2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications

Overhauling of a ASEA robot IR6 with open architecture

Bomfim, M. H. S.
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
sgtbomfim@yahoo.com.br

Gontijo, R. A.
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
robertoalmeidagontijo@hotmail.com

Bracarense, A. Q.
Departamento de Engenharia Mecânica
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
bracarense@ufmg.br

Lima II, E. J.
Departamento de Engenharia Mecânica
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
ejlima2@gmail.com

***Abstract*— Industrial robot retrofitting usually applies specific solutions for each type and size of robots. This work aims create universal control cabinet to get one interface with computer (PC) and the manipulator (arm, motors and position sensors), following worldwide tendencies of standardization in automation and control systems. In this system, the PC runs the trajectory generation program and sends actuators set-points over a parallel port connection. The cabinet synchronizes the joints´ movements and returns to the PC the actual position of each axis. The retrofitting is to match the new robot technology by leveraging existing systems such as motors, sensors, feedback, power supplies and so on. The initial idea was to reduce costs, but we can not ensure the reliability of these used parts without knowing the historic of the same. For example, a motor used for decades until when will it work? There is not how offer guarantees of a robot, if we do not know the probability of failure of their systems. To solve these problems was introduced the concept of overhauling. In this new design only mechanical part of the old robot would be used, due to the fact that the exchange of electronic and electromechanical components do not greatly increase the cost of adaptation to new technologies.**

Keywords: Retrofitting, Overhauling, Industrial Robot.

I. INTRODUCTION

The Laboratory of Robotics, Welding and Simulation (Laboratório de Robótica, Soldagem e Simulação - LRSS) from Federal University of Minas Gerais (Universidade Federal de Minas Gerais - UFMG) works currently on a project for building small didactic welding robots and industrial robots for especial applications. These robots have big differences in size, number of axes, geometry and power solicitation. These differences motivate the development of a control cabinet with open architecture [1]. The same controller can be connected to any of these robots with few or no modifications and to realize this we had to make a standard platform of motors, controller, connectors, cables and support trails [2].

The cost of the retrofitting of these robots is a critical parameter to make viable this alternative. The retrofitting of an industrial robot usually use a specific solution for each type and size of robot and this generates costs and spends time because each company may have a different project to each robot. The mechanical parts of an old robot must be revised and repaired but in most cases a new project is not necessary.

The robot was completely dismantled and its mechanical condition was evaluated. In the specific case of the ASEA IRB6 robot (35 years), its movements, structure and joints were in perfect condition. Following the principle of retrofitting, it was found that only the power section of the cabinet would be reused [3].

The big problem is that retrofitting has often used parts that do not have a report stating, conditions of use nor its historic maintenance. In the Figure 1 we have a WEG motor with 50,000 hours of use. Looking at a duty cycle low, around 30%, we see that this motor worked 15 years until their coils to enter short [4].

Figure 1. Electric motor winding shorted.
Source: [4].

Robots that are retrofitted usually have more than ten years of use and do not have a historic of use of its parts, making it difficult to carry out technical opinion favorable or unfavorable to the reuse of systems. Therefore we use a new concept which we call overhauling, that will be used only

978-0-7695-4745-9/12 $26.00 © 2012 IEEE
DOI 10.1109/TrustCom.2012.217

mechanical part of the robot, given that the exchange of all or part of the electronic or electromechanical components to a negligible influence on the total project cost.

During the overhauling was decided to develop an open architecture. In this way the accessories, cards and I/O communication, for example, allow the use of high-level languages, making the interchangeable control cabinet with various programming languages.

II. RETROFITTING OF INDUSTRIAL ROBOTS

Industrial robots with anthropomorphic arm began to be manufactured on a large scale in the 70's by the Swedish company ABB and the German company KUKA. Thus it can be seen that the robot oldest produced by these large factories would have now 40 years old. The technological advancement in 40 years makes many robots produced in the 70's and 80's become obsolete today. Generally, when a robotic manipulator complete around 20 years old and needs repair parts or needs to recast its programming based in new technologies. It is scrapped because these modifications generate costs and companies do not meet technical expertise in the market.

To resolve this problem, in the 90's started a research line aimed at older robots adapt to new technologies in a process known as retrofitting. In this new project, the mechanics parts of the robot are reused, usually in good condition, along with electronic and electromechanical components that were in good condition.

A retrofitting work was executed in 2010 by the Universidade Federal do Rio de Janeiro (UFRJ), in this work developed a process of technological modernization of an industrial robot IRB 2000 model [5]. The main objective of the adaptation was the development of hardware and software for control (Fig. 2), configuration and reporting of drivers with a PC running Simulink/Matlab. The Department of Cybernetics at the University of Reading, U.K., also retrofitted an industrial robot by replacing its controller, servo boards and interface cards [6]. As in [5], they reused the power amplifiers, sensors and motors and implemented the new control structure using Simulink/Matlab. In both works, retrofitting proved to be a viable alternative for laboratories and research centers, being possible to reuse the mechanical structure, sensors and actuators of the robot to the development of fully open control architecture.

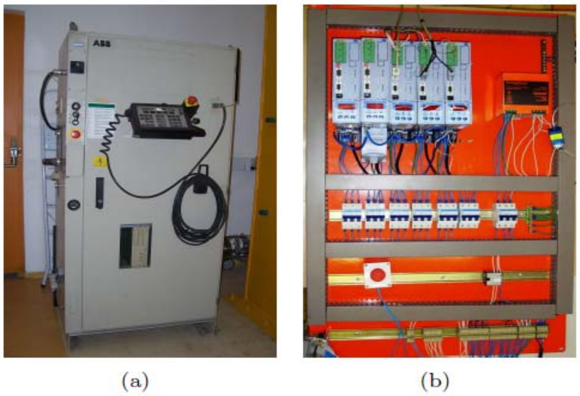

(a) (b)

Figure 2. Original cabinet control (a) and (b) new control panel developed by UFRJ.

III. RETROFITTING OF ASEA ROBOT IRB6

At LRSS has been proposed the implementation of retrofitting the robot ASEA IRB6 manufactured at 1977 (Fig. 3). The implementation of the retrofitting was guided by the flow diagram in Fig. 4.


Figure 3. ASEA ROBOT IRB6, year of manufacture 1977.
Source: [7].

LRSS - UFMG

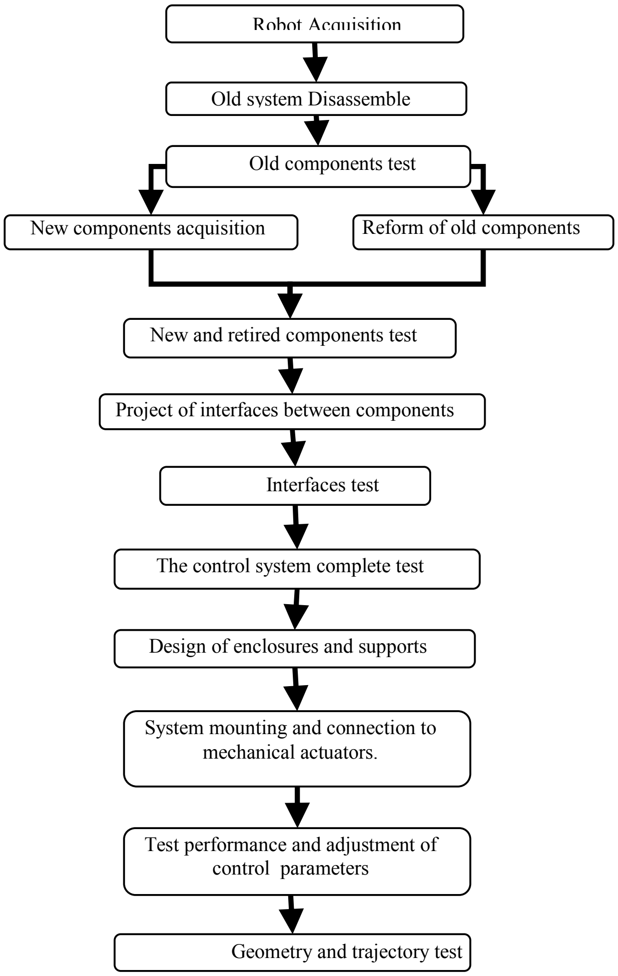


Figure 4. Diagram of possible methodology used for robot retrofitting. Source: [2].

Following the diagram flowchart the third step was the testing of old components. The first component tested was the power supply of the original cabinet. In the test only the power diodes were in good condition. Consequently a new source was mounted just reusing the diodes (Fig. 5).

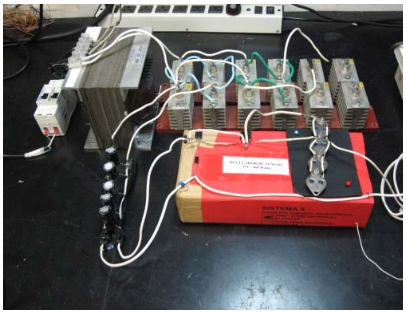


Figure 5. Power supply 35V/1080W. Source: [7].

After design of the power supply, tests were performed on the motors. The tests were performed in open loop connecting with the source connected to the motor. At first the result was satisfactory, because the motors responded to commands to turn on and off of the interrupter, Fig. 5, which powers the source. The next step was the construction of the cabinet control, taking of the old system diodes, motors and feedback sensors (encoders).

A. Development of an universal robot controller (open architeture)

The first step to develop the robots controller is the choice of the architecture to be used (Fig. 6), or which could be used, to list which component are essentials to this architecture. Starting with a basic architecture of discrete movement control, a software generates the trajectory inside a PC and it is sending commands for a pulses generator that will control the drivers.

Inside MS Windows platform, is possible to install the drivers software and APIs for connection with an external pulse generator. This option simplifies the development of user graphical applications due the many tools already developed for this, diminishes the training and software development cost, as the drivers and APIs are developed by the manufacturers. In disadvantage we have a less steady operational system that does not support real time applications. For the usual robot control, this disadvantage is solved using a command input buffer in the pulse generator. So, even if the time between commands varies inside of a certain limit, the pulse generator isolates the motor drivers of this type of variation, but an advanced control system that requires working in real time is not needed.

Another possibility is to use a dedicated port of the PC (usually parallel port, LPT1) as the pulse generator and an operating system capable of running applications in real time (RTAI). Some Linux distributions have been specially developed for these purposes (Linux CNC), so the operating system serves as a trajectory planner and pulse generator. As advantages we have an open system which supports real time applications. However, the cost of developing and maintaining software and hardware grows and is necessary to train not familiar users with the operating system.

In this work we chose to use the first mentioned architecture (external pulse generator and MS Windows operating system).

Another feature that should be considered is whether the trajectory planner works with open or closed loop. Depending of the precision required by the application in which the robot will be used, the closed loop is essential, but if the application does not require extreme precision, the open loop simplifies the interfaces and reduces costs. In our development we use a virtually closed loop, because the pulse generator does not receive the return signal from the position sensor: only the motor driver receives this information. The pulse generator records how many pulses were sent to each driver and creates a virtual robot position. However we let available for interfaces the feedback motion for further expansions.

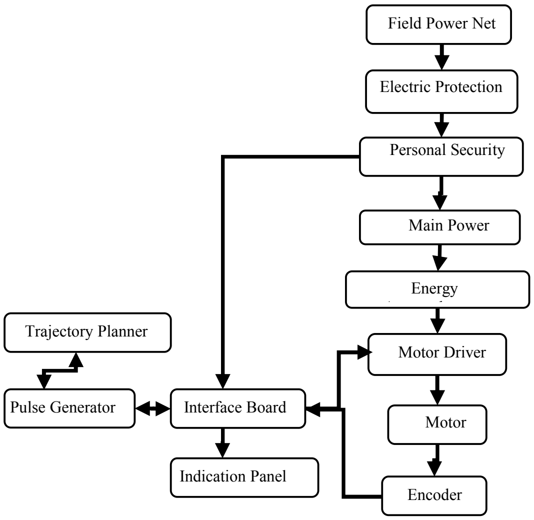


Figure 6. Control System architecture of projected cabinet.

Source: [2].

Aiming to create an open architecture the universal cabinet, Fig. 7, was designed for the retrofitting of the robot. This cabinet consists basically of drivers, power supply and parallel interface connection between the drivers and the computer (to avoid burning of the parallel port). Through the parallel interface can connect any generator of trajectory, meeting the specifications drives. These specifications are basically the operating frequency and type the command signal can be pulse width modulation or displacement proportional to the number of input pulses. For cabinet in question could replace the trajectory generator (Matlab + Mach3) for any other high level language. As an example we mention the C++, Java, Pascal and others. Thus the cabinet meets the characteristics of an open architecture [3], making repairs and programming changes are not tied to specific manufactures.

Figure 7. Cabinet control with open architecture.

B. *New components specifications*

The choice of new components should be done with caution to avoid unnecessary costs and difficulties of integration between components. The selection of the pulse generator should be mainly based on the availability of drivers and APIs for the operating system being used, the parameters of maximum frequency of pulse generation and the number of axes to be controlled simultaneously. So it's more easy to integrate the trajectory generator software with the pulse generator and prevent an unexpected low motion speed.

In our application we chose as the interface between the computer and drivers a break out board (Fig. 11), capable to control 6 axes and to implement one emergency circuit in the board, and there is no need for firmware updates (parallel port).

The switched driver have to answer a wide range of voltage / power of input / output, so it can be used with different sources without the need of change. The switching frequency should be compatible with the pulse generator to avoid the loss of pulses. The possibility of an emergency stop input and an external current limit setting is necessary to increase the security of the system. The PID controller driver should have adjustable parameters to be optimized for various types of motors.

The choice for the cabinet developed drive was G320 Gecko Drive (Fig. 8) with a large power range (18-80 VDC) and current (20 amps max.). This drive has to meet a wide range of engines with power ratings up to 1600 Watts [8]. It uses TTL quadrature feedback, the maximum switching frequency is 25KHz, has an emergency stop and maximum external current limiter and adjustable PID parameters. For overhauling motors in question are used for 60 volts and 6 amperes and encoders 2000 pulses (8000 pulses per square), hence the Gecko Drive G 320 serves very well. Figure 9 shows how the computer interfaces are connected to the pulse generator controllers G320.

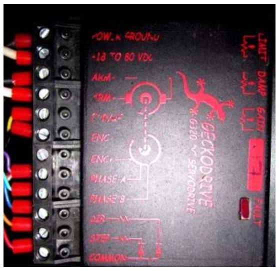


Figure 8. G320 Geckodrive Motor Driver.

Source: [8].

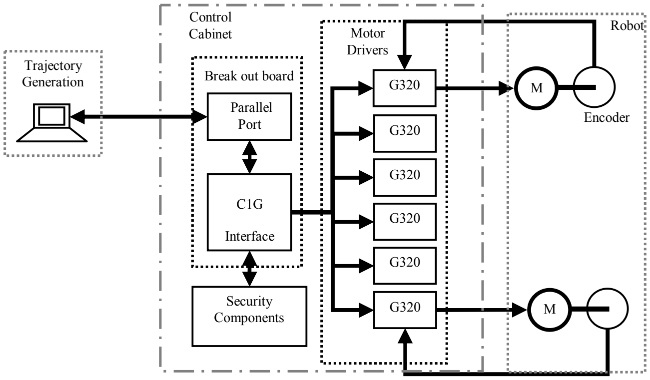


Figure 9. Components interaction diagram.

Source: [2].

The standards for electrical systems suffered major changes in recent years. An electric protection system (Fig. 10) should be specified, containing the current limiters (circuit breakers), voltage surge and residual current device to ensure protection for the components and for the people who use the equipment (NR-10). The components specifications of collective and personal protection system should always take the premise of safe failure, the failure statistics of components should be considered in order to always ensure safety [9].

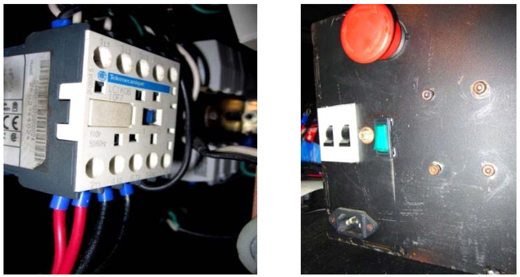


Figure 10. Emergency Stop Circuit and Circuit breaker and emergency stop. switch.

The specification of the power source is a critical cost point and some characteristics of the system should be observed as differences between the original robot installations and where it will be installed, whether the motor changes, if there was an increase of the power system (installation of new components). The average cost per watt is U$0.15 for a new power source. Some characteristics of the components cost should be taken into account. For transformers, a greater transformation ratio and a higher output current means a higher cost, because it uses more material, hence for the same power a lower output voltage means a higher cost. Furthermore the cost of capacitive filter increases with elevation of voltage, while the cost of diode rectifiers increases with increasing current. A good practice is starting from these components and to specify the other components accordingly.

One method to increase the instantaneous power of a source is based on motor use demand; the critic case would be simultaneous start of motors as the start current of a motor is 6x higher than its rated current. Thus we can add a capacitive energy accumulator to support sporadic current peaks. Another interesting point is the amount of power phases used in the primary source. A greater number of phases and rectifier bridge quality makes lower the ripple noise of the source and a lower capacitive filter is needed.

C. *Interfaces and wrappers project*

The design of the interfaces must to meet the specifications of the chosen architecture and modules to be interconnected, and if possible to consider expansions or exchange of other compatible components. This is possible due to the standardization of the main signals as feedback signals of the encoders and step and direction. In the main interface of the developed system controller can switch to other compatible models. To interconnect the trajectory generator (item 3.5) with the actuators was used the breakout board C1G model - Parallel Port Interface (Fig. 11) due to the fact that the parallel port has a reduced cost compared to USB and does not require or perform configuration and firmware updates. This card has been designed to provide a flexible interface for CNC projects using parallel port control software. Proven implementation of buffering and optoisolation circuits provides for a quick and reliable solution. This board is offered for users that do not need optoisolation for the output signals that go to the drivers, because they are using drivers that already have optoisolated inputs, such as the Geckodrive family of products [10].

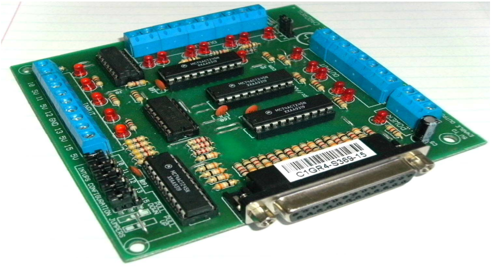


Figure 11. Break out board C1G model.

Source: [10].

The breakout board has Light Emitting Diodes (LED) that refer to step and actuators direction it is good to facilitate the operation and verification of failure. There are 12 slots which allow the connection of six actuators, allowing the remanufacturing of robots with six degrees of freedom.

The interface between drivers and security systems should also to follow the premise of safe failure, or any loss of signal or irregularity should take the system to the emergency stop state. The system should act in the power source, electrical emergency shutdown, as in the drivers and the pulse generator, electronics emergency stop, so that some accumulated energy does not allow any movement after an emergency. The stop system should be integrated with the physical isolation system of the robot (safety doors etc.) and the interface should be compatible with its standards.

The system box (Fig. 12) should be metallic in order to isolate the environment from possible electromagnetic noise generated by power switching of the drivers. Due to the proximity of the robot system and also the weather to which it is exposed, the casing should be specified according to the degree of protection required in the workplace. To support and storage of the casing, a good practice is use the racks found in the market due to the versatility of mounting and expansion, good finishing, good heat dissipation, and low cost since it is not necessary to design and manufacture a support for the system, and are found on a great range of models, sizes and accessories.

Figure 12. Cabinet front panel.

IV. INICIAL TEST (OVERHAULING – A NEW CONCEPT)

In the initial test of the control system in the laboratory, the power supply was connected to the actuator, Geckodrive G320, along with one motor. A PIC board generated the pulses that modulate the motor movement set point. Figure 13 shows an overview of these initial tests.

Due to advanced age and the incompatibility between the motor and position sensor (encoder) and actuator the results were not satisfactory [7]. The originals motors of the ASEA robot failure to rotate the shaft a few degrees. Of the five tested engines, a motor showed high power consumption and high noise, giving symptoms of short coil, because the current values reach values higher than 30 A, while the nominal value is 6 A.

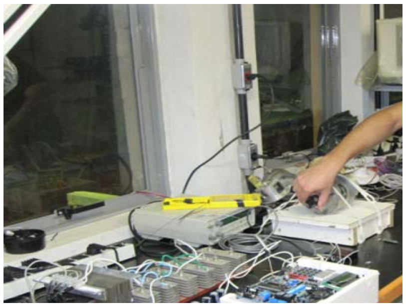


Figure 13. Initial tests - application of torque to the motor shaft.

Source: [7].

Because of the incompatibility of technology and the advanced age of some components, we concluded that it would be more easy the exchange of all electronic and electromechanical components reusing only the mechanics of the robot. With this change in methodology, we left the retrofitting and went into the overhauling.

Fig. 14 shows the old and the new motor with the same power.

(a) (b)

Figure 14. New motor (a) and old one (b).

Source: [7].

Figure 15 shows the bridge rectifiers originally used and the new ones, much smaller but with the same power capacity.

Figure 15. Rectifier bridges with the same power capacity.

Source: [7].

In this new design it would be preferred to reuse the robotic manipulator, with its gears and arms and changing the control cabinet, along with motors and position sensors. Given that the cabinet of command and motors add only 14% of the total cost of overhauling the robot [7], it would be quite interesting in situations that the robot already has an advanced age, which makes difficult the suitability of new technologies and the reuse of components that have no historic of operation, as is the case of ASEA which has about 35 years.

The market price of an ABB robot with similar characteristics to the IRB6 is around U$17.500,00 in the Brazilian Market. In implementing the retrofitting (reusing motors and some electronic components such as diodes) would cost U$1.850,00 [7]. This represents 10% of a new robot, but could not give assurance of this process, because we do not have a historic of how these motors ran and as consequence is not possible know when they needed to predict when maintenance or will fail. Given the above there is a new concept that is the overhauling which only the mechanical part is reused. The overhauling of the ASEA robot IRB6 increases U$500,00 in the cost, this raises the cost of the process from 10% to 14%, reducing the need to test old components and increasing the reliability of the process. The Table 1 has comparison between the cost of overhauling and retrofitting cabinet detailing the price of each component.

TABLE 1. Comparison between the cost of overhauling and retrofitting cabinet.

Description	Quantity	Price (U$)	Retrofitting	Overhauling
Transformer 60V/30A	01	200,00	X	X
Diode - 100V/30A	04	100,00		X
Capacitor 63V/2.200 µF	04	200,00		X
Voltage regulator	05	50,00	X	X
Geckodrive - G320	05	500,00	X	X
Paralell port - G1C	01	50,00	X	X
Servomotors - 60V/6A	05	350,00		X
Encoders	05	750,00	X	X
Safety device and cabinet	01	300,00	X	X
Total cost (U$)			1.850,00	2.500,00

Due to low value addition in the process of retrofitting (35% or U$ 650,00), we opted for the creation of Kit-overhauling that only the mechanics of the robot would be exploited, simplifying the process of remanufacturing. This new methodology saves time and costs on testing of old components, and there is an increase of system reliability.

V. TRAJECTORY GENERATION

After the replacement of components we developed the tool tip trajectory control program. Robots, differently of a conventional CNC machine, can not reproduce a linear movement described in cartesian coordinates. This is because robot has rotational joints and their rotations determine the tool tip position. In other words to determine robot´s tool tip position is necessary at first to make a relation between the cartesian space, joints space and actuators space, this is named by inverse kinematic [11,12].

The ASEA robot has five motors. Each motor is connected to one axis and motor´s rotation results in angular movement of each link. Then it is obtained a specific positioning of the joints. The computer program objective is to allow the robot´s operator gives the initial position and the final position of the tool and the type of wanted movement like linear, circular etc. To make the relation between cartesian space and the joints space mathematics equations were used. These equations consider the mode which the joints have to move to get one specific cartesian position, in other words, these equations system solutions give the angle which each robot´s joint must move. Then joint angles were related to the motor´s rotation. This is because the movement of links is caused by the joint´s movement, and the joint´s movement is caused by motor´s rotation, this is an inverse kinematic (Fig. 16).

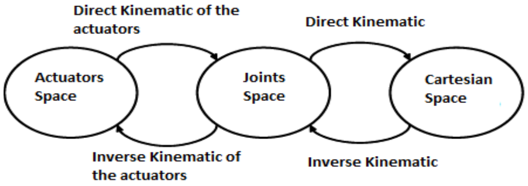


Figure 16. Spaces used to define a position of a robot.

Source: [11].

In this job Matlab and Mach 3 were used (Fig. 17). Matlab was used to generate a sequence of G-codes that allows a determined movement of the robot. The mentioned mathematics equations were implemented in the Matlab and the solutions were reproduced in G-codes built by Matlab [10]. Then G-codes generated for one determined movement are copied and pasted in Mach 3. When Mach 3 starts the program robot reaches the chosen position with the linear movement, circular etc. The robot´s control logic is presented in the blocks diagram above.

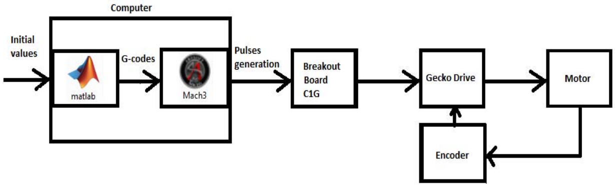
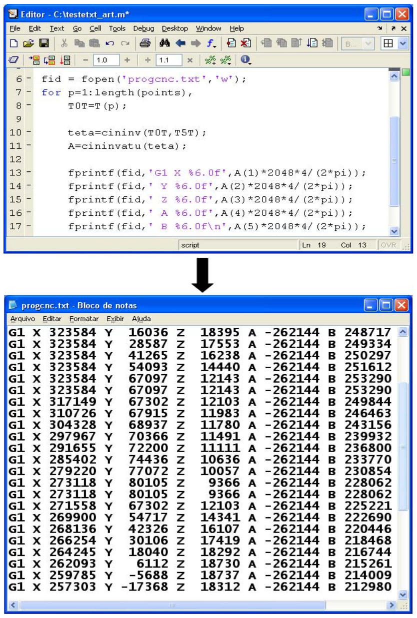


Figure 17. Blocks diagram to one motor.

Matlab was chosen in this work because it is a program that allows calculation of matrices, making it ideal for calculations related to positioning of each link of robot. And Mach 3 was used because it can generate the motor ramp acceleration and can control the motor´s rotation speed. Furthermore Mach 3 has breakout board friendly interface and is very easy to implement a sequence of G-codes in this program.

Figure 18 shows a program developed in Matlab to generate a linear trajectory with 100 mm of length in the direction X. This program calculates the inverse kinematics of actuators and generates the G code sequence in a .txt file to be loaded into Mach3, causing the joint motion of the links and resulting in the final movement of 100 mm linear in the direction X.



```
fid = fopen('progcnc.txt','w');
for p=1:length(points),
    T0T=T(p);

    teta=cininv(T0T,T5T);
    A=cininvatu(teta);

    fprintf(fid,'G1 X %6.0f',A(1)*2048*4/(2*pi));
    fprintf(fid,' Y %6.0f',A(2)*2048*4/(2*pi));
    fprintf(fid,' Z %6.0f',A(3)*2048*4/(2*pi));
    fprintf(fid,' A %6.0f',A(4)*2048*4/(2*pi));
    fprintf(fid,' B %6.0f\n',A(5)*2048*4/(2*pi));
```

```
G1 X 323584 Y  16036 Z  18395 A -262144 B 248717
G1 X 323584 Y  28587 Z  17553 A -262144 B 249334
G1 X 323584 Y  41265 Z  16238 A -262144 B 250297
G1 X 323584 Y  54093 Z  14440 A -262144 B 251612
G1 X 323584 Y  67097 Z  12143 A -262144 B 253290
G1 X 323584 Y  67097 Z  12143 A -262144 B 253290
G1 X 317149 Y  67302 Z  12103 A -262144 B 249844
G1 X 310726 Y  67915 Z  11983 A -262144 B 246463
G1 X 304328 Y  68937 Z  11780 A -262144 B 243156
G1 X 297967 Y  70366 Z  11491 A -262144 B 239932
G1 X 291655 Y  72200 Z  11111 A -262144 B 236800
G1 X 285402 Y  74436 Z  10636 A -262144 B 233770
G1 X 279220 Y  77072 Z  10057 A -262144 B 230854
G1 X 273118 Y  80105 Z   9366 A -262144 B 228062
G1 X 273118 Y  80105 Z   9366 A -262144 B 228062
G1 X 271558 Y  67302 Z  12103 A -262144 B 225221
G1 X 269900 Y  54717 Z  14341 A -262144 B 222690
G1 X 268136 Y  42326 Z  16107 A -262144 B 220446
G1 X 266254 Y  30106 Z  17419 A -262144 B 218468
G1 X 264245 Y  18040 Z  18292 A -262144 B 216744
G1 X 262093 Y   6112 Z  18730 A -262144 B 215261
G1 X 259785 Y  -5688 Z  18737 A -262144 B 214009
G1 X 257303 Y -17368 Z  18312 A -262144 B 212980
```

Figure 18. G-code generated in Matlab with inverse kinematics.

In Figure 20, we have a picture of the ASEA robot IRB6 after the process of overhauling. The Figure 19 is a monitor with the G-code generated in Matlab (left side) running on Mach3. The result of joints´movement is shown in the figure below (Fig. 20). The robot during the test showed good repeatability and acceleration.

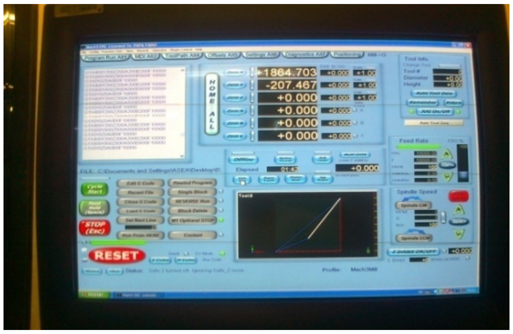


Figure 19. G-code generated in Matlab controlling the movement of the joint through Mach3.

Figure 20. G-code generated in Matlab controlling the movement of the joint through Mach3.

VI. CONCLUSIONS

Using the techniques described in this paper, it can be developed a standard platform for overhauling of old robots and prototyping new robots with a final lower cost than the equivalent equipment already found in the market. However, some drawbacks have to be further studied, like reliability and repeatability of the overhauling system. Several steps have to be reviewed in order to maintain low cost in any configuration of robot to be overhauled, and generic components should be preferred in order to obtain the benefits of scale.

For the particular case of ASEA robot model IRB6, recycling of electronic and electromechanical parts would not be compensatory because of its advanced age, about 35 years. Because of this the motors had several problems like high noise, heat and oscillation. It was also shown the lack of compatibility between the technology of actuators, Geckodrive G 320, encoders and sensors used as feedback of the robot. Other factors that should be taken into consideration when reusing old components would be the power density and power consumption.

Although the project was initially conceived as a retrofitting, due to problems it was decided to do a complete overhauling. In this new design it would be used only the mechanical parts of the old robot, replacing the control cabinet, motors and sensors. This change would result in an increase of 35% of production cost, but would make it more reliable and fast.

Before starting a overhauling is important to conduct a thorough assessment of how are the mechanical parts of the robot. One should check the effects of corrosion, buckling, warping, and damage mainly in the joints. After being certified that all the mechanics are in a good state, the overhauling can be carried out without the worry of testing electromechanical components or adapt the old electric components with new technologies.

REFERENCES

1] Lages, W.F., Henriques, R.V.B., Bracarense, A.Q., 2003. "Arquitetura Aberta para Retrofitting de robôs", Manet Notes Workshop, Bragança Paulista, SP, Brazil.

[2] Bracarense, A.Q., Cordeiro, J.F., Kienitz, K.M., Bomfim, M.H.S., Souza, A.J.A., Filho, F.A.R., Lima II, E.J., 2009. "Development of control cabinet for industrial robots up to six degrees of freedom", 20 th International Congress of Mechanical Engineering, Gramado, Rio Grande do Sul, Brazil.

[3] Lages, W. F., Bracarense, A. Q., 2003. "Robot Retrofitting: A Perspective to Small and Medium Size Entreprises", 3rd Austrian-Brazilian Automation Day, 2003, São Bernardo do Campo, SP, Brazil.

[4] Catálogo WEG, Motores trifásicos, March 25, 2012, http://www.weg.com.br.

[5] Silva, A.L., Orenstein, L.P., Leite, F.L.A.C., Gleiser, G., 2010. "Atualização de Hardware e software de um robô industrial", Congresso Brasileiro de Automática, Bonito, Mato Grosso do Sul, Brazil.

[6] Becerra, V.M.; Cage, C.N.J.; Harwin, W.S.; Sharkey, P.M.; , "Hardware retrofit and computed torque control of a Puma 560 Robot updating an industrial manipulator," *Control Systems, IEEE*, Oct. 2004.

[7] Bomfim, M.H.S., 2009. "Projeto de um gabinete de comando com interface USB para robôs de até seis graus de liberdade", trabalho de conclusão de curso, Belo Horizonte, Minas Gerais, Brazil.

[8] G320-REV-7 Manual, Installation Notes, March 3, 2012, http://www.geckodrive.com/

[9] Bracarense, A.Q., Henriques, R.V.B., Torres, G.C.F., Vasconcelos, D.C.M, 2003. "Retrofitting do robô ASEA IRB6'', 2 Congresso Brasileiro de Engenharia de Fabricação, Uberlândia, Minas Gerais, Brazil.

[10] CNC4, Breakout boards, March 26, 2012, http://www.cnc4pc.com.

[11] Lima, E.J., 2005, "Soldagem Robotizada com Eletrodo Revestido", Programa de Pós-Graduação em Engenharia Mecânica da UFMG, Belo Horizonte, Minas Gerais, Brazil.

[12] Craig, J.J., 1989, "Introduction to Robotics Mechanics and Control", second edition, Addison-Wesley Publishing Company, Inc.