Implementation of an Intelligent Robotized GMAW Welding Cell, Part 2: Intuitive visual programming tool for trajectory learning

I. Lopez-Juarez[1], R. Rios-Cabrera[1] and I. Davila-Rios[2]
[1]*Centro de Investigacion y de Estudios Avanzados del IPN*
[2]*Corporacion Mexicana de Investigacion en Materiales SA de CV*
Mexico

1. Introduction

Robotized GMAW welding is a demanding process. Current robots are able to perform welding tasks continuously under different working conditions in low-scale production such as shipbuilding or in high-scale production such as in the automotive industry. In well defined and structured environments such as in the automotive industry robot reprogramming is still necessary in order to cope with uncertainties. This additional task involves hiring specialized personnel, lost of production time, quality assessment, destructive testing, etc., which necessarily increases the production costs.

During the welding task, the joint part specification has to be met in order to meet the desired quality and productivity in industry. However, there are several factors that affect the process accuracy such as welding part positioning; motion errors in the production line, mechanical errors, backlash, ageing of mechanisms, etc. which are error sources that make robots to operate in uncertain conditions, i.e. unstructured environments. The scope of this work is focused on the compensation of these stochastic errors generated during the process and that the robot system needs to cope with in order to meet the required quality specification. To reach this goal, it is required to have an appropriate test-bed integrated with the process parameters sensing capacity (laser system, camera, proximity sensors, etc.) to follow the welding bead and to provide robust information to the robot controller. The use of multiple sensors and different computers make a centralised control very complex, hence it is preferred the use of the CORBA specification to implement a Distributed System. In this chapter we present the machine vision system and the distributed control for the welding cell as well as the Human-Machine Interface (HMI) developed to "teach" the manipulator any welding trajectory.

[1] This work was carried out during Dr. Lopez-Juarez research visit at Corporacion Mexicana de Investigacion en Materiales SA de CV (COMIMSA) under General and Specific CINVESTAV-COMIMSA Collaboration Agreements.

The robust design solution as proposed in this research is a two-fold issue. First, it is necessary to minimize design errors by simulating the whole welding process considering issues like floor plant space, robot configuration, welding equipment and supplies, etc. and second, the utilization of a novel teaching tool for welding trajectory. The contribution of this research has been split in two parts: In part I, the robotic cell set up (including off-line and on-line programming) using current 3D software simulation, voice command simulation design, equipment commissioning and testing was presented; whereas in this Part II, a novel robot programming tool for teaching welding trajectories and a built-in error compensation during production of welded parts are presented.

The programmed tool for trajectory learning is implemented in a Visual C++ application named *StickWeld V1.0* that involves the use of a friendly Graphical User Interface (GUI) for trajectory *compensation* and *teaching*. The software runs in a PC-based computer and uses a top mounted fast Firewire Colour Camera, a wireless gamepad and a pointing stick. The purpose of the software is to *compensate* on-line any error misalignment during perimeter welding of flat metal parts. The system compensates any offset error in the robot's welding torch due to conveyor or line production transport errors. The misalignment is captured by the camera and the image is processed in the server computer to find the new perimeter information, which is translated into a new robot trajectory and sent to the robot controller for execution. The *teaching* option can also be accessed via the GUI; by selecting this option the teaching/learning mode is activated. While in this mode, the user can define any welding trajectory using a stick as a pointer to define the trajectory. The trajectory input data, parameters selection and the robot motion control is made through the wireless gamepad controller so that the user has always full motion control on the robot assuring the safety within the cell. Once the new trajectory is entered, the robot can repeat the operation in another metal part. The system uses a three layer communication structure. The lowest layer is the serial standard communication RS232, followed by the SIEMENS 3964r protocol and at the top are the *StickWeld* commands that communicate the host PC master controller with the KUKA KRC2 robot controller. The defined trajectory path is stored continuously in a processing FIFO allocation in order to have a continuous interpolated motion at execution time.

The organisation of the chapter is as follows. This introduction belongs to section 1, which also includes the description of the distributed system and related work. In section 2, the GMAW welding process and different subsystems of the workcell are explained. In section 3, issues concerning with the server-robot communication protocol are provided. In section 4, the program *StickWeld V1.0*, GUI and the use of the peripherals and programming modes are described in detail. Finally, in section 5 conclusions are provided and the envisaged future work is highlighted.

1.1 Distributed System and Related Work

The CORBA specification (Henning, 2002), is developed by the OMG (Object Management Group), where it is specified a set of flexible abstractions and specific necessary services to give a solution to a problem associated to a distributed environment. The independence of CORBA for the programming language, the operating system and the network protocols, makes it suitable for the development of new application and for its integration into distributed systems already developed. In this investigation, it was decided to implement CORBA due to previous experience in Robotic Assembly (Lopez-Juarez & Rios-Cabrera, 2006). For a comprehensive description of the specification as well as its integration in the workcell, the reader is referred to (Davila-Rios, et al., 2009), where the distributed system is described in detail.

Jia proposes robotized systems using CORBA as the communication architecture in telerobotic applications like in virtual laboratories, remote maintenance, etc. (Jia, et al., 2002). Other authors look more at new paradigms rather than interoperability at the object level, but at the service level to facilitate the interoperability of industrial robots in the service environment and what has been called: Service Oriented Architectures (SOA) (Veiga, et al., 2007). Other authors use simple I/O devices like digitising pens to facilitate robot programming (Pires, et al., 2007). In our case, we have taken ideas from the mentioned authors and we have implemented a novel distributed programming tool to teach a robot random welding trajectories.

2. Welding process and Robotic System

Gas Metal Arc Welding (GMAW) is a welding process which joins metals by heating the metals to their melting point with an electric arc. The arc is between a continuous, consumable electrode wire and the metal being welded. The arc is shielded from contaminants in the atmosphere by a shielding gas.

GMAW can be done automatic by using an industrial robot manipulator as it is the case in this research and without the constant adjusting of controls by a welder or operator.

Basic equipment for a typical GMAW automatic setup are:

- Robot manipulator
- Welding power source: provides welding power.
- Wire feeders (constant speed and voltage-sensing): controls the supply of wire to welding gun.
- Constant speed feeder: used only with a constant voltage (CV) power source. This type of feeder has a control cable that will connect to the power source. The control cable supplies power to the feeder and allows the capability of remote voltage control with certain power source/feeder combinations. The wire feed speed (WFS) is set on the feeder and will always be constant for a given preset value.
- Voltage-sensing feeder: can be used with either a constant voltage (CV) or constant current (CC) - direct current (DC) power source. This type of feeder is powered by the arc voltage and does not have a control cord. When set to (CV), the feeder is similar to a constant speed feeder. When set to (CC), the wire feed speed depends on the voltage present. The feeder changes the wire feed speed as the voltage changes. A voltage sensing feeder does not have the capability of remote voltage control.
- Supply of electrode wire.
- Welding gun: delivers electrode wire and shielding gas to the weld puddle.
- Shielding gas cylinder: provides a supply of shielding gas to the arc.

When this process starts, the weld pool is shielded by an inert gas, giving the process the popular designation of Metal Inert Gas (MIG). Nowadays actives gases such as carbon dioxide or mixtures of inert and active gases are also used and the designation GMAW includes all these cases. This process is widely used in industrial application due to its numerous benefits. It can weld almost all metallic materials, in a large range of thicknesses (above 1 mm up to 30 mm or more) and is effective in all positions. GMAW is a very economic process because it has higher deposition rates than for example the manual metal arc process, and does not require frequent stops to change electrodes, as is the case of this former process. Less operator skill is required than for other conventional processes because electrode wire is fed automatically (semi-automatic process) and a self-adjustment mechanism maintains the arc length approximately constant even when the distance weld torch to work-piece varies within certain limits. These advantage make the process very well adapted to be automated and particularly to robotic welding applications. The process is sensitive to the effects of wind, which can disperse the shielding gas, and it is difficult to use in narrow spaces due to the torch size (Holliday, D B 2005).

2.1. Robotized Welding System

The welding system used for experimentation is integrated by a KUKA KR16 industrial robot. It also comprises a visual servo system with a ceiling mounted Basler A602fc CCD camera as it is shown in figure 1.

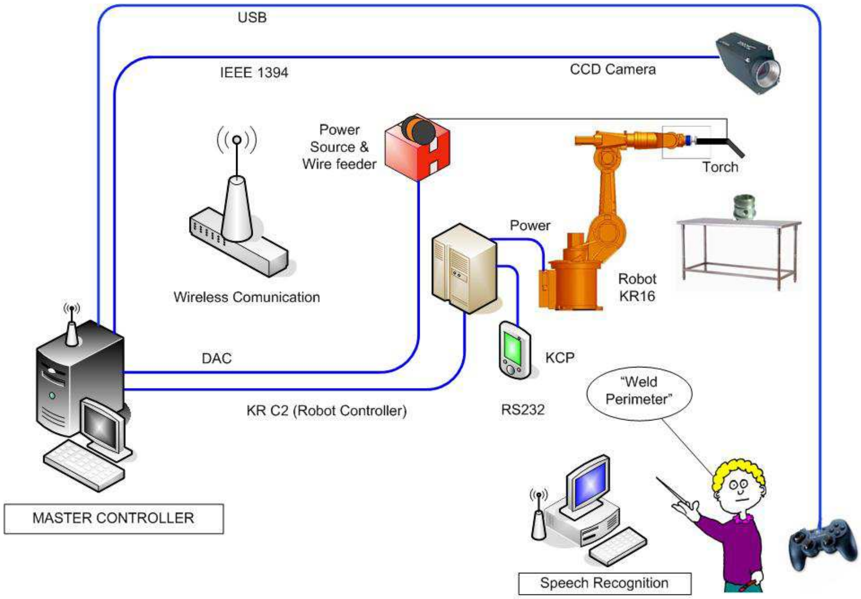


Fig. 1. Robotized Welding System

Two computers are used, the Master Controller and the Speech Recognition computer. The Master Controller is a PC Intel Xeon a 1.86GHz with 3GB RAM in charge of low-level serial communication with the robot controller using the 3964a protocol. It also connects to the Lincoln 455M power source and 10R wire feeder using an I/O Data Acquisition Card so that the welding process can be switched on-off and the current and voltage can be controlled by this computer. Additionally, it also handles the programming user-interface through a wireless gamepad. On the other hand, the Speech Recognition computer is in charge of giving voice commands to the robot in order to carry out the welding tasks.

The cell is based on the CORBA omniORB 4.1 Open Source GNU. The distributed system is designed to work within a wireless network. The elements of the wireless distributed system are described in the following sections.

2.1.1 Master Controller

The master controller connects to four subsystems as it could be seen in figure 1, namely:

KUKA KRC2 Robot Controller
GMAW Power source and wire feeder
Vision System
Programming Tool for Trajectory Learning

Each of these subsystems is explained below.

KUKA KRC2 Robot Controller

The KUKA KR16 robot is used in slave mode. Its motion is directed in low-level using the 3964a protocol and the RS232 serial communication. During operations a slave motion program in KPL runs in the KRC2 controller. This motion program is in charge, among other options, of the arm incremental motions, selection of tool/world coordinates, motion distance and speed. The other communication program resides in the Master Controller and forms part of the Stick Weld application.

GMAW Power source and wire feeder

The control and communication between the master controller and the power source and wire feeder is established using a general purpose DAC Sensoray 626. This is illustrated in figure 2. This card connects to the power feed to switch ON/OFF the power. It is also possible to modify the voltage and current. This was achieved by replacing the encoders from the voltage and current controls of the 455M Power Feed and to emulate its operation using a PIC microcontroller. The microcontroller receives a voltage or current data from the welding application and it translates in Gray code and send its data to the power feed, which in turn controls the wire feeder to the welding gun.

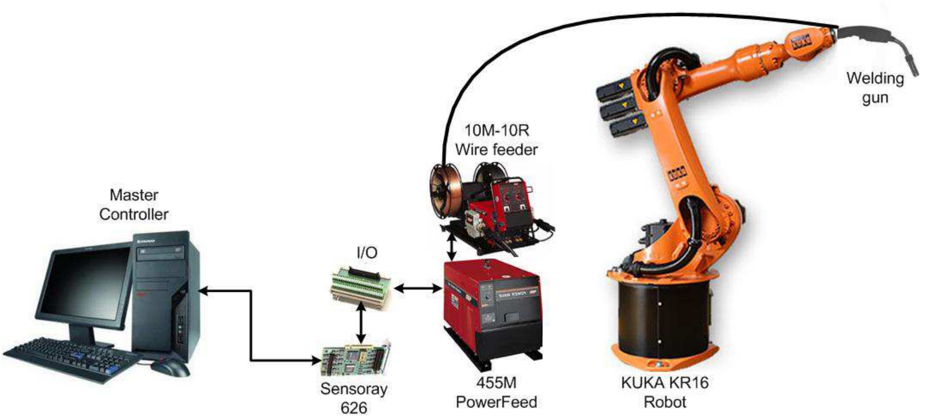


Fig. 2. Welding System

Vision System

The vision system was implemented using a fast Firewire Camera Basler A602fc with a 1/2" CCD sensor with 656 x 490 pixel resolution and using a IEEE 1394 bus. The application program for the vision system was developed in C++ using Visual Studio 2005 and using the SDK from Basler. This application is embedded into the Master Controller.

During welding tasks, the vision system is used to recognise welding trajectories. The distributed system defines several methods to be used that are available for other clients/server elements. The IDL functions that are available from other elements are (See (Davila-Rios, et al. 2009) for further details):

- Functions for robot motion
- Robot status
- Workcell status
- Force/Torque sensing
- Available welding trajectories for clients
- Modification functions for the welding power source
- Welding machine status
- General parameters
- Others.

The camera system is used to input data to program the robot new trajectories and to correct any misalignment during part welding but also it is being used as an input to the object recognition application. This object recognition is a developed application for 2.5D object recognition, complete details of the algorithm and development is explained further in this book (Lopez-Juarez, et al., 2009).

The server share methods for other object recognition through the following IDL methods:

- Object recognition
- Data from the recognised object
- Robot training execution
- Data from the training task
- Others

Programming Tool for Trajectory Learning

The master controller computer also uses the vision system, a wireless gamepad and a teaching tool (pointing stick) to train the robot new welding trajectories. In terms of the IDL functions there are some elements that are used for teaching purposes.

Industrial Robot KUKA KR16

The robot only works as a service provider. The master controller has access to its general parameters such as speed; world/tool coordinates selection, motion axis, etc. The communication is serial and there are no IDL functions, since it is not a CORBA client or server, but it is used by a server (the master controller).

Speech recognition

This system is based on the Microsoft SAPI 5.0 and works with Context Free Grammar. It generally works as a client making requests to the other systems. The IDL functions mostly interact with the robot using the following services:

- Speech recognition initialisation
- Stop the robot
- General parameters

The speech recognition system can be used directly to control the workcell or to initialise a specific process within the cell. The reader is referred to (Davila-Rios, et al., 2009) for further details on the implementation of the speech recognition system.

Additionally, the robot controller also can use another system for part location purposes. The workcell includes a Hytrol TA model belt conveyor, which is speed/position controlled by a Micromaster 420. The belt conveyor IDL functions are:

- Belt motion with X speed
- Status from the current location
- General parameters configuration

The belt conveyor only accepts directions and provides current status. This is considered a server only since it never acts as a client.

The issues concerning with the server-robot communication protocol are described in detail in section 3. In section 4, the program StickWeld V1.0, GUI and the use of the peripherals and programming modes are explained.

3. Programming and communication

The welding software is connected through CORBA to the other servers (the other modules of the manufacturing cell) as it was showed in figure 1. But a complete real-time communication is hold as well with the robot controller in order to access the robot movements.

In order to obtain continuous robot movements it was necessary to implement a stack of communication protocols. As it can be seen in figure 3, the welding SERVER (Master Controller) holds both, a CORBA module, and a set of functions to control the robot. The CORBA module embedded in the Stick Weld system gives access to other modules of the cell to manipulate some robotic tailored functions. This allows for example to receive direct commands from the speech module and to execute them.

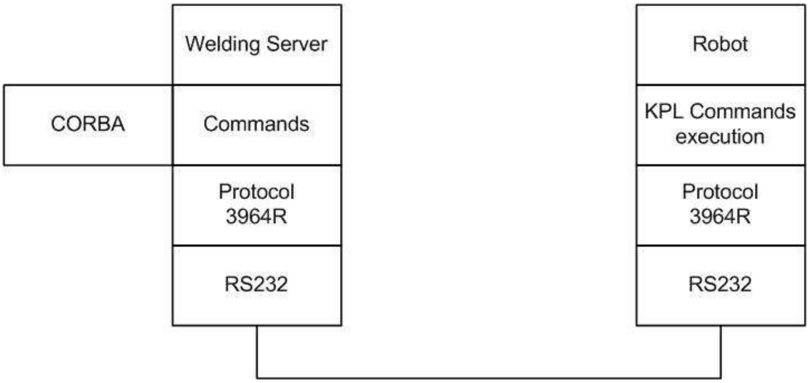


Fig. 3. Stack of communication protocols Master Controller PC - KUKA KR16

In the lowest layer, the RS232 standard is used for communication protocol. The second layer uses the SIEMENS 3964R protocol. The third layer contains the basic commands definition to be executed in the robot.

Some of these commands are presented in table 1.

Functions
Go Home
Coord World
Restart FIFO
AddMovement (x,y,z,speed)
Start FIFO processing
Stop FIFO processing
From current+1 delete all FIFO
Emergency Stop
DefineApproximation
SimpleMovement(x,y,z,speed)
SimpleRotation(x,y,z,speed)
Exit program

Table 1. Basic functions available in the communication.

In order to create a continuous movement of the robot, interpolation is carried out using a FIFO structure in the robot controller programming. This structure maintains the movements until a buffer is empty, or until a stop command is received. This could be an emergency stop for example or other stop function. The internal programming uses different interruptions to achieve continuous movements while receiving other commands at the same time, and to generate soft movements of discrete coordinates.

4. Stick Weld 1.0

Stick Weld 1.0 is a beta software project, presented as a prototype to create a functional powerful tool to teach easily different trajectories to an industrial robot. A user interface was developed containing two main functions:

1. It allows the robot to follow the contour of irregular or regular metal objects to be welded.
2. It allows a real-time robot programming tool to follow and weld random paths on flat surfaces.

To execute these tasks, an industrial robot is used, as well as different basic programming elements to teach the robot the desired trajectories. It includes basically: Fast colour camera, pointing stick, a gamepad, and a welding table. These elements are showed in figure 4.

In order to keep the programming tools simple, the robot is taught using an interface consisting of a pointing stick and a wireless gamepad. With the pointing stick, the user only has to move the stick on the metal part using the desired trajectory. The camera captures the 2D trajectories that the robot must follow later, executing the welding task. When the teaching is being done, the vision system tracks the movements and records the Cartesian coordinates of the pointing stick as illustrated in figure 5.

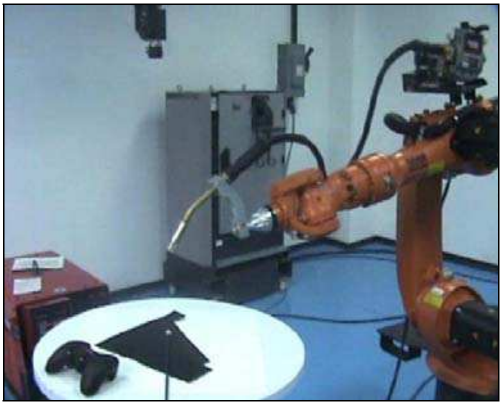


Fig. 4. Basic elements for trajectory programming

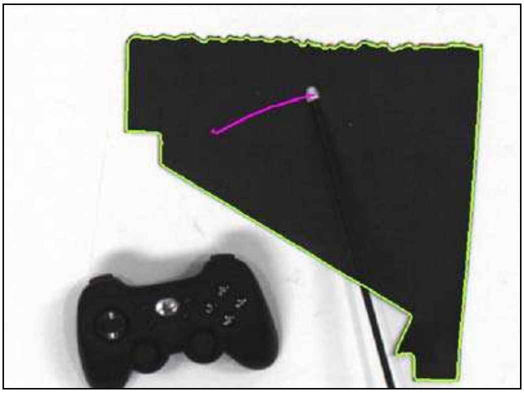


Fig. 5. Tracking of the pointing stick

All main functions are included in the buttons of the gamepad, to provide a complete useful interface. Among these functions, the start-finish robot movements are included as well as an emergency stop. Once the programming is finished, it is intended that the vision system, must recognize a specific piece, and also recall the already programmed objects (including all trajectories and configurations) and apply that in the task execution.

While scanning the object to be welded, all the contour, holes and special forms founded in the piece are recorded and converted to coordinates that the robot could follow if necessary. An example is provided in figure 6 where two embedded contours were identifies and recorded. In the upper part of the piece desired trajectories are computed. The vision system also has the option to work with different background colours. As showed in figure 6, the system works with a white metal surface and in this case a white pointing stick with a black tip was employed.

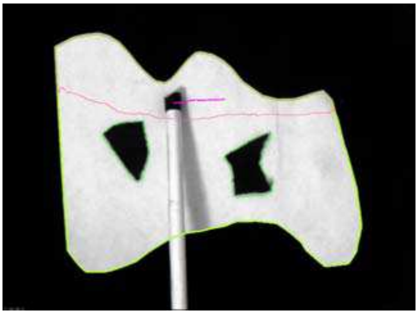


Fig. 6. Automatic detection of contours

Figure 7 shows a main screen of the software showing the designed GUI. There is a direct access to different configurations of the visual programming. The button: *Mover Robot*, execute different manual movements such as Go Home Position, Rotation in Z direction, etc.

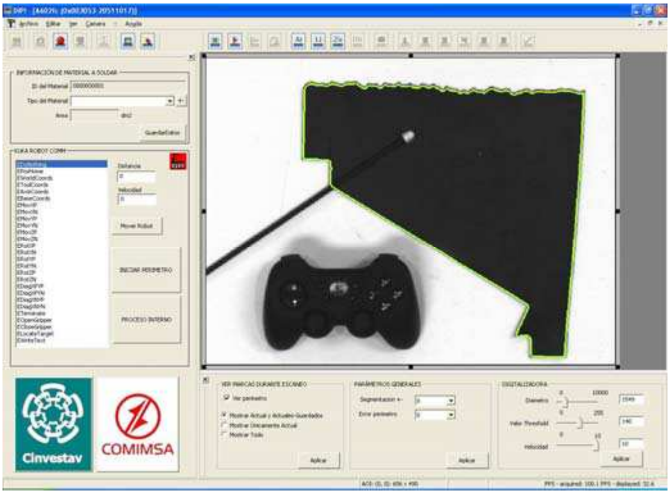


Fig. 7. Main Program Interface

It has two inputs, *Distancia* given in millimetres and *Velocidad* (Speed), which is given in m/s. The interface has also two buttons for automatic processing of the already recorded trajectories. The first one is *Iniciar Perimetro* (Start Perimeter), which starts the communication with the robot to follow the trajectory of a workpiece perimeter for welding purposes. The second button *Proceso Interno* (Internal Process), gives the instructions to the robot of welding all internal pre-recorded trajectories in the current object. Those options can be triggered also from the wireless gamepad.

The general configuration includes, diameter of pointing stick, threshold values, and scanning speed for the movements of the stick. For the programming of trajectories, some geometrical predefined figures have been programmed, such as closed areas, open lines, and zones. Also a visual configuration is available to help introducing new trajectories of a specific type (i.e. closed area, line). These are identified in the upper toolbar as Ar, Li and Zo.

There are other features that can be accessed via the menu bar or the upper left toolbar. They are intended to modify the camera settings such as brightness and shutter speed and other options such as taking a single snapshot or taking image data continuously.

5. Conclusions and further work

During welding tasks there are quality specifications to be met. However, there are several factors that affect the process accuracy such as welding part positioning; motion errors in the production line, mechanical errors, backlash, ageing of mechanisms, etc. which are error sources that make robots to operate in uncertain conditions, i.e. unstructured environments. The scope of this work has been focused on the compensation of these errors generated during the welding process and that the robot system needs to compensate automatically.

The proposed solution includes the development of the *StickWeld V1.0* application which is a windows based solution programmed in Visual C++ that uses a CCD camera, Basler SDK, wireless gamepad and pointing stick as a teaching tools. The system also uses IDL functions so that the manipulator can receive verbal instructions such as motion commands or start/stop the task.

The developed user interface contains two main functions. One operation is to follow the contour of irregular or regular metal objects to be welded and the other is to follow and weld random paths on flat surfaces.

Simple welding trajectories were tested using the KUKA manipulator as shown in figure 8. Several issues rose from the accomplished welding tasks such as starting point synchronization, best torch angle, setting of the correct parameters (voltage and wire speed). It was detected that the geometric parameters such as width and high of the seam weld were not uniform along the paths, though the trajectory was correctly followed.

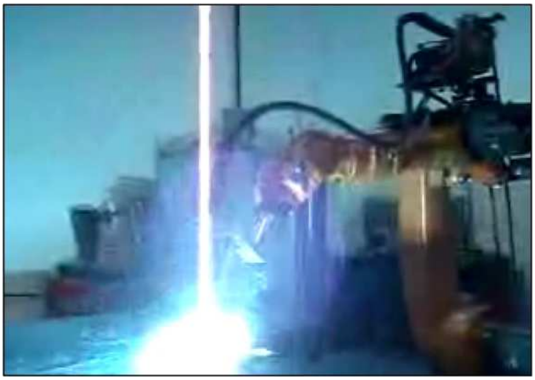


Fig. 8. Robot welding

On going work is looking at improving the actual welding stage by improving the selection of the welding parameters namely, voltage, current and wire feeding speed. At present, the robot is able to work only on flat surfaces but future work has been envisaged to work on 3D surfaces using depth information as well as monitoring the pool weld that would require implementing an effective seam tracking mechanism including a robust vision system to be used on-line during welding tasks.

Acknowledgements

The authors wish to thank the following organizations who made possible this research: The Consejo Nacional de Ciencia y Tecnologia (CONACyT) through Project Research Grant No. 61373, and for sponsoring Mr. Davila-Rios during his doctoral studies and to the Corporacion Mexicana de Investigacion en Materiales for its support through the Research Grant Project No. GDH - IE - 2007.

6. References

Davila-Rios, I.; Lopez-Juarez, I.; Martinez-Martinez, L; and Torres-Treviño, L.M.(2009) Implementation of an Intelligent Robotized GMAW Welding Cell, Part I: Design and Simulation. *In Advances in Robot Manipulators*, ISBN 978-953-7619-X-X. Edited by IN-TECH, Vienna, Austria.

Henning, Michi, Steve Vinoski. (2002) "Programación Avanzada en CORBA con C++",*Addison Wesley*, ISBN 84-7829-048-6.

Holliday, D B, Gas-metal arc welding, (2005) ASM Handbook, Vol 6, *Welding, Brazing and Soldering*, 2005 pp (180-185).

Jia, Songmin; Hada, Yoshiro; Ye, Gang; Takase, Kunikatsu (2002) "Distributed Telecare Robotic Systems Using CORBA as a Communication Architecture" *International Conference on Robotics & Automation Washington, DC.* May 2002.

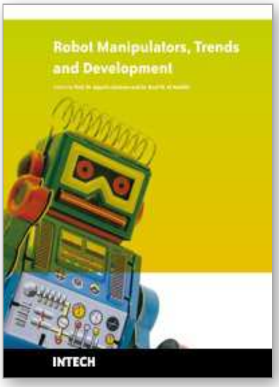
I. Lopez-Juarez, R Rios Cabrera. (2006) Distributed Architecture for Intelligent Robotic Assembly, Part I: Design and Multimodal Learning. In Manufacturing the Future: Concepts, Technologies & Visions. Edited by Vedran Kordic, Aleksandar Lazinica, Munir Medran. *Advanced Robotics Systems International.* Pro Literatur Verlag, Mammendorf, Germany. Pp. 337-366.

I. Lopez-Juarez, M. Peña-Cabrera*, A.V. Reyes-Acosta (2009). Using Object's Contour and Form to Embed Recognition Capability into Industrial Robots. *In Advances in Robot Manipulators*, ISBN 978-953-7619-X-X. Edited by IN-TECH, Vienna, Austria.

Peña-Cabrera, M & Lopez-Juarez, I. (2006). Distributed Architecture for Intelligent Robotic Assembly, Part III: Design of Invariant Recognition Vision Systems. In Manufacturing the Future: Concepts, Technologies & Visions. Edited by Vedran Kordic, Aleksandar Lazinica, Munir Medran. *Advanced Robotics Systems International.* Pro Literatur Verlag, Mammendorf, Germany. 2006. Pp. 400-436.

Pires, J.N., Godinho, T. Nilsson, K., Haage M., Meyer, C.(2007). Programming industrial robots using advanced input-output devices: test-case example using a CAD package and a digital pen based on the Anoto technology. *International Journal of Online Engineering (iJOE),* Vol 3, No 3.

Veiga, G., Pires, J.N., Nilsson, K., (2007). On the use of SOA platforms for industrial robotic cells. In: Proceedings of, *Proceedings of Intelligent Manufacturing Systems,* IMS2007, Spain, 2007.



INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821