Implementation of an Intelligent Robotized GMAW Welding Cell, Part 1: Design and Simulation

I. Davila-Rios, I. Lopez-Juarez[1], Luis Martinez-Martinez
and L. M. Torres-Treviño[2]
Corporacion Mexicana de Investigacion en Materiales (COMIMSA)
[1] Centro de Investigacion y de Estudios Avanzados del IPN (CINVESTAV)
[2] Universidad Autonoma de Nuevo Leon (UANL)
Mexico

1. Introduction

Today, welding tasks can be accomplished in two ways: 1) Manual or conventional, which is made by humans with a welding gun; and 2) Automated welding that is carried out by robots, this robots are called "Welding Robots". Current welding 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. Even 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.

The design proposal considers first to simulate the whole welding process considering issues like floor plant space, robot configuration, trajectory planning, welding equipment and supplies, etc. and secondly, the utilization of novel teaching tools for welding trajectories. The contribution has been divided in two chapters as follows: In Part I, the robotic cell set up (including off-line and on-line programming) using current 3D software simulation, equipment commissioning and testing, distributed workcell communication and the voice-command program design are presented; while in Part II, the description of an error recovery strategy in conjunction with a novel teaching trajectory using machine vision is presented.

In this chapter, we describe the design to integrate a robotic welding cell within a 3D simulation environment. The design was used as a reference for the actual construction of the robotized GMAW welding cell. We report a methodology to set up a welding robot cell from scratch using a 3D software robot simulation from Delmia named Robotics V5®. The simulation included welding accessories that were designed first in CAD software and imported by the Delmia® software. These accessories were gas tanks, wire coils, Lincoln 455

Power wave station, 10R automated Lincoln wire feeder, Magnum 400 robotic torch, welding part conveyor and an industrial KUKA KR16 industrial robot.

The methodology included layout definition, welding part design, robot and welding station commissioning, off-line programming including collision avoidance reports during simulations. The whole setting was completed by simulation and different layout schemes were tested. The design also considered a voice-command driven environment, so that the robot cell could be commanded via voice. The voice-command software was developed in C++ using the Microsoft Speech Application Interface (SAPI V5.0). This interface was also simulated off-line using two computers first with wireless communications and later, including the robot.

The organisation of the chapter is as follows. In section 2, the GMAW welding process is reviewed. In section 3, different aspects of the robot welding system are considered including the design stages of the welding workcell from simulation to implementation. Section 4, describes the robot simulation as well as an assessment of collision failures. Section 5 presents a robot program example using off-line programming. In section 6, the design of the distributed robotic workcell is explained as well as some issues regarding distributed systems using CORBA. The voice-command application and grammar description are presented in section 7. Finally, section 8 provides the conclusions and further work.

2. Gas Metal Arc Welding Process

In Gas Metal Arc Welding (GMAW) process an electric arc is established between a consumable electrode fed continuously to the weld pool and the work-piece. 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 and 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 Welding equipment

Basic equipment for conventional GMAW consist of the power source, the electrode feed unit, the welding torch and the shielding gas regulator, as represented schematically in Figure 1.

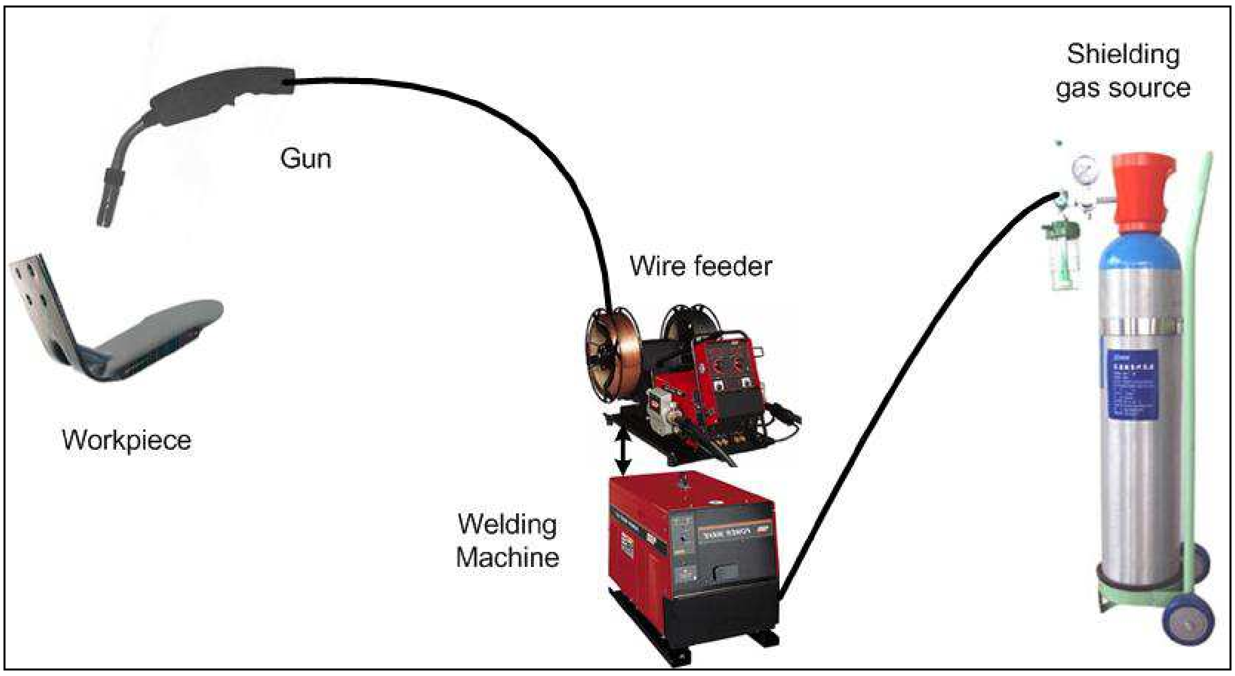


Fig. 1. Schematic representation of gas metal arc welding process (GMAW)

2.1.1 Power Source

Most common GMAW power sources are of the inverter type providing a constant-voltage output. A constant-voltage power source used in conjunction with a constant speed wire feeder can provide self-adjustment and stabilization of the arc length, in order to compensate for the variation in the torch to work-piece distance that occur mainly during manual welding operation. In addition these machines provide slope control of the power source characteristics and of the inductance in order to control spatter in short-circuiting transfer (Norrish, J. 1992).

2.1.2 Electrode Feed Unit

The electrode feed unit and the welding control mechanism are generally furnished in one integrate package. The electrode feed unit pulls the electrode from the reel and pushes it through a conduit to the welding torch (gun). This unit is composed of a direct-current motor, that varies the motor speed over a large range, a gear box and two pairs of rolls with a pressure adjusting screw and wire guides, that transmit mechanical energy, straighten and guide the electrode. The welding control mechanism regulates not only the electrode feed speed and the start and stop of the electrode but also the delivery of shielding gas, current and cooling water (when necessary) to the torch. When the torch cable is externally attached to the robot arm it is exposed to work-piece interface and to premature wear. Modern robotic systems can include special arms with internal cabling, in order to prevent interference, increasing cable life.

2.1.3 Welding Torch

Main functions of the welding torch are to furnish the electrode with electrical current and direct the electrode and gas flow to the work-piece. Main components of the welding torch are the contact tube, where the current is transmitted to the electrode, the nozzle, which provides a laminar gas flow to the weld pool, the torch switch, which sends signals to the feed unit, and the handle. The handle supports the gas and water (if necessary) tubes, the electrode guide tube and cables for current and signals. MIG torches for low current and light duty cycle (up to 60%) are gas cooled. Robotic torches are in general water cooled, but if gas cooled torches are used they must be larger than manual torches. Robotic torches usually have emergency-stop capability to prevent damage to the robot arm and the welding torch is in the event of a collision. They are also provided with automatic cleaning, that may include a pressurized air system for blowing spatter out of the nozzle (Hancock & Johnsen, 2004).

2.2 Process Parameters

Welding parameters affect the way the electrode is transferred to the work-piece, the arc stability, spatter generation, weld bead geometry and overall weld quality. The main parameters of the process are current, voltage, travel speed, electrode extension and electrode diameter, though others, such as electrode orientation, electrode composition and shielding gas, also have direct influence on the metal transfer mechanisms. These parameters are not independent. The current and voltage, for example, are correlated by the arc characteristic curves. Voltage depends not only of the arc length but also on the electrode extension and on the shielding gas.

2.2.1 Current

Direct current electrode positive (DCEP) is the most used current in GMAW because it gives stable electric arc, low spatter, good weld bead geometry and the greatest penetration depth. For low currents and voltages in combination with active shielding gases or mixture containing active gases, dip or short-circuiting transfer is obtained. The utilization of relatively low current can give insufficient penetration and excessive weld reinforcement, occasioned by poor wetting action of the weld metal. Globular repelled transfer can be found when electrode negative polarity is used with solid wire, but this mode of transfer has no industrial application due to poor stability and high spatter levels.

2.2.2 Voltage

Arc voltage is directly related to current, as indicated above, and with arc length, increasing with it. Voltage also depends on the shielding gas and electrode extension. The increase of arc voltage widens and flattens the weld bead. Low voltages increase the weld reinforcement and excessively high voltages can cause arc instability, spatter, and porosity and even undercut.

2.2.3 Welding Speed

Increase in the welding speed gives a decrease in the linear heat input to the work-piece and the filler metal deposition rate per unit of length. The initial increase in welding speed can cause some increase in penetration depth, because the arc acts more directly in the parent material, but further increase in speed decreases penetration and can cause undercut, due to insufficient material to fill the cavity produced by the arc.

2.2.4 Electrode Extension

The electrode extension is the electrode length that is out of the contact tube. The increase of electrode extension, produced by the increase of the torch distance to the work-piece for a specific parameters set, increases electrode melting rate because of the Joule effect. Electrode extension ranges from 5 to 15 mm for dip transfer, being higher (up to 25 mm) for the other transfer modes.

2.2.5 Shielding Gas

Shielding gases have an effect on arc stability, metal transfer mode, weld bead shape and melting rate. Gases used in GMAW can be pure gases, binary, ternary and exceptionally quaternary mixtures. Common pure gases are argon, helium and carbon dioxide. The first two are inert gases and are used principally in welding of light alloys, nickel, copper and reactive materials. Helium has a higher ionization potential than argon, providing larger weld pools, but is more expensive. Carbon dioxide is an active gas and is used in welding of carbon steel. It produces high levels of spatter but provides high penetration depth. Binary mixture are commonly argon/carbon dioxide (up to 20% CO2), argon/oxygen (up to 5% O2) and argon/helium (up to 75% He). The first is used in the welding of carbon and low alloy steels, the second of stainless steels and the third of nonferrous materials. The addition of oxygen or carbon dioxide to argon stabilizes the welding arc and changes the bead shape. The objective of adding helium to argon is to increase heat input and consequently welding speed, but also to reduce the incidence of weld porosity (Lyttle, K. A. 2002).

2.2.6 Health and Safety

The major potential hazards of arc welding processes are the high-voltage electricity, which can injure and kill personnel, the fumes and gases, which can be dangerous to health, the electric arc radiation, which can injure eyes and burn skin and the noise that can damage hearing. The exposure to the high open - circuit voltage of power supplies can cause dangerous electric shocks, which can be prevented by connecting all the electrical equipment and work – pieces to a suitable electrical ground. All electric cables should be suited to the maximum current and must remain insulated and dry.

Fumes and gases are generated in all arc welding processes, being particularly intense in the flux cored arc welding process. Metal fumes of nickel, chromium, zinc, lead or cadmium, for example, and gases such as carbon monoxide, ozone and nitrogen oxides formed in the arc are very harmful to the health. Enough ventilation or exhaust at the arc, or both must be used in order to keep fumes and gases from the personnel breathing zone.

3. Robotic Welding: System Issues

Robotic welding research deals with the relevant technical and scientific aspects involved in the task of reproducing the work of the experienced and skilled human welder. Welding was for a long time a task performed only by humans, being a craft that combines skills with art and science. Automating welding is therefore a very difficult and demanding objective, because of the required adaptive behaviour of the automatic system. Therefore, for the design of the automated system, a simulator tool is highly desirable in order to consider all the aspects before the actual construction of the cell. In this section, we present the simulation design using Delmia Robotics V5 and wherever needed the design of the required parts using Solid Works ® is explained.

3.1 Environment in 3D Simulator

The simulation is used for the identification of design problems in an early phase, for an iterative design process from design to the finished product. The use of dynamic 3D simulations constitutes an advanced methodology for the analysis of the risks associated to the installation of the systems and solutions in robotics. The advantages of these techniques become evident when facilitating, among other things, the determination of impact risks (collisions) of the robot's part in the tasks during assembly, ergonomics of the position, demarcation of the environment of the robot's work, etc.

Delmia Robotics V5 allows to be carried out simulations of automated manufacturing processes, specifically with robots; this software has a great variety of robots models, as well as of different brands. Delmia Robotics V5 offers an easy and flexible use of solutions for the definition of tools and simulation cells. It provides all the necessary tools to define and to analyze the behaviours and the necessary resources to apply the process plan. The software helps to describe exactly how parts are loaded and unloaded, fixtures, and selection of welding tools. The analysis and validation of the process can be carried out easily and accurately through simulations based on the resources available and have the option to incorporate additional tooling or fixtures (Caie, J. 2008).

The Robotics V5® environment is composed for a great variety of tools called "toolbars", which are presented as icons around the main screen, inside it; they are two very important functions, the compass and the tree PPR (Product, Process and Resource). The compass is used to change position and to move the elements to use, and the tree PPR is used to show the products, resources and how it is carrying out the process inside the simulator, as well as to take a relation among them because they work together. Figure 2 shows the main screenshot when selecting the robot type to be used.

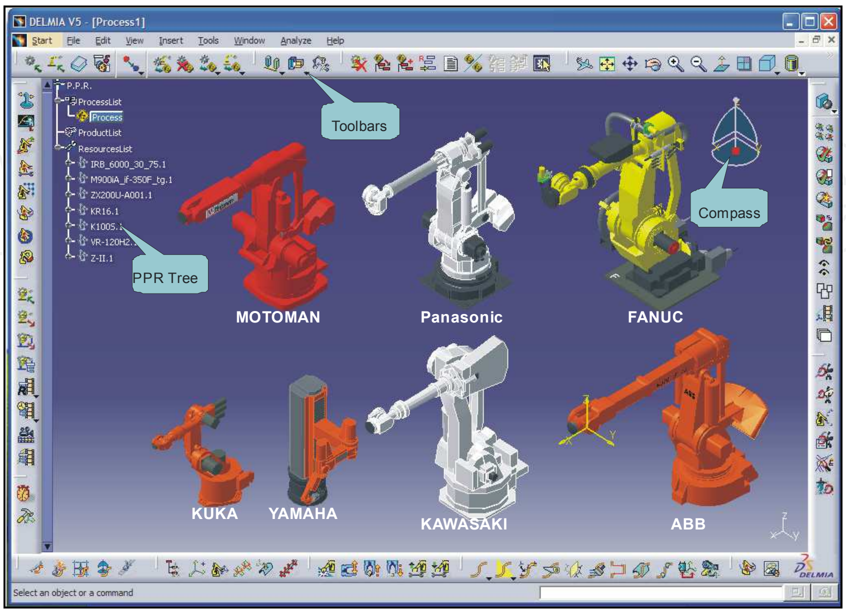


Fig. 2. Main screenshot of the Robotics V5

For the design of specific components, there are many types of software, in our case Solid Works® was used, due to its compatibility with Robotics V5 when exporting the parts.

3.2 Virtual Development of a Robotic Workcell

The first step in our design was to identify the physical location of the welding cell as well as its area dimensions in the shop floor. An important aspect to consider is the floor and environment; the floor requires withstanding the weight of the various devices and machines and the environment have to be dust-free to guarantee appropriate working condition for the electronic devices.

For the workcell simulation, some of the components that are not included in the libraries had to be designed separately. For instance, the welding table, the conveyor belt, the KRC2 robot controller, the physical structure of the cell (walls, windows), etc. These components were designed using Solid Works® and exported to Robotics V5®. A very important issue at this stage is that the design of the components must be carefully made to scale. Finally, but not the least, it is recommended to consider the security of the cell considering emergency exits, placement of fire extinguishers, fume vents and emergency kit location. A screenshot of the cell layout is shown in Figure 3.

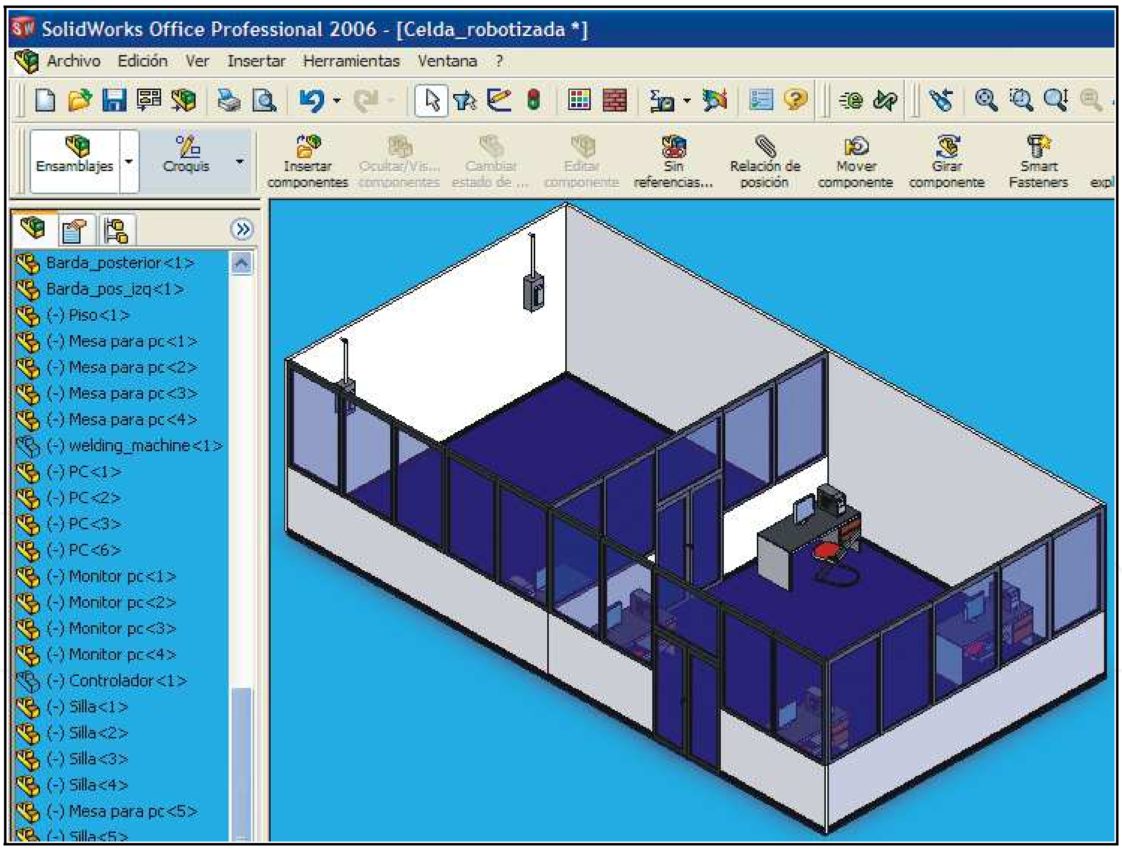


Fig. 3. Workcell in Solid Works® environment

It is also possible to design parts using CATIA® software. This software can be used directly in the simulator Robotics 3D without having to export the designs. They only have to be saved using a CATPart extension. The first step to design the part is to open an application in Delmia called “Mechanical Design”, and then in the option of part design choice open a new project in CatProduct mode. An example of a piece to be welded it is shown in Figure 4.

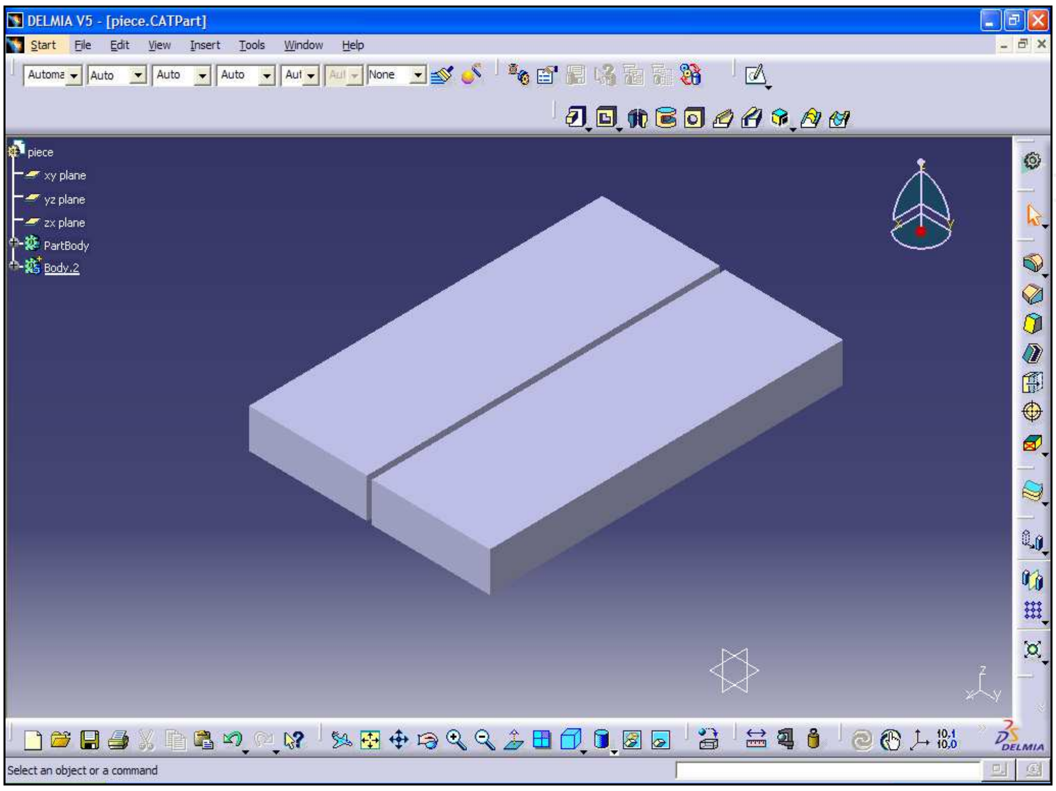
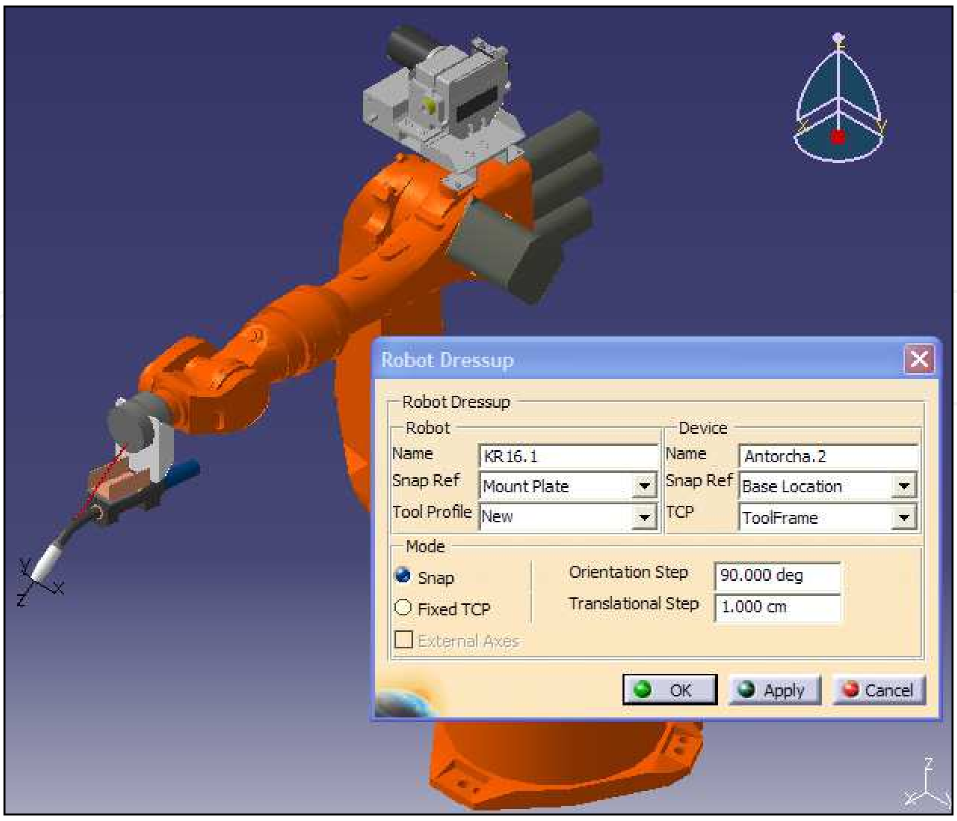


Fig. 4. Piece to be welded in CATIA environment

3.3 Layout Construction in the Robotics V5

Industrial robots generally work with others devices such as conveyor belts, production machines, holding fixtures and tools. The robot and the equipment form a workcell. Also you can use the term work station, but this term is generally used to identify 1) One workcell with only one robot or 2) a work position along of a line production of some station of robots work. Two problems in applications of robots engineering are, the physical design of the work cell and the design of the control system that it will coordinate the activities among the diverse components of the workcell (Groover M. et al, 1995).
For the construction of the layout in the simulation software Robotics V5, is it necessary to open a document in “process“ mode in order to be able to work in the simulation environment and to be able to open the “PPR tree“. It is necessary to program in the beginning mode and simulation resource the option of “Device task definition“. To create the layout in the first screen, it is necessary to insert products and resources. Inserting products and resources will depend on the way they are used in the process. Products such as *Body Side Assembly Part, Robot Kuka KR16, welding table, torch, workcell and a riser* (platform for the Robot) need to be selected from the PPR screen and then attached together. To mount the welding torch on the robot, the toolbar “Robot management>Robot dress up“needs to be used as it is illustrated in Figure 5. DELMIA can import files with .cgr extension from Solid Works (EES 2006).



Robot Dressup

Robot Dressup

Robot

Name: KR16.1

Snap Ref: Mount Plate

Tool Profile: New

Device

Name: Antorcha.2

Snap Ref: Base Location

TCP: ToolFrame

Mode

● Snap

○ Fixed TCP

□ External Axes

Orientation Step: 90.000 deg

Translational Step: 1.000 cm

Fig. 5. Mounted torch and power feed to the Robot Kuka KR16

Finally, after considering all accessories and equipment needed in the welding cell, the final layout was completed and built as shown in Figure 6(a) and 6(b), respectively.
The lay out construction is presented in Figure 6.

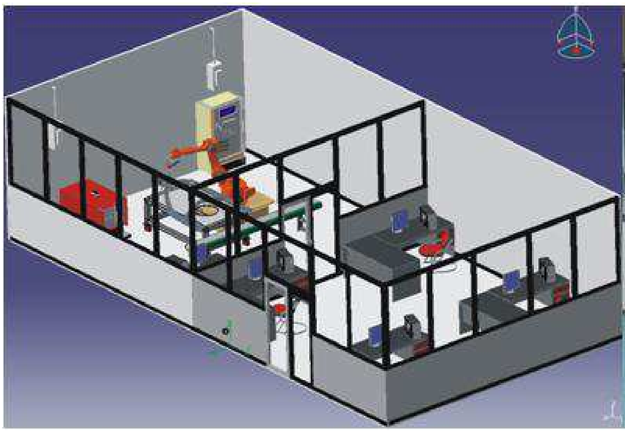


Fig. 6(a). Lay out of the workcell.

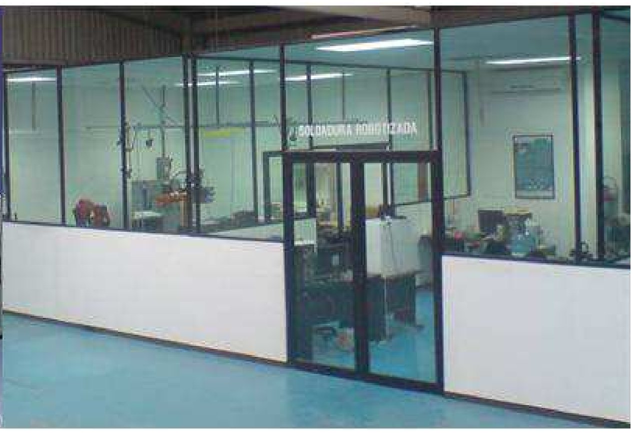


Fig. 6(b). Finished workcell.

4. Robot Welding Simulation

The second part of the research involved the simulation of the robot welding in order to analyse the robot's working envelope to prevent any crash or collision with the environment. This is an important off-line stage since the user can learn how to move the robot and program diverse tasks and later, if needed, the robot program can be translated into another robot's manufacturer language. This is also a useful feature that could be used for learning how to program certain robot.

In the simulator, it is necessary to create *tags* and robot tasks associated to them. *Tags* are those points for which the robot will carry out the welding operations. To complete the programming process, the robot's motion has to be adjusted with the help of the Teach pendant and moving the compass to determine the robot movement. Alternatively, a function inside the Teach Pendant called Jog, that manipulates the robot's movement through each one of the DOF, can be used (EES 2006).

4.1 Simulation Example

First of all, it is important to maintain intact the positions of the components within the workcell at the beginning of each simulation, an initial state has to be selected and a robot type. In our case we worked with the KUKA KR16 Industrial Robot. A snapshot of the simulation is shown in figure 7(a) and in figure 7(b) the actual workcell is shown.

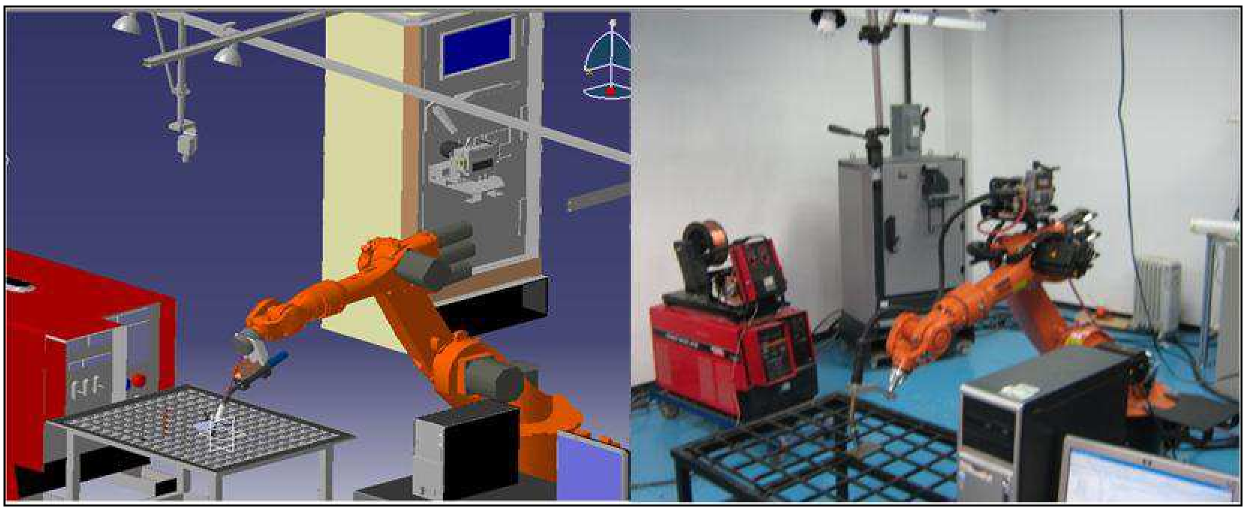
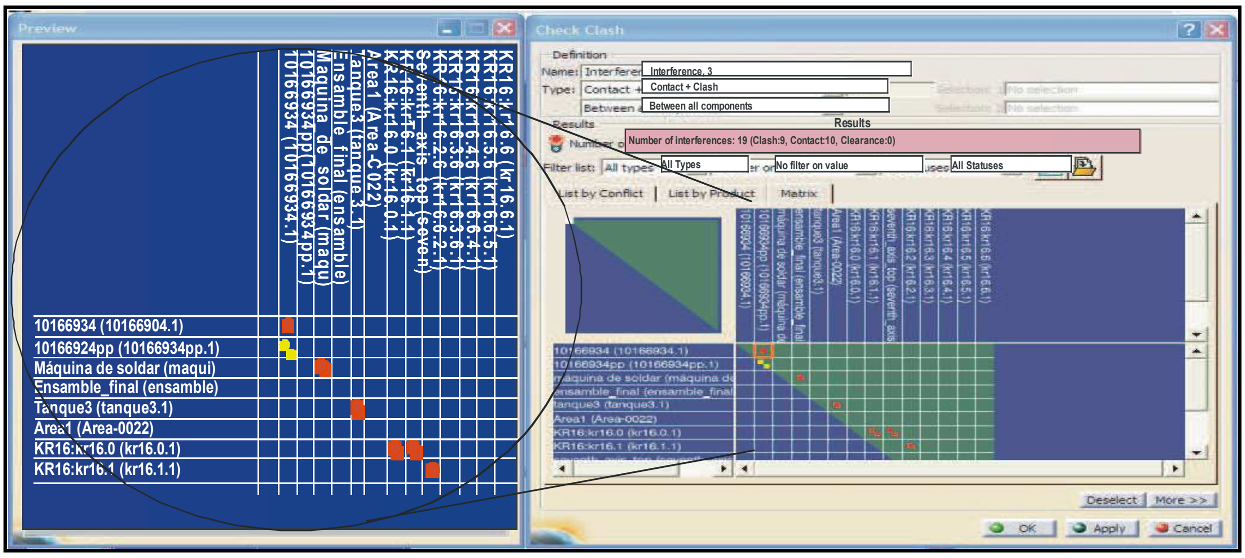


Fig. 7(a). Simulation welding process. Fig. 7(b). Real workcell

Despite the cell distribution and robot location and configuration, it is possible to make contact and collision with other components. The simulator provides a Matrix of contacts and clashes in order to reprogram the trajectories if necessary. Figure 8 shows the results of the clashes with high relevance due to detected collisions between the torch and the work table and also small contacts between robot articulations.

In Figure 8, it is also shown on the left hand side a graph of the area affected by the collision between the robot and the torch with one product named "piece to be welded". On the right hand side, it shows the relation matrix of collisions and contacts, among all the devices. In this example, the simulation resulted in a total of 19 interferences, from which 9 were collisions and 10 contacts. In these results, the collisions are important since they can damage some devices and even the robot.



Name: Interference. 3

Type: Contact + Clash

Between all components

Number of interferences: 19 (Clash:9, Contact:10, Clearance:0)

Filter list: All Types | No filter on value | All Statuses

Fig. 8. Matrix simulation results

A human welder must be able to interpret all the results of this simulator. The examination of the Standard CRAW-OT (Certified Robotic Arc Welding, Operator and Technician) by the AWS is designed to test the knowledge of welding fundamentals and robotic welding (EES 2006). Table 1 presents the CRAW-OT subjects that can be evaluated in the simulator in comparison with a real workcell.

Subject	Real Workcell	Virtual Simulator
Weld Equipment Setup	✓	
Welding Processes	✓	✓
Weld Examination	✓	
Definitions and Terminology	✓	✓
Symbols	✓	
Safety	✓	✓
Destructive Testing	✓	
Conversion and Calculation	✓	✓
Robot Programming	✓	✓
Welding Procedures	✓	✓
Programming Logic	✓	✓
Kinematics Concepts	✓	✓
Robot Arc Weld Cell Components	✓	✓

Table 1. Comparison between Real Workcell and Virtual Simulator

5. Robot Program Using Off-Line Programming

One of the most important tasks is programming the robot, which is considered very difficult to evaluate because each brand has its own robot programming code. This programming task is easier to evaluate since the simulator provides a conversion utility between different commercial robots. A code example produced by the simulator is provided in Table 2.

Robot KUKA	Robot FANUC	Robot RAPID

&ACCESS RVP
&REL 1
&PARAM EDITMASK = *
DEF Welding_Task( )
;FOLD PTP ViaPoint2
Vel= 50 %
PDAT1 Tool[1]:cell2-
torch.1.
ToolFrame
Base[0];%{PE}
%R
4.1.5,%MKUKATPBASIS,
%CMOVE,%VPTP,%P 1:PTP,
2:ViaPoint2, 3:, 5:50,
7:PDAT1
$BWDSTART=FALSE
PDAT_ACT=PPDAT1
BAS(#PTP_DAT)
;ENDFOLD
END
	/PROG Welding_Task
	/ATTR
	OWNER = MNEDITOR;
	PROG_SIZE = 0;
	FILE_NAME = ;
	VERSION = 0;
	LINE_COUNT = 0;
	MEMORY_SIZE = 0;
	PROTECT = READ_WRITE;
	TCD: STACK_SIZE = 0,
	TASK_PRIORITY = 50,
	TIME_SLICE = 0,
	BUSY_LAMP_OFF = 0,
	ABORT_REQUEST = 0,
	PAUSE_REQUEST = 0;
	DEFAULT_GROUP = 1,1,1,*,*;
	/POS
	P[1]{
	GP1: UF : 1, UT : 2, CONFIG:
	'S 2 , 0, 0, 0',
	X = 1157.208 mm, Y =
	212.886 mm, Z = 1002.855
	mm,
	W = 136.239 deg, P =
	11.951 deg, R = 103.725
	deg };
		%%%
		VERSION:1
		LANGUAGE:ENGLISH
		%%%
		MODULE Welding_Task_mod
		PERS robtarget ViaPoint2:=
		[[1157.208,212.886,1002.8551,
		0,0,0
		PERS robtarget
		Tag1:=[[938.236,285.366,460.4
		29],
		[0.111893,0.51089,0.851944,0.
		025754]
		,[,0,0,0],[9E+09,9E+09,9E+09,
		9E+09,9E+09,9E+09]];
		PERS robtarget
		PROC Welding_Task()
		MoveJ
		ViaPoint2,Default,Default,cel
		l2-torch_1_ToolFrame;
		MoveJ
		Tag1,Default,Default,cell2-
		torch_1_ToolFrame;
		ENDPROC
		ENDMODULE

Table 2. Program code for three different robots

The simulation process produces a robot program like the one shown in Table 2. This program can be loaded to the specific robot controller to perform a real welding cycle using the specified robot. This program is ready to start a welding task which can be programmed either by using a predefined routine (off-line) or via a teaching device (on-line) (Lopez-Juarez, I et al. 2009). In both cases a friendly user interface via voice has been designed in order to facilitate robot programming which is described in section 7.

6. 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 9.

Two computers are used, the Master Controller and the Speech Recognition. The Master Controller is 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.

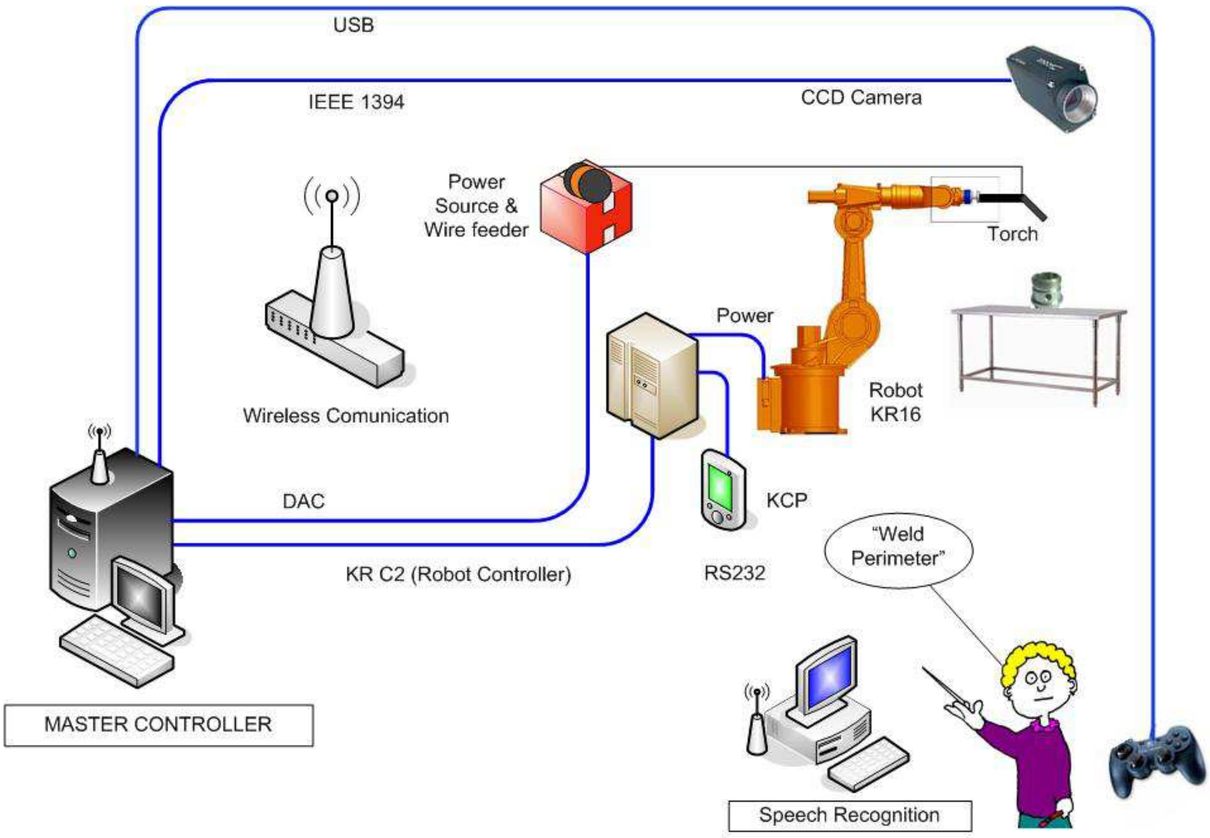


Fig. 9. Robotized Welding System

6.1 Distributed Robotic System

The concept of distributed systems and other technologies recently have made possible the creation of new application called "Networked Robot Systems". The main idea is to solve the heterogeneity problem found in robotic systems due to the multiple component vendors and computational platforms.

The development of robot systems based on distributed components is well supported by different researchers. In (Amoretti et al., 2003), Michael Amoretti et al., present an analysis of three techniques for sensor data distribution through the network. In (Amoretti, 2004) it is proposed a robotic system using CORBA as communication architecture and it is determined several new classes of telerobotic applications, such as virtual laboratories, remote maintenance, etc. which leads to the distributed computation and the increase of new developments like teleoperation of robots. In (Bottazzi et al., 2002), it is described a software development of a distributed robotic system, using CORBA as middleware. The system permits the development of Client-Server application with multi thread supporting concurrent actions. The system is implemented in a laboratory using a manipulator robot and two cameras, commanded by several users. In (Dalton et al., 2002), several middleware are analyzed, CORBA, RMI (Remote Method Invocation) and MOM (Message Oriented Middleware). But they created their own protocol based on MOM for controlling a robot using Internet. In (Lopez-Juarez & Rios-Cabrera, 2006) a CORBA-based architecture for robotic assembly using Artificial Neural Networks was introduced.

In the current investigation, though the system only includes two computers using the same OS, the master controller and the speech recognition. It is important in this early stage to consider the overall layout considering that additional components are being included in the network.

6.1.1 CORBA specification and terminology

The CORBA specification (Henning, 2002), (OMG, 2000) 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.
It is necessary to understand the CORBA terminology, which is listed below:

- A *CORBA object* is a virtual entity, found by an ORB (Object Request Broker, which is an ID string for each server) and it accepts petitions from the clients.
- A *destine object* in the context of a CORBA petition, it is the CORBA object to which the petition is made.
- A *client* is an entity which makes a petition to a CORBA object.
- A *server* is an application in which one or more CORBA objects run.
- A *petition* is an operation invocation to a CORBA object, made by a client.
- An *object reference* is a program used for identification, localization and direction assignment of a CORBA object.
- A *server* is an entity of the programming language that implements one or more CORBA objects.

The petitions are showed in the figure 10: it is created by the client, goes through the ORB and arrives to the server application.

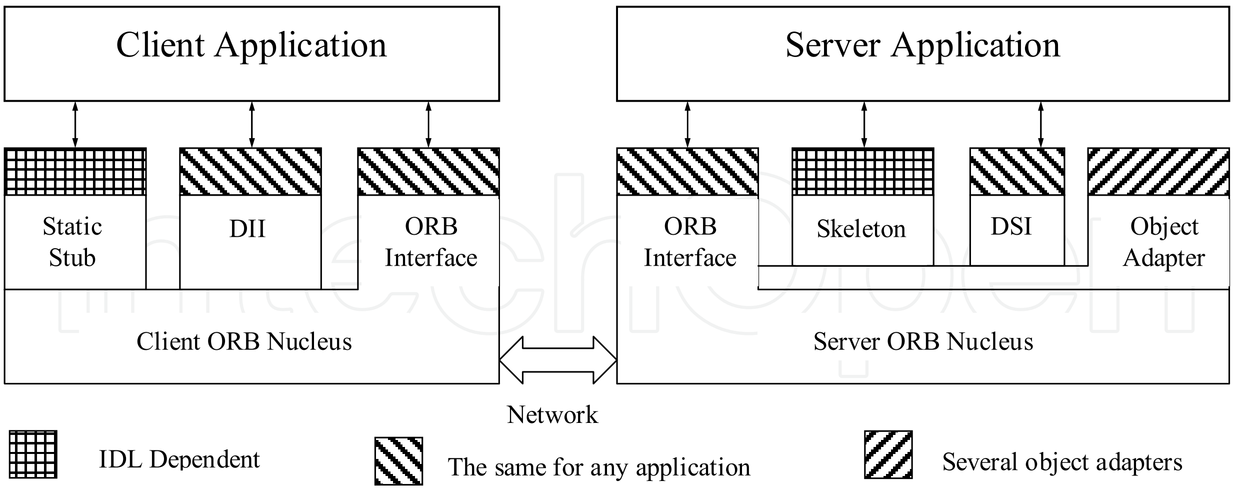


Fig. 10. Common Object Request Broker Architecture (CORBA)

The client makes the petitions using static stub or using DII (Dynamic Invocation Interface). In any case the client sends its petitions to the ORB nucleus linked with its processes. The ORB of the client transmits its petitions to the ORB linked with a server application. The ORB of the server redirect the petition to the object adapter just created, to the final object.

The object adapter directs its petition to the server which is implemented in the final object. Both the client and the server can use static skeletons or the DSI (Dynamic Skeleton Interface). The server sends the answer to the client application.

In order to make a petition and to get an answer, it is necessary to have the next CORBA components:

Interface Definition Language (IDL): It defines the interfaces among the programs and is independent of the programming language.

Language Mapping: it specifies how to translate the IDL to the different programming languages.

Object Adapter: it is an object that makes transparent calling to other objects.

Protocol Inter-ORB: it is an architecture used for the interoperability among different ORBs.

The characteristics of the petitions invocation are: transparency in localization, transparency of the server, language independence, implementation, architecture, operating system, protocol and transport protocol. (Henning, 2002).

The aim of having a Master Controller, is to generate a high level central task controller which uses its available senses (vision and voice commands) to make decisions, acquiring the data on real-time and distributing the tasks for the welding task operation.

The architecture of the distributed system uses a Client/Server in each module. Figure 11 shows the relationship client-server in the Master Controller and Speech Recognition. With the current configuration, it is possible a relationship from any other future server to any client, since they share the same network. It is only necessary to know the name of the server and obtain the IOR (Interoperable Object Reference). The interfaces or IDL components would need to establish the relations among the modules.

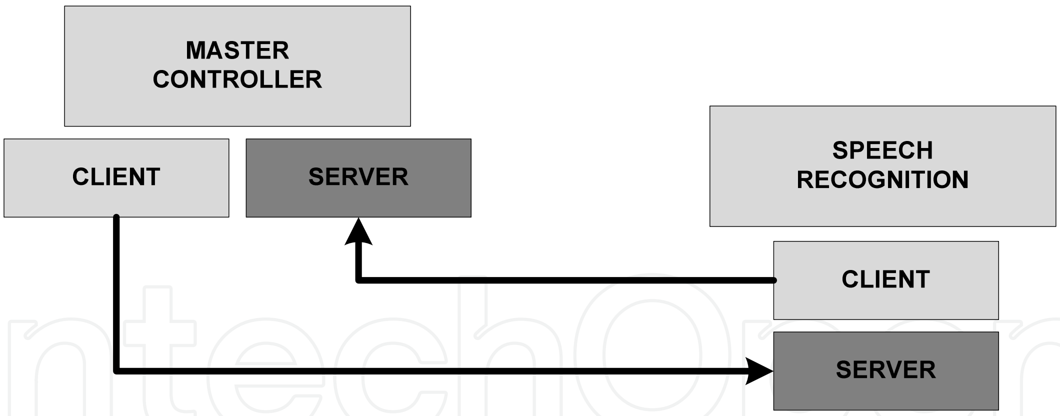


Fig. 11. Client/server architecture of the distributed cell

7. Robot Controller Using Voice-Command Software

The system provides a user interface to receive directions in natural language using natural language processing and Context Free Grammars (CFG). After the instruction is given, a code is generated to execute ordered sentences to the welding system. To effectively communicate the robot controller, it was needed to work in *speech recognition* (*speech-to-text*) as well as in *speech synthesis* to acknowledge the command (*text-to-speech*). Using these features it is possible to instruct the robot via voice-command and to receive an acknowledgement when tasks such as the weld perimeter, weld trajectory, stop, start, go-home, etc. are accomplished.

The Voice Interface was based on Windows XP SP3 operating system using a Speech Recognition PC (§ see section 6). The implementation of the voice command software for the robotic welding system was developed in C++ using the Microsoft Software Development Kit (SDK) and the Speech Application Programming Interface (SAPI) 5.0, which is a programming standard that provides tools and components to speech recognition and speech synthesis.

The SAPI is a high-level interface between the application and the speech engine that implements low-level details to control and to manipulate the real-time operation in several speech engines. There are two basic SAPI engines as it is shown in figure 12. One is the text-to-speech system that synthesis strings and files into spoken audio signals using predefined voices. On the other hand, the speech recognition engine or speech-to text converts the human spoken voice into text strings and readable files. The SAPI is middleware that provides an API and a device driver interface (DDI) for speech engines to implement. The managed System.Speech namespace communicates to these engines both directly and indirectly by calling through the SAPI.DLL. Native

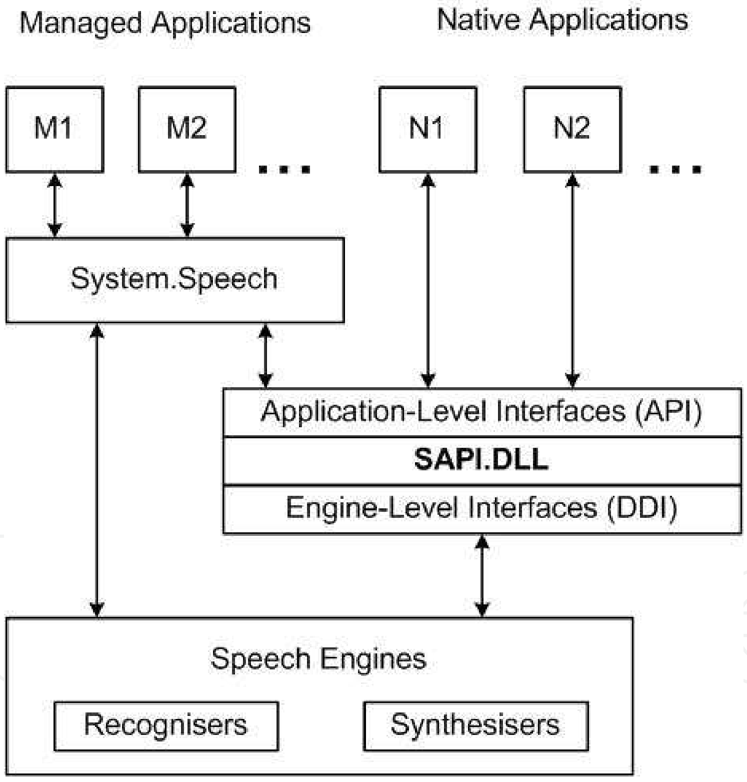


Fig. 13. SAPI engines

There are several category interfaces apart from the speech engines that were used in the application:

Audio
Used to control and customize real-time audio streams compatible with speech synthesis.

Grammar Compiler
Used to dynamically define Context-Free Grammars (CFGs) and compile them into the binary form used by the speech recognition engine.

Lexicon
Provides a uniform way for applications and engines to access the user lexicon, application lexicon, and engine private lexicons. Lexicons provide custom word pronunciations for speech synthesis.

7.1 Grammar

The Context Free Grammar (CFG) format in SAPI 5 defines the structure of grammars and grammar rules using Extensible Markup Language (XML). The CFG/Grammar compiler transforms the XML tags defining the grammar elements into a binary format used by SAPI 5-compliant SR engines. This compiling process can be performed either before or during application run time.

The Speech SDK includes a grammar compiler, which can be used to author text grammars, compile text grammars into the SAPI 5 binary format, and perform basic testing before integration into an application. An example of the developed code is as follows:

```
<GRAMMAR LANGID="409">
<RULE NAME="RobotTask" ID="139" TOPLEVEL="ACTIVE">
<OPT>
<L>
<P>robot</P>
<P>stop</P>
<P>hello</P>
</L>
</OPT>
<OPT>
<L>
<P>weld</P>
<P>go</P>
<P>kuka</P>
</L>
</OPT>
<OPT>
<L>
<P>weld</P>
<P>objects</P>
<P>home</P>
</L>
</OPT>
</RULE>
</GRAMMAR>
<GRAMMAR LANGID="409">
</GRAMMAR>
```

In the program, <GRAMMAR> has a numeric attribute LANGID. We can observe at the beginning of the program, there is also a RULE NAME where "RobotTask" is the grammatical rule; ID, the language identification and TOPLEVEL is declared ACTIVE, but it can be dynamically configured in real-time. The user has to talk only TOPLEVEL rules for the robot to recognise the words. For instance, in the program the words `robot, stop, hello`, can be recognized by the engine. Note that these words are enclosed by <OPT> and </OPT> directives.

Several words were included in the Lexicon being the more important: `weld perimeter, weld trajectory, stop, start, go-home.`

8. Conclusions and Ongoing Work

In this chapter, we described the design and integration of a robotic welding cell using a 3D simulation environment. The design was useful for building the CORBA-based distributed robotized welding cell in this research project. Issues such as layout definition, communication design, welding part design, robot and welding station commissioning were considered. The design also included a voice-command driven environment using the Microsoft Speech Application Interface V5.0. Definition of Context Free Grammars were used so that it was possible to start a typical robot task using a human operator's voice using verbal commands such as "weld perimeter" or "weld trajectory".
The design and simulation previous to the implementation of an automated welding cell is useful, because possible errors can be prevented such as problems of area distribution, security, dimensions, etc. In addition to its great utility to save costs and avoid unnecessary damage to machinery and equipment.

The design of complex robot systems using multisensorial inputs, high-level machine interfaces and distributed networked systems will be elements is of primary importance for advance robot manipulators in the near future so that the work reported in this chapter intents to demonstrate alternative guidelines to design such complex systems.

9. 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 Research Grant Project No. GDH - IE - 2007.

10. References

Amoretti Michele; Stefano Bottazzi; Monica Reggiani & Stefano Caselli. (2003). "Evaluation of Data Distribution Techniques in a CORBA-based Telerobotic System" *Proceedings of the 2003 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2003)*, October, Las Vegas, NV.

Amoretti, Michele, Stefano Bottazzi, Stefano Caselli, Monica Reggiani, (2004), "Telerobotic Systems Design based on Real-Time CORBA", *Journal of Robotic Systems* Volume 22, Issue 4 , PP. 183 – 201.

Barney Dalton, Ken Taylor, (2000). “Distributed Robotics over the Internet”, *IEEE Robotics and Automation.* 7(2): 22-27.

Bottazzi S., S. Caselli M. Reggiani & M. Amoretti, (2002). “A Software Framework based on Real-Time CORBA for Telerobotic Systems”, *Proceedings of the 2002 IEEE/RSJ Int. Conference on Intelligent Robots and Systems,* EPFL, Lausanne, Switzerland, October, 2002.

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

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

I. Lopez-Juarez, R. Rios-Cabrera, & I. Davila-Rios (2009). Implementation of an Intelligent Robotized GMAW Welding Cell, Part 2: Intuitive visual programming tool for trajectory learning”. In *Advances in Robot Manipulators*, ISBN 978-953-7619-X-X. Edited by IN-TECH, Vienna, Austria (In press).

Norrish, J. (1992) Advanced welding processes, *Proceedings of the Institute of Physics Publishing*, 1992.

Hancock, R. & Johnsen, M. (2004) Development in guns and torches, *Welding J* 2004, 83(5), pp29-32.

Lyttle, K A, Shielding gases, *ASM Handbook*, Vol 6, Welding, Brazing and Soldering, pp. 64-69.

Anibal Ollero Baturone, (2001). Robotic, manipulators and Mobile robots. *Alfaomega*.

AWS: American Welding Society (2004). Certified Robotic Arc Welding Operator and Technician. Approved *American National Standard* ANSI.

AWS: American Welding Society (2005). Specification for the Qualification of Robotic Arc Welding Personnel. Approved *American National Standard* ANSI.

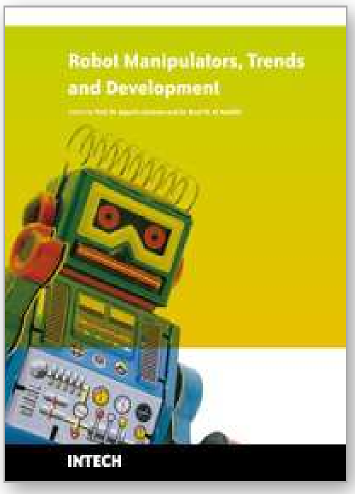
EES: Enterprise Engineering Solutions (2006). V5 Robotics Training Manual. Delmia Education Services Enterprice.

Caie, Jim (2008). Discrete Manufacturers Driving Results with DELMIA V5 Automation Platform. ARC Advisory Group.

Ericsson, Mikael, (2003). “Simulation of robotic TIG-welding“. PhD Thesis, Division of Robotics Department of Mechanical Engineering *Lund Institute of Technology Lund University,* P.O. Box 118, SE-221 00 Lund, Sweden.

Groover Mikell P., Weiss Mitchell, Nagel Roger & Odrey Nicholas. (1995). *Industrial Robotics.* McGraw-Hill, Inc., USA pp. 375-376.

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.



Robot Manipulators Trends and Development

Edited by Agustin Jimenez and Basil M Al Hadithi

ISBN 978-953-307-073-5
Hard cover, 666 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents the most recent research advances in robot manipulators. It offers a complete survey to the kinematic and dynamic modelling, simulation, computer vision, software engineering, optimization and design of control algorithms applied for robotic systems. It is devoted for a large scale of applications, such as manufacturing, manipulation, medicine and automation. Several control methods are included such as optimal, adaptive, robust, force, fuzzy and neural network control strategies. The trajectory planning is discussed in details for point-to-point and path motions control. The results in obtained in this book are expected to be of great interest for researchers, engineers, scientists and students, in engineering studies and industrial sectors related to robot modelling, design, control, and application. The book also details theoretical, mathematical and practical requirements for mathematicians and control engineers. It surveys recent techniques in modelling, computer simulation and implementation of advanced and intelligent controllers.

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