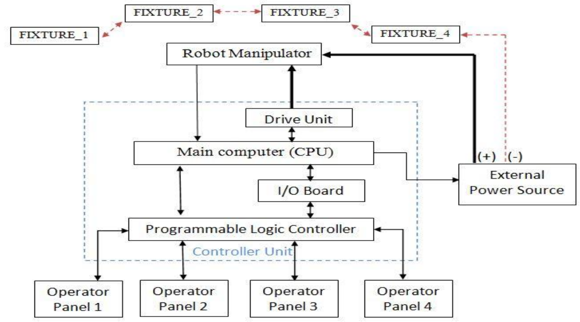


Fig.3: Block diagram of a Robot arc welding system with 4 operator panels interfaced with help of PLC

Programming language of controller will be RAPID and PLC will be programmed in ladder logic, this means same system will have to be programmed in two different languages which in turn lead to difficult for maintenance and trouble shooting. Since, PLC is handling majority of the I/O signals this will result in under usage of Controller's main Computer.

2. System overview

2.1 Architecture of the system

Robot arc welding system consists of the following parts: robot manipulator (IRC-1410), robot controller (IRC5), flex pendent, welding power source, Wire feeder, welding torch as an end effector of the robot, four fixtures that have to be welded and four individual operator panels for the operator to communicate with the robot arc-welding system.

The proposed solution is to eliminate the PLC and to use multiple I/O boards to interface I/O signals directly to controller CPU and program the system to directly latch these signals. All the functions that were performed by the PLC will be done by the controller main computer and this will leads to increased usage of CPU and the elimination of the PLC leads to reduction of overall cost of the system. Fig.4 shows the proposed block Diagram for the arc welding system with 4 operator panels interfaced directly with help of I/O board.

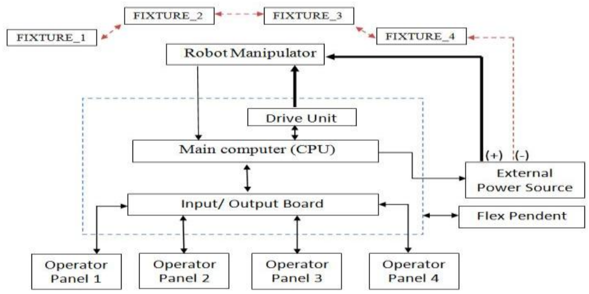


Fig.4: Block diagram of arc welding system with 4 operator panels interfaced directly with help of I/O board.

2.2. Tasks to be accomplished

1. Welding priority should be based on first come first serve basis.
2. Avoid the error inputs given by the operator negligence.
3. Providing common stop (pause) to stop the welding process.
4. Providing individual resume to resume the stopped process from the given fixture.
5. System must identify the present shift and also should monitor the change of shift.
6. Display of Production rates on demand.
7. At the each shift production rates have to save automatically in a log file created for each fixture and the production rate should be rested to zero for next shift.
8. In case of maintenance or emergency an option input should be provided to save the production rate of that shift in a log file manually.

3. Programming logic

3.1 Logic for First Come First Serve (FIFS)

In welding cell normally there will be multiple operators who will be working on different fixtures. The basic requirement is to provide priority to the operators based on first come first serve. In additional to that possible error inputs given by the operator should be avoided. The logic for the same is as shown in the Fig.5.

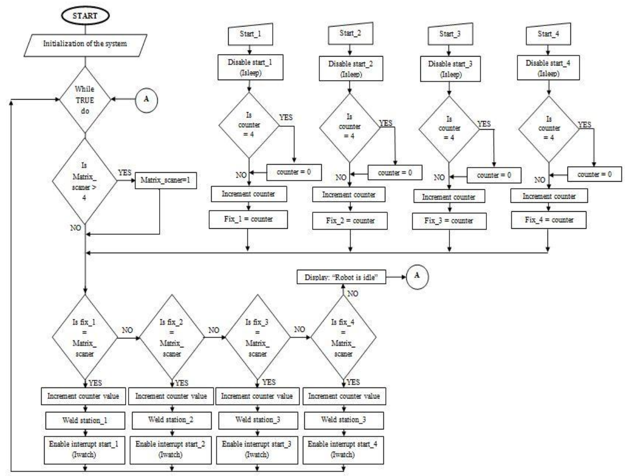


Fig.5: Logic for First Come First Serve (FIFS)

Algorithm: Logic for First Come First Serve (FIFS)

1. Start the system.
2. Initialize the system.
3. Create a endless while loop in which following conditions are checked;
- Matrix_scanner > 4, if matrix_scanner is > 4, then make it = 1.
- Register fix_A = matrix_scanner value.
- Register fix_B = matrix_scanner value.
- Register fix_C = matrix_scanner value.
- Register fix_D = matrix_scanner value.

Input side

There are 4 input (*cycle start*) from 4 operator panel, when high (manually pressed), a Trap routine will be executed.
- Disable the interrupt particular interrupt (Isleep) until corresponding weld cycle is completed.
- Check: counter value > 4. if counter value is > 4, then make it = 0.
- Counter value will be incremented by 1.
- Particular reg_fix_value will be made equal to counter value.

4. When the counter value = reg_fixt_value, then counter value will be incremented, particular weld routine is called and executed.
- Increment the counter value.
- Make the Reg_fix_value = 0.
- Perform the welding.
- Enable the particular interrupt.
- Return to step3.

5. When there is no input provided form operator a message will be displayed "ROBOT IS IDLE" and Return to step3

3.2 Shift identifier

At the start program, based on the time, the system will identify which shift it is working on and it will Display it on flex pendent. The logic for the same is as shown in the Fig.6.

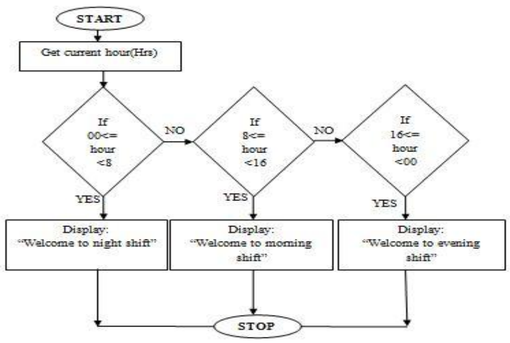


Fig.6: Logic for shift identifier

Algorithm: Shift identifier
1. START
2. Read the current Hour.
3. Check;
   - If the hour is between 00th to 08th then, Display "NIGHT SHIFT"
   - If the hour is between 08th to 16th then, Display "MORNING SHIFT"
   - If the hour is between 00th to 08th then, Display "EVENING SHIFT"
4. STOP

3.3 Monitor shift change

In the industries it is important to keep a track of change of shift. The logic for the same is as shown in the Fig.8.

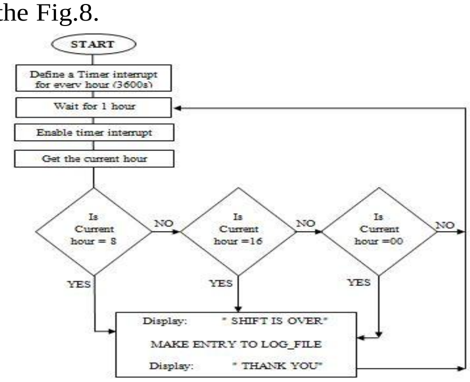


Fig.7: Logic for Monitor shift change

Algorithm: "Monitor shift change"
1. Start
2. Define a timer interrupt that will be activated every 3600 sec (1 hour).
3. Wait for 1 hour, to pass by
4. Get the current hour
5. Check for;
   - Is the hour 08th
   - Is the hour is 16th
   - Is the hour is 00th
6. If any of the condition is satisfied then display: "SHIFT HAS ENDED"
7. Make the log_file entry and display: "Thank You"
8. Return to Step 3.

3.4 Display production rate

In the manufacturing industries there is a timely requirement of knowing the production rate during the particular shift so it is important to provide an option to display production rate. The logic for the same is as shown in the Fig.7.

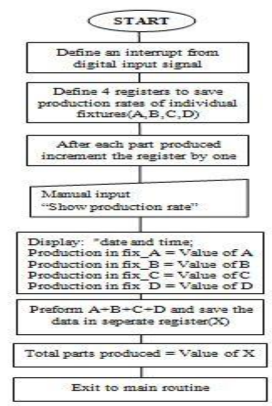


Fig.7: Flow diagram to display production rate

Algorithm: "rate of production"
1. Start
2. Define a digital input signal to display production rate.
3. Define 4 registers whose values will be incremented after each part has been produced.
4. On demand Display the production rate of individual fixture.
5. Total production = sum of production rate in individual fixtures.
6. Display total production rate.
7. Exit to main routine.

3.4 Display production rate

In the manufacturing industries there is a timely requirement of knowing the production rate during the particular shift so it is important to provide an option to display production rate. The logic for the same is as shown in the Fig.8.

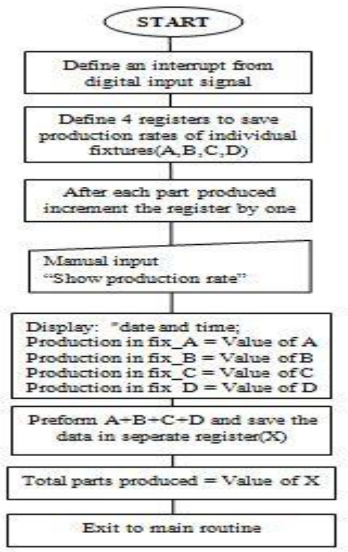


Fig.8: Flow diagram to display production rate

Algorithm: "rate of production"

1. Start
2. Define a digital input signal to display production rate.
3. Define 4 registers whose values will be incremented after each part has been produced.
4. On demand Display the production rate of individual fixture.
5. Total production = sum of production rate in individual fixtures.
6. Display total production rate.
7. Exit to main routine.

3.5 Auto log file entry

In most of the manufacturing industries it is a very essential thing to maintain a record of produced parts in each shift. It is possible to record this data for a long period of time and retrieve it whenever it is needed. The logic for the same is as shown in the Fig.9.

Algorithm: "Auto log file entry"
1. Start
2. Define a routine to update the log file automatically.
3. At the end of the shift call the routine.
4. Fetch the current hour
5. Print the production data along with the time and date.
6. Clear the production data.
7. Display: "Auto entry of production rate successful".
8. Exit to main routine.

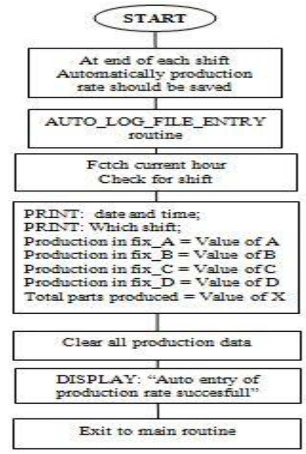


Fig.9: Flow diagram for auto log files entry

3.6 Manual log file entry

During emergency as well as maintenance period the log file entry procedure has to be manually performed. The logic for the same is as shown in the Fig.10.

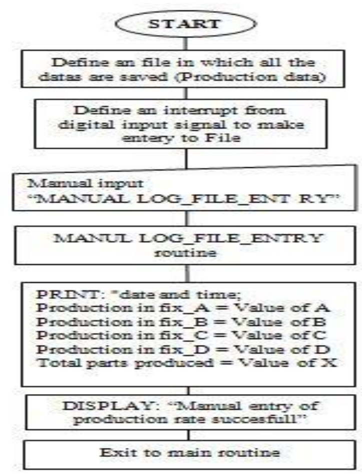


Fig.10: Flow diagram for manual log files entry

Algorithm: "Manual log file entry"
1. Start

2. Define a routine for manual log file update.
3. Define an input signal to call the manual log file update.
4. Define a routine to update the log file automatically.
5. Fetch the current hour
6. Print the production data along with the time and date.
7. Display: "manual entry of production rate successful".
8. Exit to main routine.

4. Results

System is started in AUTO Mode and the program is played, the initialization will be started robot will display date, time, and a welcome message, controller checks for home position of the robot. Robot will wait for the inputs. The displayed messages are as shown in Fig.11.

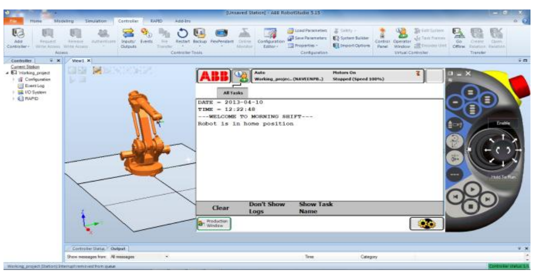


Fig.11: Flex pendent displaying an initialization message on flex pendent.

Inputs are directly provided for the controller form operator panel, while performing simulation inputs are provides in Input/ Output window as shown in Fig.12

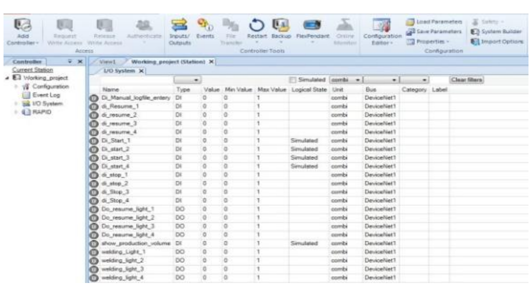
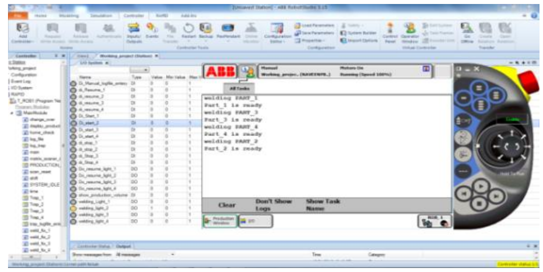


Fig.12: Input/ output window where status of the signal simulated.

When the inputs are provided to the random way, the controller saves the input in first come first serve method and will deliver the work in same order. Flex pendent in the Fig.13 is displaying the work delivered in first come first serve method

Fig.13: Flex pendent displaying order in which work is delivered



For testing a random order input was provided as, start_1, start_3, start_4, and start_2. System saved these works and delivered in first come first serve order. Similarly number of experiments was performed on the system in different order for a longer time and system has performed well under all conditions.

Figure.14 shows the flex pendent displaying the production rate when production rate is demanded. Figure.15 Flex pendent displaying notification of updating log file with production rate automatically and manually when needed.

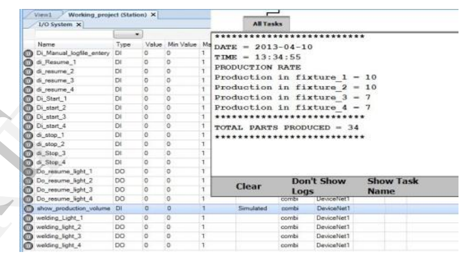


Fig.14: Flex pendent indicating the production rate.

5. Conclusion

In this work, optimization of arc welding system has been achieved by eliminating PLC. The tasks handled by the PLC are now directly latched to the controller's main computer which has resulted in increased usage of CPU. The elimination of PLC has resulted in cost reduction of the system, performance of the system remains same as earlier when used PLC. Robot ARC welding system has been provided with additional features such as shift identifier, shift monitoring, display of production rate, manual and auto log file update of production data.

6. Acknowledgments

At the outset I would like to express my gratitude to my guide **Dr. B. Yogesha**, professor, Department of Mechanical Engineering, Malnad College of Engineering, Hassan, for his guidance and support for the successful completion of work and **Mr. Manish Jain**, Technical Head, ABB Robotics Ltd, Bangalore, whom I thank for becoming my guide and for his keen interest, valuable and excellent guidance, continuous encouragement and infinite patience during the project work.

7. References

[1]. Pires, J.Norberto, "Welding robots", Dept. of Mech. Eng., Coimbra Univ., Portugal, june 2003
[2]. N. Pires, "Industrial Robots Programming, Building Applications for the Factories of the Future", Springer, New York, USA, 2006.
[3]. Pedro Neto, J. Norberto Pires, "High-Level Programming for Industrial Robotics: using Gestures Speech and Force Control" Springer, New York, USA, 2006.
[4]. Courtesy:http://www.abb.com/products/IRC5_manual/data_sheet/html
[5]. Courtesy:http://www.abb.com/products/IRB1410_manual/data_sheet/html.