

A LOW COST METHODOLOGY APPLIED TO REMANUFACTURING OF ROBOTIC MANIPULATORS

MARCELO H. S. BOMFIM¹, ALEXANDRE Q. BRACARENSE², FAGNER G. F. COELHO²,
EDUARDO J. LIMA II², ROBERTO A. GONTIJO²

¹ *Brazilian Institute of Robotics (BIR), SENAI CIMATEC, Av. Orlando Gomes, 1845, 41650-010, Salvador, BA, Brasil*

² *Laboratório de Robótica Simulação e Soldagem (LRSS), Departamento de Engenharia Mecânica, Universidade Federal de Minas Gerais, Av. Antonio Carlos, 6627, 31270-901, Belo Horizonte, MG, Brasil*
E-mails: sgtbomfim@yahoo.com.br, bracarense@ufmg.br, fgfcoelho@yahoo.com.br, eduardo@demec.ufmg.br, robertoalmeidagontijo@hotmail.com

Abstract—Nowadays, an increasing number of robotic manipulators are discarded on the factory floor, because they are classified as unserviceable equipment, while still being able to be reused. This problem occurs due to the fact that robot manufacturers use fully closed architecture control to their hardware and software systems. Thus, there is no interchangeability between the control cabinets and the mechanical manipulators, which make the repairs, modifications and implementations to system more expensive, due to these processes being dependent of each manufacturer. This paper proposes a low cost methodology for the implementation of remanufacturing industrial robotic manipulators. In this methodology, mechanical components can be reused and its control systems are adapted to new technologies. Thus, the hardware devices are interchangeable with other processes and the programming language is not proprietary and can be easily modified or replaced by another.

Keywords—robotics manipulators, remanufacturing and open architecture.

1 Introduction

Two historical events completely changed the philosophy of thought production in the automobile industry in the last century. The first event was the creation of assembly line by Henry Ford at 1913 in United States. The second was the use of industrial robotic manipulators in assembly lines of vehicles in Europe and in United States mainly the 70s (Richard, 1995). The development of silicon industry coupled with the growing needs of high productivity and dimensional finishing that meets the increasingly demanding markets have made the use of robotic manipulators expand in almost all industrial sectors.

The amount of robots in manufacturing activities grows as fast as implementations, repairs and maintenance that these robotic manipulators needs. The biggest problem is the closed control architecture used by robotic manipulators manufacturers. Thus, each manufacturer has specific programming language and, sometimes, each robot model and its hardware structure has electronic and electromechanical components that aren't interchangeable between different manufacturers and models. Any intervention depends on the manufacturer and it is consequently much expensive. Therefore, it is better to get a new robotic manipulator and to discard the old one.

In this context, this paper presents a low cost methodology for recycling industrial robots as an attractive alternative for companies that want to reclaim their robotic manipulators and reduce the environmental impact caused by their discards.

An IRB6-S2 robotic manipulator, produced in 1977 by ASEA, was remanufactured to validate the

proposed methodology (Figure 1(a)). Figure 1(b) presents a vehicle produced by this robot. The robot was used in FCAW welding processes to generate a cost analysis of recycling process and verification of precision in position, speed and acceleration.

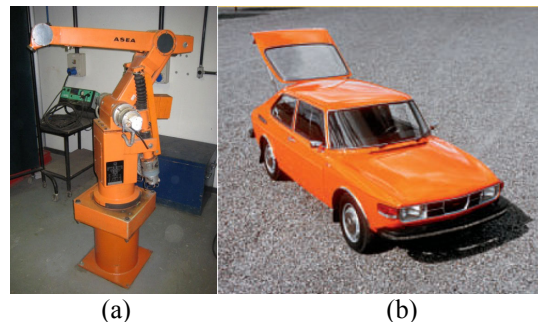


Fig. 1. ASEA IRB6-S2 Robot, 1977 (a) and SAAB Vehicle, 99 model, 1975 (b).

2 Materials and methods

The ASEA IRB6-S2 Robot was widely used in automotive industry in the 70s and 80s and their main features are the capability to handle 6 kg tools, five degrees of freedom arm and harmonics gears based on ratios between 1/128 and 1/158. The industrial robotic manipulators produced in these ages, such as ASEA Robot, generally exhibit mechanical structure in condition to be reused. Sometimes, they require minor repairs as the mechanical structure is designed according conservative failure criteria, like Goodman and Soderberg (Shigley, 2005). In these criteria, the safety factor values arrived at greater than twelve times the yield strength of the material for static loads, yielding high strength and super-sizing of the mechanical structure.

The methodology presented in Figure 2 can be used for recycling robotic manipulators.

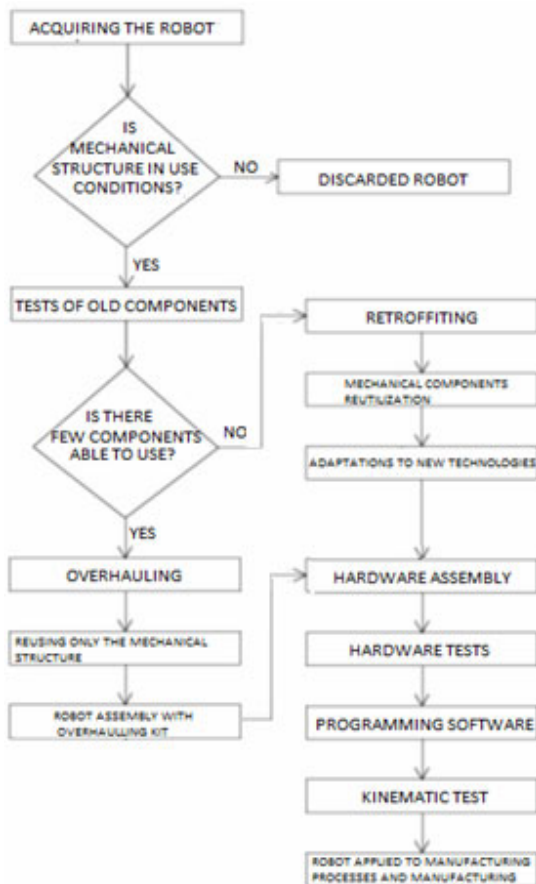








Fig. 2. Flowchart with proposed methodology for remanufacturing process execution.

According to the proposed flowchart, it is viable to reuse mechanical structure of robot manipulator and to change some damaged parts by prolonged use (Lages *et al.*, 2003).

During the tests of old components, Bomfim (2009) and Bomfim *et al.* (2012) proved that there were few electronic components in working condition because these components are obsolete or presented excessive energy consumption, without reliability, high heating producing and emitting noises. Therefore, the remanufacturing was performed using Overhauling process. In this particular case, only mechanical structure recycling is viable.

This paper proposes an “Overhauling-kit”, where hardware structure is replaced. This philosophy is valid on robots with advanced age and there isn’t a historic of working conditions. In these situations, it is difficult to make a favorable or unfavorable report about the reuse of electronic and electromechanical components (Bomfim *et al.*, 2012). Table 1 shows the components of the “Overhauling-kit”.

Table 1. Electronic components constituting the “Overhauling-Kit”.

Description	Amount	Image
Encoder disk + Transmission Module	05	
Driver model G320 Servo Drive.	05	
Set CPU+Monitor	01	
Motor C.C. 70V, 5A. Max speed 1.000 rpm.	05	
Breakout board model C1G.	01	
Power supply 1.750W/70V.	01	

According to a fully open control architecture principle, the program of the robotic manipulator trajectory was implemented using two very common softwares in academia and CNC machines.

The robotics manipulators kinematics (Bracarense *et al.*, 2009) is described by homogenous transformation matrices (Craig, 1989). The motion of TCP would be easily described using mathematical simulation software that performs matrix operations. Therefore, the MATLAB[®] software was selected by well meeting the necessary requirements. In conjunction with MATLAB[®], Mach 3[®] was used because this software is capable of generating setpoint curves for commanding the position and speed of the manipulator, and to generate acceleration and slowdown ramps, controlling simultaneously up to six joints.

3 Results and discussion

The implementation of Overhauling basically consisted of design and construction of hardware driver and the desired trajectory generation and commanding through MATLAB[®] and Mach 3[®] softwares.

3.1 Development of an universal control cabinet (open architecture)

The first step to develop a robotic manipulator controller is to choose what will be the used control architecture at the hardware system. The Figure 3 presents a general vision of the architecture control cabinet. Basically, the whole power driver structure is located at the cabinet, with exception of motors and encoders.

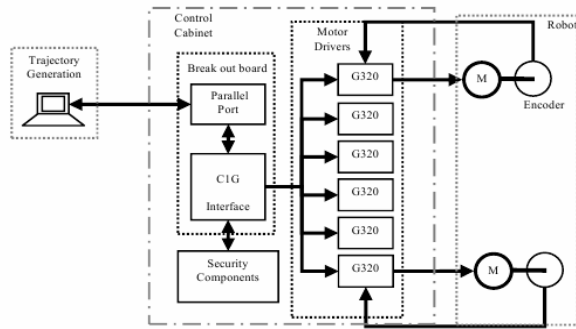


Fig. 3. Control cabinet architecture.

This proposed architecture allows the control cabinet to be interchangeable with other systems. Through the exchange of motors and adjustments on the power supply, this cabinet can be used to drive other systems or robotic manipulators with lower degrees of freedom, such as milling cutter and plasma cutting machines, for instance.

The control cabinet development is divided in four steps: power supply construction, drivers installation, building computer interfaces and final tests.

Power supply construction: the correct way to design the power supply is to consider the peak current of the motors, which mostly occurs during acceleration from zero speed. Thus, the power source must have capability to drive five motors simultaneously, in the worst case. The power source must provide a voltage of 70V and peak current of 25A, totaling 8.750W. The peculiarity of ASEA Robot is that it is used in welding, which requires low motors speeds (100-200 rpm), which drastically reduces the start-up current. Consequently, the start-up current adopted was the nominal operating current (5A), because motors having rotation speed five times less than the maximum speed. Thus, power supply has a total power of 1750W, showed in Figure 4 and Figure 5, orange arrow.

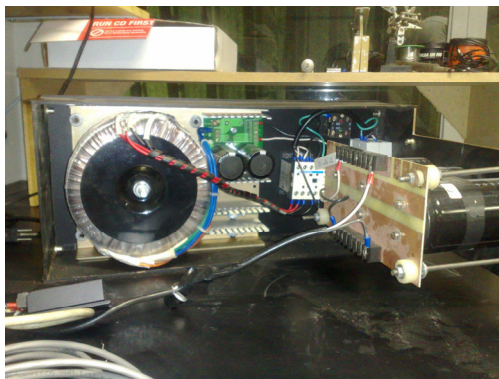


Fig. 4. 1.750W power supply used in experiments.

Drivers installation: the drivers layout was taken in a modular structure to facilitate the traceability of defects and carrying out repairs, showed at Figure 5, green arrow.

Construction of computer interface and connectors: following the methodology of a remanufacturing with low cost, interfacing between the

computer and control cabinet was made through a DB25 connector, red arrow, Figure 5. Besides the parallel port, bornes were connected to each driver, allowing a diversity of forms of control, Figure 5, blue arrow.

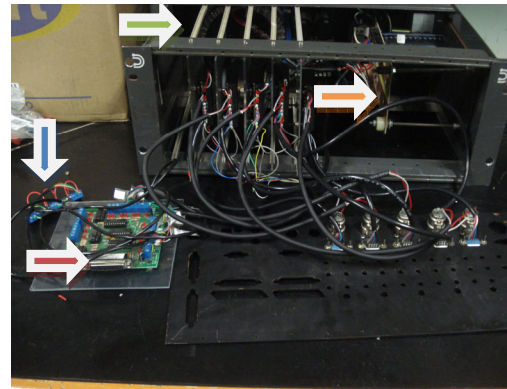


Fig. 5. Control Cabinet in final step of assembly.

Final tests: after control cabinet assembly, it was connected to encoders and motors. In test procedures, each motor was operated independently by a button, shown in Figure 6, and a setpoint of 200rpm clockwise and counterclockwise was generated through PIC18F4550 microcontrollers. In the final test it was placed a 6kg load in the end-effector, and the robot operated for twelve hours without failure in the hardware systems.

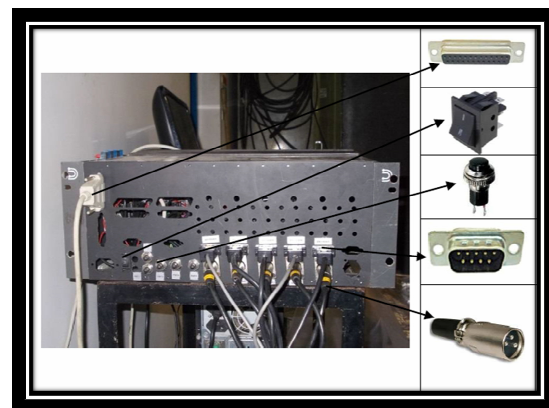


Fig. 6. Control cabinet with open architecture used in experiments.

3.2 Trajectories generation using non-proprietary programming languages

Softwares that use MS Windows[®] platform for the plants automation are widespread in the industrial sector, since this option simplifies graphics applications development, as several tools have already been developed for that operating system. One drawback is that Windows[®] doesn't support real time applications (except Windows RT[®]), because its multi-task architecture can't give total priority for one specific process execution. Another possibility is to use dedicated ports of computers (usually a parallel port, LPT1) with operating systems able to generate applications in real time. Some Linux operating

systems were designed for these purposes, as Linux CNC (Bracarensse *et al.*, 2009). The problem is that is necessary to train users not familiar with these operating systems.

This paper chooses MS Windows® operating system because the programming of robotic manipulator doesn't need a real time operating system, as the control of each joint is made by hardware. As in most industrial applications, to make changes in programmed path it is necessary to stop the process and load a new code with the desired trajectory.

3.3 Direct and Inverse kinematic of ASEA Robot

To describe the position and orientation of a robot manipulator, it can be used Cartesian Space, Joints Space and Actuators Space (Craig, 1989). Cartesian coordinate system (Cartesian Space) defines the position (x, y and z) in the Space and around these axis (roll, pitch, yaw) of TCP orientation attached to the robot. In other words, to generally describe the robotic manipulator positioning there will be necessary six independent coordinates $P(t)=\{x,y,z,r,p,w\}$. In the ASEA case, it is described by only five coordinates: $P(t)=\{x,y,z,r,p\}$. Joints Space uses displacements of variable joints to define the angle values, $\theta(t)=\{\theta_1,\theta_2,\theta_3,\theta_4,\theta_5\}$ and actuators Space use motor's axis angles to define tool position $A(t)=\{A_1,A_2,A_3,A_4,A_5\}$.

Thus, there is a robotic manipulators mathematical model defined in direct and inverse kinematic, where the direct kinematic is the transformation between the Joint and Cartesian Spaces: $\theta(t) \rightarrow P(t)$ and the inverse kinematics determines relationship between Cartesian Space and Joints Space: $P(t) \rightarrow \theta(t)$. These relations are also valid when it comes to direct and inverse actuators kinematics (Craig, 1989).

At Table 2, the Denavit-Hatemberg parameters that were used in the Overhauling project to determine the direct kinematic model are presented and the Figure 7 shows each degree of freedom of the ASEA Robot.

Table 2. Denavit-Hatemberg parameters.

Number of joint	α_{i-1} (°)	a_{i-1} (mm)	d_i (mm)	θ_i
1	90	0	700	θ_1
2	0	450	0	θ_2
3	0	670	0	θ_3
4	90	0	0	θ_4
5	0	0	95	θ_5

After obtaining direct geometric model, the inverse kinematic is determined (Eq .1). In this way, there is a relation between joint angles and manipulator dimensions.

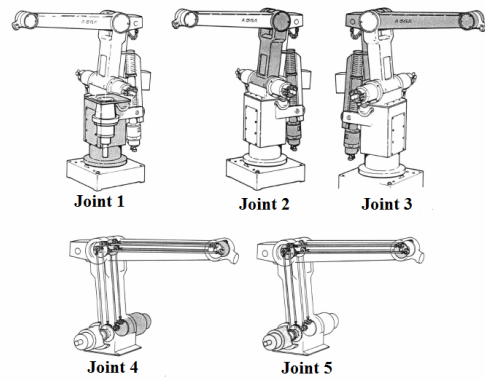


Fig. 7. Degrees of freedom of an ASEA Robot.

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} -S_1 & 0 & C_1 & 0 \\ C_1 & 0 & S_1 & 0 \\ 0 & 1 & 0 & 700 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1_2T &= \begin{bmatrix} -S_2 & -C_2 & 0 & -450S_2 \\ C_2 & -S_2 & 0 & 450S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^2_3T &= \begin{bmatrix} S_3 & C_3 & 0 & 670S_3 \\ -C_3 & S_3 & 0 & -670S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^3_4T &= \begin{bmatrix} -S_4 & 0 & C_4 & 0 \\ C_4 & 0 & S_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^4_5T &= \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & 95 \\ 0 & 0 & 0 & 1 \end{bmatrix} & & (1)
 \end{aligned}$$

$${}^0_5T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T = \begin{bmatrix} S_1S_{234}C_5 + C_1S_5 & -S_1S_{234}S_5 + C_1C_5 & -S_1C_{234} & 450S_1S_2 - 670S_1C_{23} - 95S_1C_{234} \\ -C_1S_{234}C_5 + S_1S_5 & C_1S_{234}S_5 + S_1C_5 & C_1C_{234} & -450C_1S_2 + 670C_1C_{23} + 95C_1C_{234} \\ C_{234}C_5 & -C_{234}S_5 & S_{234} & 700 + 450C_2 + 670S_{23} + 95S_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\theta_1 = \tan^{-1} \frac{-p_x}{p_y}$$

$$\theta_3 = \tan^{-1} \frac{S_2}{C_2}, S_3 = \frac{w_1^2 + w_2^2 - (l_2^2 + l_3^2)}{2l_2l_3}, C_3 = \sqrt{1 - S_3^2},$$

$$\theta_2 = \tan^{-1} \frac{S_2}{C_2}, S_2 = \frac{w_2l_2C_3 - w_1(l_2S_3 + l_2)}{l_2^2C_3^2 + (l_2S_3 + l_2)^2}, C_2 = \frac{w_1l_2C_3 + w_2(l_2S_3 + l_2)}{l_2^2C_3^2 + (l_2S_3 + l_2)^2}$$

$$w_1 = -S_1p_x + C_1p_y + \lambda_5S_1a_x - \lambda_5C_1a_y,$$

$$w_2 = p_z - \lambda_1 - \lambda_5a_z$$

$$\theta_{34} = \tan^{-1} \frac{\lambda_5S_{34}}{\lambda_5C_{34}},$$

$$\lambda_5S_{34} = S_1S_2p_x - C_1S_2p_y + C_2p_z - \lambda_1C_2 - l_2 - l_3C_3$$

$$\lambda_5C_{34} = -S_1C_2p_x + C_1C_2p_y + S_2p_z - \lambda_1S_2 - l_3C_3$$

$$\theta_4 = \theta_{34} - \theta_3,$$

$$\theta_5 = \tan^{-1} \frac{S_5}{C_5}, S_5 = C_1n_x + S_1n_y, C_5 = C_1o_x + S_1o_y$$

Where: $S_n = \text{Sine joint } n$; $S_{nmr} = S_n * S_m * S_r$; $C_n = \text{Cosine joint } n$; $\theta_n = \text{Angle joint } n$; $\lambda_n = \text{Offset joint } n$; $T = \text{Homogeneous transformation matrix}$; $P_n = \text{Position in cartesian axis } n$.

When a programmer develops a program to generate trajectories for robotic manipulators, the first step is the construction of direct kinematics. The function of actuators' direct kinematic is providing a relationship between motors rotation and angles that will move joints. The kinematics provides a direct

relationship between angles and position in Cartesian Space. After the creation of the direct kinematics, the inverse kinematics is also modeled, and the program has a relation between Cartesian coordinates and the angle that the motors must move to achieve desired position. The resulting motion is the displacement of the robot TCP programmed path. Thus, when the operator generates a trajectory in particular software, there is an algorithm that relates the Cartesian coordinates with the amount and rate at which pulses are to be sent to the motors controllers.

Despite programming the trajectory of a robot manipulator is relatively easy, the kinematics involved in a simple linear interpolation is much more complicated to be performed, as will be seen in following topics.

3.4 MATLAB® - software used to generate G code

MATLAB® was selected in this work because it is a mathematic simulation software that performs operations involving linear algebra. Therefore, MATLAB® is indicated to perform matrix operations including kinematic of industrial robotic manipulators. Authors opted to implement each routine without using specific toolboxes, to maintain the control architecture as open and didactic as possible.

Figure 8 shows the G-code generated by MATLAB® program. In this program, it's possible to program the dimensions of the tool to be used in TCP robot, the displacement speed and the trajectory's start point. After these lines, inverse kinematics and inverse actuators kinematics are used by the program. A file is generated in ".txt" format with G-code, where the displacement is defined for each axis.

```

1 G1 X -15815 Y 26937 Z 100506 A -262144 B 865 F1290000
2 G1 X -12661 Y 26606 Z 100530 A -262144 B 2547 F 215387
3 G1 X -9501 Y 26348 Z 100548 A -262144 B 4237 F 215569
4 G1 X -6336 Y 26164 Z 100561 A -262144 B 5934 F 215739
5 G1 X -3169 Y 26053 Z 100569 A -262144 B 7638 F 215896
6 G1 X 0 Y 26017 Z 100572 A -262144 B 9347 F 216042
7 G1 X 3169 Y 26053 Z 100569 A -262144 B 11061 F 216176
8 G1 X 6336 Y 26164 Z 100561 A -262144 B 12778 F 216300
9 G1 X 9501 Y 26348 Z 100548 A -262144 B 14499 F 216415
10 G1 X 12661 Y 26606 Z 100530 A -262144 B 16223 F 216521
11 G1 X 15815 Y 26937 Z 100506 A -262144 B 17948 F 216620
12 G1 X 0 Y -902 Z 52688 A -262144 B 48421 F1290000
13
    
```

Fig.8. G-code generated in MATLAB by inverse kinematics.

3.5 Mach 3® – software to pulses generation

To develop friendly interfaces between operator and CNC machines, many softwares were developed to implement the movement control using G-code. In this scenario, the software Mach 3® is widespread in the industry and runs in MS Windows®. Mach 3® is a very flexible program, designed to control machines such as

mills and plasma cutters.

Mach 3® was created to control Cartesian machines which present a linear relationship between Actuators Space and Cartesian Space. The movement of actuator is directly proportional to displacement in Cartesian direction that it control.

One of the most attractive characteristics of Mach 3® is the ability to simultaneously control six axes. If the mechanism to be controlled is a robot arm which generally has six degrees of freedom and a shaft connected at each motor, Mach 3® can control them, since the operator performs kinematic calculations to relate the position of tool Cartesian coordinates to the length and rotation of the robot arms. In this work, this task is performed by the routines created in MATLAB®.

3.6 Trajectory generation

Figure 9 shows the control loop in block diagrams. The operator enters the initial values of trajectory of the homogeneous transformation matrices presented in the trajectory generation developed in MATLAB®. After MATLAB® generates the sequence of G-codes, it is loaded in Mach 3®. The function of Mach 3® is generating pulses which will command each driver. This software is also responsible for generating acceleration and deceleration ramps for each joint.

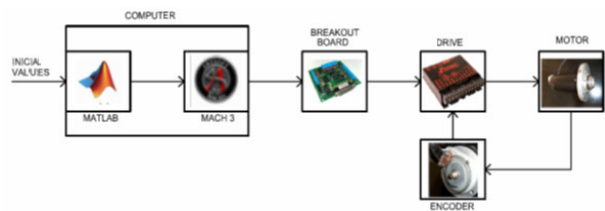


Fig.9. Control loop diagram to drive a motor.

The connection between drivers and computer is made through a Breakout Board that allows distributing the signals of step and direction from the parallel port of the PC to terminals which are connected to drivers (GECKODRIVE, 2012). The Breakout Board interface also serves to isolate the noise generated by motors using optoisolators and emergency stops which can be easily connected to presence detector sensors, load sensors or end of course sensors (CNC4, 2012). Figure 10 shows joint movement obtained by the programming.



Fig.10. G-Code generated in MATLAB® controlling joint movements through Mach 3®.

After this movement, complex robot trajectories was generated with marking landmarks on a table, to check the accuracy in positioning and speed, which can be verified by Figure 11. It was observed that the ASEA Robot obtained small errors in positioning accuracy. Because of this, tuning the PID parameters in controller was necessary.



Fig. 11. Reference points Marking for positioning verification.

3.7 Tuning and performance characteristics in PID-controllers

Tuning of proportional, integral and derivative control actions is a difficult task in this type of plant. The position of the joints varies the mathematical model. This occurs because industrial robots are a complex system with dynamic couplings between joints, because of gravitational force, inertial and Coriolis effects, centrifugal torques and viscous damping these couplings are nonlinear.

These characteristics make mathematical models used in the literature, such as Ziegler-Nichols Method, for instance, to be difficult to implement physically, because these methods consider behavior of plant's transfer function to be linear and its coefficients to be invariant at the time.

Due to dynamic behavior of robot manipulator, a heuristic method for setting the control actions was adopted. This adjustment basically consists of mapping the most severe operating configurations of the robot and calculation gains of control actions for each motors configuration. This approach proved to be successful due to the fact the driver used in the experiments has a very high robustness, which has been empirically proven in other projects developed in LRSS (Pena, *et al.*, 2011).

The most severe operating point was obtained from the application of a step voltage at 2V with manipulator with a load of 6kg in TCP. The pose to produce slower response time is severe operating point. In this position, the PID controller parameters were calibrated. A multi-turn potentiometer was used as a displacement sensor of the joint, which can be seen in Figure 12.

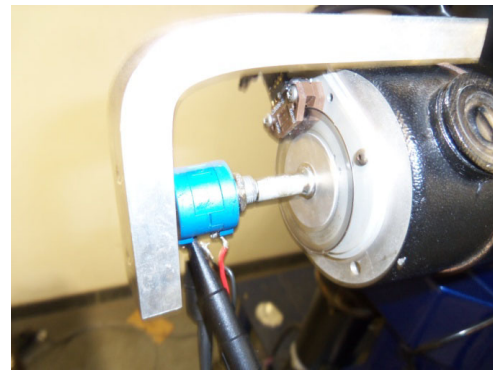


Fig. 12. Multi-turn potentiometer linked in motors shaft.

The calibration of PID parameters is made moving the cursor potentiometers of the control drivers to set proportional, integral and derivative gains.

The calibration parameters of the PID was performed as follows:

First step: Set the control action P.

Reason: In control systems, the P action is always the first tuned because other control actions act on the error generated by this action. The gain P was adjusted to position 3 (90 degrees of the potentiometer) because for values above 5, the controller enters in failure because this is the critical gain (K_c) where oscillation is quite high. To simplify the results, tuning applied will be represented in form P / I / D. In this case, the tuning is 3/0/0.

Second step: 2 Set the control action D.

Reason: D action provides the acting bug, initiates corrective action and increases the system stability, which allows to use higher values of K_p and K_i , resulting in greater accuracy in continuous operation. The selected action for D setting was set to 1. This value was selected because values greater than 1 didn't show improvements at system response. Thus, the tuning adopted was 3/0/1.

Third step: 3 Set the control action I

Reason: The system residual error is around 5.5%. Consequently, the action I was set to value 1, which resulted in a steady error of 2.4%. When the action I was set to 2, it results in a stationary error of 0.4%. This tuning value was adopted because a result higher than 2 didn't have effect in error values. For values greater than tuning 6, the system goes into instability, which could be checked with an abrupt rotation of shaft 5.

This heuristic methodology was adopted due to the fact that the adjustment of PID parameters is achieved via potentiometers (Fig 13). Because of this, the application of advanced control techniques is impossible, such as Artificial Neural Networks and Fuzzy Logic, for instance.

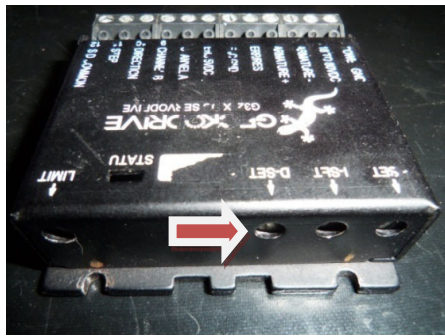


Fig. 13. Servodrive G320 with potentiometers in evidence.

For an example, motor five tuning will be shown. In Figure 14, it can be seen that tuning (3/0/0) exhibits high oscillation and high residual error. With 3/0/1 tuning, the system response with damping, attributed to the effect of derivative control. In the figure, it can be analyzed that the derivative action does not affect the steady error, because the curves of 3/0/0 and 3/0/1 tunings have the same divergence in error when compared to setpoint (input signal). To mitigate the residual error, the action of integrative control was tuned (tuning 3/2/1) and the error in displacement was virtually eliminated, which can be seen by the Figure 14.

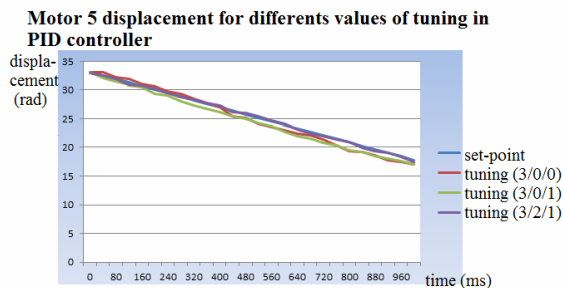


Fig. 14. Motor 5 response to an entry ramp, with different PID controller tunings.

3.8 Welding execution with FCAW

To validate the proposed method, weld beads were done using a welding machine used for manual welding applications (Smashweld Model 408, manufactured by ESAB). Table 3 presents the result of experiments and the welding parameters used in it.

Table 3. Welding parameters for carrying out the experimental procedure.

Nr Experiment	Welding Velocity (mm/s)	Welding voltage (Volts)	Size of Stick-out (mm)	Welding current (Ampers)
Exper. 1	20	23	18	200
Exper. 2	20	25	18	250
Exper. 3	10	25	18	324
Exper. 4	15	25	18	304
Exper. 5	10	33	18	336
Exper. 6	8	26	18	320
Exper. 7	8	32	18	348
Exper. 8	10	33	18	328

The parameters of Experiment 5 presented the best features of weld bead. This bead can be seen in Figure 15. In this experiment, the penetration and bead

width had the dimensional parameters substantially invariants in time, and high productivity was obtained. In Experiment 5, the welding parameters used were: welding current – 336A; welding voltage – 33V; torch displacement speed – 9.6 mm/s; step – 10 mm; stick-out – 18mm and weld bead length – 150mm.



Fig. 15. Weld bead obtained in Experiment 5.

3.9 Economic Viability

A new robotic manipulator that presents similar characteristics of ASEA Robot costs about US\$35.000,00. In this project, shown in the Table 3, the total cost of remanufacturing was US\$2.000,00. This price justifies the remanufacturing method of this article.

Table 4. Description of component's cost used in Overhauling process.

Description	Amount	Producer	Cost (US)
Incremental Encoder	05	US-Digital	150,00
Driver - model G320	05	Geckdrive	750,00
CPU + monitor	01	LG	300,00
DC Motor	05	Kelling	500,00
Parallel Interface - C1G model	01	CNC4	50,00
Power Supply 2.100W	01	-	250,00
Total cost			2.000,00

It is important to accentuate that Overhauling process of this article is an academic solution. The main idea is to create an universal control cabinet, where software and hardware can be easily replaced or implemented, to validate scientific works, like application with computational vision, for instance.

For industrial application, softwares like MATLAB® must be replaced, due to high cost of licenses. A better alternative is to use programs for Linux platform, like Robotic Operating System (ROS), Robotic Construction Kit (ROCK) and EMC2. These softwares are open source and don't increase costs to remanufacture process.

Figure 16 shows the robotic cell constituted by ASEA Robot after Overhauling process. With this cell, 2500 cycles were simulated for 12 hours. The results were satisfactory because the manipulator obtained good repeatability and accuracy (Fig. 17), when comparing with a robotic manipulator model KUKA KR6 manufactured on 2012 (KUKA, 2012).



Fig. 16. ASEA Robot after Overhauling process.

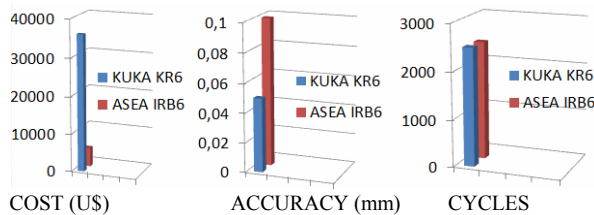


Fig. 17. Comparison between KUKA KR6 and ASEA Robot in terms of cost, accuracy and repeatability.

4 Conclusion

Using the methodology described in this article, it can be developed a standard platform for recycling robotics manipulators. This platform becomes a viable alternative considering that the final cost of remanufacturing is much smaller than equivalent new equipment.

In conducting the remanufacturing, there are two important concepts that must be evaluated before the process is started. In one line of research, Retrofitting seeks the maximum reuse of robot systems that are in operation, making adjustments and upgrades to new technologies. In another philosophy, Overhauling provides only the reuse of the mechanical structure of robots that have advanced age. Thus, the process of Overhauling was adopted for implementation of ASEA Robot, as the robot and its electromechanical and electronic components didn't have historic of use and it is a complex task to raise a technical opinion favorable or unfavorable to its reuse.

During experimental phase, the ASEA Robot showed excellent accuracy of trajectories in execution and implementation of weld beads by FCAW process. Performing a cost analysis, it was observed that "Overhauling-Kit" represents only 5,7% of a new robot value with similar characteristics to market price of US\$35.000,00. Thus, the remanufacturing process provides a benefit/cost ratio very high.

Another important addition to remanufacturing process was that, during the selection of hardware and software, it adopted an open architecture and a non-proprietary programming language, where kinematic generation is not coupled to a dedicated operating

system. The MATLAB[®] programming language facilitates changes implementation on the system and implementation of new tasks and control cabinet is interchangeable with other plants.

For more information see the web site address <https://www.youtube.com/watch?v=s4QJFVUEkxw>.

Acknowledgments

The authors would like to acknowledge the LRSS/UFGM (Robotic, Welding and Simulation Laboratory of the Federal University of Minas Gerais) for having provided the necessary structure to perform this work.

References

- Chase, R. B.; Aquilano, N. J., *Gestão da produção e das operações: perspectiva do ciclo de vida*. Lisboa: Editora Monitor, 1995, p.21.
- Shigley, J.E.; Mischke, C.R.; Budynas, R.G., *Projeto de Engenharia Mecânica*. 7ª Edição. Porto Alegre: Editora Bookman, 2005, p. 960.
- Lages, W.F.; Henriques, R.V.B.; Bracarense, A.Q., 2003. "Arquitetura Aberta para Retrofitting de robôs". Manet Notes Workshop, Bragança Paulista, SP, Brazil.
- Bomfim, M.H.S., *Projeto de um gabinete de comando com interface USB para robôs de até seis graus de liberdade*. Trabalho de conclusão de curso. Programa de Graduação em Engenharia Mecânica da UFGM, Belo Horizonte, 2009, p. 46.
- Bomfim, M.H.S.; Bracarense, A.Q.; Lima II, E.J.; Gontijo, R.A., "Overhauling of an ASEA robot IRB6 with open architecture". 11th EIII International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom-2012). Liverpool, UK, 2012, pp. 482-489.
- Bracarense, A. Q.; Cordeiro, J. F.; Kienitz, K. M.; Bomfim, M. H. S.; Souza, A. J. A.; Ramalho Filho, F. A.; Lima II, E. J., "Development of control cabinet for industrial robots up to six degrees of freedom". 20th International Congress of Mechanical Engineering. Gramado, BR, 2009, pp. 427-432.
- Craig, J.J. *Introduction to Robotics Mechanics and Control*, second edition, Addison-Wesley Publishing Company, Inc, 1989.
- CNC4, *Breakout boards*, March 26, 2012, <http://www.cnc4pc.com>.
- G320-REV-7 Manual, *Installation Notes*, March 3, 2012 <http://www.geckodrive.com>.
- Kuka Robotics, KR6 model, March 3, 2012. <http://www.kuka-robotics.com/en/>.