Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

RETROFITTING OF ASEA IRB2-S6 INDUSTRIAL ROBOT USING NUMERIC CONTROL TECHNOLOGIES BASED ON LINUXCNC AND MACH3-MATLAB

Alberto J. Alvares[1], Toquica, J. S.[1], Eduardo José Lima II[2], Marcelo Henrique Souza Bomfim[2]

Abstract— **This work presents a comparative study of numerical control machines controllers integration using LinuxCNC controller and MatLab/Mach3 in Windows solutions applied in retrofitting process of an old industrial robot. The experience associated with retrofitting technique with ASEA IRB6-S2 by Federal University of Minas Gerais (UFMG) and University of Brasilia (UnB), have adopted different robot control platforms alternatives to include robot kinematics. Thus, the article makes a comparative study of performance in terms of financial and computational cost, implementation tasks facility, integration with CAD/CAM and usability, analyzing the advantages and disadvantages of each architecture and finally chooses the best alternative CNC (Computer Numerical Control) controller.**

I. INTRODUCTION

This work presents a comparative analysis for two retrofitting physical and logical architectures for ASEA IRB6-S2 robotic arm. Both architectures were based on a low cost solution using numerical control technologies. IRB6-S2 is a robot manufactured by ASEA, actual ABB. It has 5 degrees of freedom (DOF) and 6kg maximum payload. The first solution was developed at UFMG by the Robotics Welding and Simulation Group (Grupo de Robótica Soldagem e Simulação - GRSS), based on the Mach3 Numerical Command (command for positioning of the robot TCP based on G code) and MatLab (generation of forward and inverse kinematics based on Homogeneous transformation matrices).

The second solution was developed by UnB at the Innovation Group in Industrial Automation (Grupo de Inovação em Automação Industrial - GIAI) and it is based on LinuxCnC controller and the integration with robot kinematics equations (forward and inverse), are including directly in LinuxCNC controller (before 2000 named as EMC2 - Advanced Machine Controller) [1].

The two universities have the same robot model; the ASEA IRB6-S2 robotic arm was donated by Fiat Automotive Enterprise. UnB uses the same hardware specification adopted by UFMG, being associated with the update of the five DC motors and encoders, drive with PID (Proportional–Integral–Derivative) controller based on Gecko drive technologies and the same power supply and parallel port interface. The implementation of numerical control technology controllers are based on the main difference between the two approaches.

UnB opted for LinuxCNC platform that allows programming in C, C++ and Python languages used for including non-trivial kinematics for mainly with numerical control machines, in this case ASEA Robot. Thus, it was possible to implement the forward and inverse kinematics for the robot links and the inverse kinematics for the joints commands in an integrated way, that is, the CNC (Computer Numeric Control) controller interprets a G/M file generated by a CAD/CAM system for a five DOF (up to nine DOF), and the LinuxCNC controller executes the trajectories in real time, without the need additional plugins or adapters.

In the UFMG solution, two licensed softwares were used. MatLab is responsible for calculate the kinematics of joints and actuators, based on a user-defined point list. This list of points can be generated, for example, by robot simulation software such as Workspace, commercial CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) software or a computer vision trajectory definition program. MatLab® generates an output G-code program with the trajectory to be followed by the robotic arm on coordinate workspace. Then G/M file is interpreted by Mach3® CNC command software, which "understands" that it is commanding a Cartesian machine, since the points are defined in the coordinate workspace [2].

Two architecture alternatives are inexpensive and robust enough to be used as low-cost strategies to update the industrial robot controller, which are currently out of order, even though the mechanical structure is in operational condition.

II. RELATED WORKS

The ASEA manipulator was the first patent for an anthropomorphic industrial robot dating from 1973. From this patent, industrial robots for complex tasks began to be produced using this type of morphology. Due to the complex and dynamic studies in articulated robots, it is common that a safety factor used in the design of machine elements and connection links reach ten times the limit for static loads [3]. Consequently, it is possible to find currently, old robots with decades of use and that present the mechanical structure in good condition to be reused. In this way, it becomes an interesting task to reuse this mechanical structure by upgrading the hardware and software to adapt it to new

[1] University of Brasilia, Mechanic and Mechatronic Engineering Department, Brasilia, DF, Brazil. alvares@AlvaresTech.com, jstoquica@gmail.com
[2] Federal University of Minas Gerais, Mechanic Engineering Department, Belo Horizonte, MG, Brazil. eduardo@demec.ufmg.br, sgtbomfim@yahoo.com.br

978-1-5386-3742-5/17/$31.00 © 2017 IEEE

technologies, such as communication protocols, interfaces and connections in industrial networks, for example [1].

A retrofitting process for industrial robot IRB2000 [4] was concluded by assembling of a control cabinet that allows an open architecture independent of control algorithms and power system provided by manufacturers, as an example, a CAN bus standard was used to enable connections of drives robot. Monitoring parameter and manual joint control are available with a user interface developed in Simulink/MatLab.

Based on RTAI (Real Time) controller, LinuxCNC has continuous developments by a CNC expert community. It was possible to produce parts in low-density materials with an Open Architecture Controller (OAC), resulting in additional simulation with 5 DOF manipulator [5]. Machining operations are based on spherical cutters machines.

Another solution developed with LinuxCNC was completed for a 6 DOF robot, allowing the simulation of a trajectory generated through a RS-274 standard file. The work only developed the software layer, but contributes to the use of LinuxCNC in open control architecture environments with Open Source platform [6].

Contributing with OAC, Lima Silva et al. [7] developed an application based on ISO 7498-1 standard (OSI layers) for embedded system, where the programming tasks to control was done in high level programming, decoded to DSC processors, setting the control 6 DOF robot movements. With the modular capability of that work, it is possible the integration with other manipulators and new functionalities [7].

In works developed by Jokic and Lubura [8] a comparative study was made comparing the use of a controller with PC (Personal Computer), FPGA (Field Programmable Gate Array) and MARK I Controller. The controller is designed for controlling robot PUMA 560.

In works development by Cheruiyot [9] the system uses a framework for kinematics and dynamics control. This framework is OROCOS (Open Robot Control Software) and proved to be particularly useful for research as well and commercial applications. It is notable for its many features and libraries such as: XML configuration, scripting, state machines; CORBA transports pre-existing OROCOS libraries (such as the Kinematics and Dynamics Library (KDL) and the OROCOS Component Library (OCL)).

The financial investments for robot retrofitting technique can be up to 5% of the current value of a new robot with characteristics similar to those of an ASEA Robot [3]. In this way, it is possible to take advantage of a mechanical structure of robotic arm, replacing control components, allowing total compatibility with digital technologies.

III. RETROFITTING: HARDWARE SPECIFICATION

With an updated process and with a proposal for robot retrofitting, where it is used compatible parts with actual technologies. These components are usable for manipulators that are likely to be functional again in an academic or industrial setting [2]. Table I presents digital components used for a technological update process in UFMG and UnB, total cost was approximately one thousand dollars (`http://www.automationtechnologiesinc.com/`).

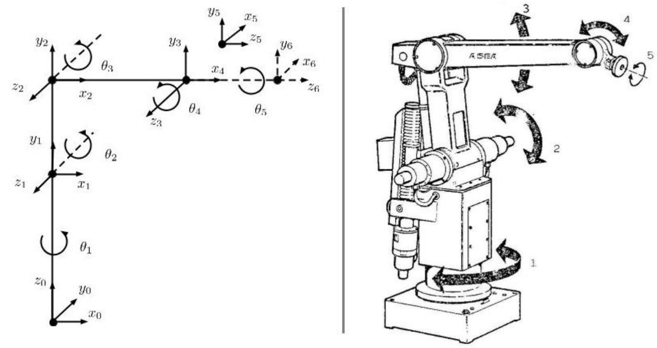

Fig. 1. Definition of a coordinate system for ASEA Robot.

TABLE I

COMPONENTS USED IN RETROFITTING PROCESS

Item	Description	Quantity
1	Parallel port for CNC	1
2	Servo-drive Gecko G320X	5
3	Motor KL23-130-60	5
4	Encoder ATM 102	5
5	Power Supply with 1200W	1
6	Desktop Pentium 4, 1 GHz with LPT1	1

Before the components were mounted in the robot structure and control cabinet they were tested to verify a good condition of each one, as well as initial configurations, according to manufacturer user's manual. Components described above are compatible with any operating system, ensuring OAC is independent with software to be used to control and generation robot movement signals.

IV. ASEA ROBOT FORWARD AND INVERSE KINEMATICS MODEL

A. Denavit-Hartenberg (DH) notation

According to the parameters obtained by the Denavit-Hartenberg (DH) notation, according to the adjacent joints of the robot, it is possible to establish the DH parameters [10]. Fig. 1 shows the definition of coordinate systems at each joint.

Table II shows the DH values of the ASEA manipulator.

TABLE II

DH PARAMETERS FOR ASEA ROBOT

Joint	α_i	$a_i[m]$	$d_i[m]$	θ_i	Limit
1	90	0	$\lambda_1 = 0.70$	θ_1	340°
2	0	$l_2 = 0.45$	0	θ_2	± 0°
3	0	$l_3 = 0.67$	0	θ_3	± 25°
4	90	0	0	θ_4	± 90°
5	0	0	$\lambda_5 = 0.095$	θ_5	± 180°

B. Kinematic model using DH Notation for ASEA Robot

To define joint variables of the robot according to the final coordinates of the tool, must necessary to know the inverse kinematic equations of the manipulator , thus obtaining the homogeneous equations to define the joint variables of the system. It allows, in this way, to insert the equations in any controller, however for the present work it was included at MatLab and LinuxCNC. After defining DH parameters, it is possible to know the system of inverse kinematics equations of robot at coordinate workspace [11]. In 1 are shown the equations:

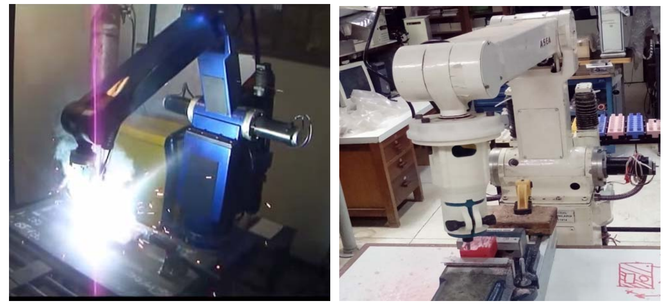

Fig. 2. ASEA manipulator after the retrofitting process: a) UFMG (Welding) and b) UnB (Machining)

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

$$\theta_3 = \tan^{-1}\frac{S_3}{C_3} \tag{1b}$$

$$S_3 = \frac{w_1^2 + w_2^2 - (l_2^2 + l_3^2)}{2 l_2 l_3} \tag{1c}$$

$$C_3 = \sqrt{1 - S_3^2} \tag{1d}$$

$$\theta_2 = \tan^{-1}\frac{S_2}{C_2} \tag{1e}$$

$$S_2 = \frac{w_2 l_3 C_3 - w_1(l_3 S_3 + l_2)}{l_3^2 C_3^2 + (l_3 S_3 + l_2)^2} \tag{1f}$$

$$C_2 = \frac{w_1 l_3 C_3 + w_2(l_3 S_3 + l_2)}{l_3^2 C_3^2 + (l_3 S_3 + l_2)^2} \tag{1g}$$

$$w_1 = -S_1 p_x + C_1 p_y + \lambda_5 S_1 a_x - \lambda_5 C_1 a_y \tag{1h}$$

$$w_2 = p_z - \lambda_1 - \lambda_5 a_z \tag{1i}$$

$$\theta_{34} = \tan^{-1}\frac{\lambda_5 S_3 4}{\lambda_5 C_3 4} \tag{1j}$$

$$\lambda_5 S_{34} = S_1 S_2 p_x - C_1 S_2 p_y + C_2 p_z - \lambda_1 C_2 - l_2 - l_3 C_3 \tag{1k}$$

$$\lambda_5 C_{34} = -S_1 C_2 p_x + C_1 C_2 p_y + S_2 p_z - \lambda_1 S_2 - l_3 C_3 \tag{1l}$$

$$\theta_4 = \theta_{34} - \theta_3 \tag{1m}$$

$$\theta_5 = \frac{S_5}{C_5} \tag{1n}$$

$$S_5 = C_1 n_x + S_1 n_y, C_5 = C_1 o_x + S_1 o_y \tag{1o}$$

V. MACH3 CNC ARCHITECTURE AND FORWARD AND INVERSE KINEMATICS IN MATLAB

In the solution based on MatLab and Mach3 integration, the trajectory data that the manipulator must follow is generated through a *.m script, where the user enter the position, velocity and interpolation type variables between the simplex points, linear or circular path, of the robot. After the entry data, the script in MatLab calls the functions of inverse kinematics (IK), joint space, and in sequence the inverse kinematics of actuators (IKA), space of actuators.

The main purpose of IK is informing the value of each angle of the joints to determine the pose of the manipulator. The IKA makes a relationship between the Joint Space and the characteristics of the motors and sensors used in the system. According to the implemented architecture, Gecko drives (5), model G320, the IKA does the conversion of degrees to pulses, according to the encoder resolution. Feedback pulses generate a control of the system more simplistic, considering that the PID logic control is performed by counting the feedback pulses. Fig. 2 shows the UFMG and UnB ASEA manipulators after the implementation of retrofitting technique, applications on welding (UFSC) and machining (UnB).

The Mach3 is intended to generate the step and direction pulses, along with acceleration and deceleration ramps, for the drivers.

VI. LINUXCNC ARCHITECTURE WITH INTEGRATED KINEMATICS EQUATIONS

A. Open Architecture Based on LinuxCNC

LinuxCNC is a project initially developed by NIST (National Institute of Standards and Technology), which eventually generated a community of volunteer developers and contributed to the fact that the source code could be published and thus made possible to be modified in June 2000. The project is called GNU/Linux licensed controller, capable to control CNC machines and systems with non-trivial kinematics [12].

The LinuxCNC high level logical structure consists in 4 main modules with independent tasks, with a main layer called HAL (Hardware Abstraction Layer), which is an interface between software and hardware, which provides real time transfer of LinuxCNC data to control hardware or low-level logic modules through NML messages [13].

One of the greater benefits of the HAL layer is the ability to control LinuxCNC internal modules as black boxes allowing the exchange and alteration without modification of hardware part control system. Therefore it is possible simulate and test the system responses with approximately results to reality, allowing detect and correct failures that can damage the hardware system components [14].

The control of the entire system is programmed and implemented in LinuxCNC, taking advantage of software-related features such as stability and throughput [15]. LinuxCNC has the key feature of source code is public and serving for any tweaking, editing and later compilation with custom configurations. The same contribute to have an alternative of open and free control architecture (OAC), independent of proprietary and closed protocols or systems. Fig. 2 shows the ASEA UnB Manipulator after the retrofitting process.

B. ASEA Simulation in LinuxCNC

After including the forward and inverse kinematics of the ASEA manipulator on LinuxCNC directly in controller source code based in C++ programming language, it is possible in Fig. 3 to verify the virtual robot design and how it is following the proposed trajectory.

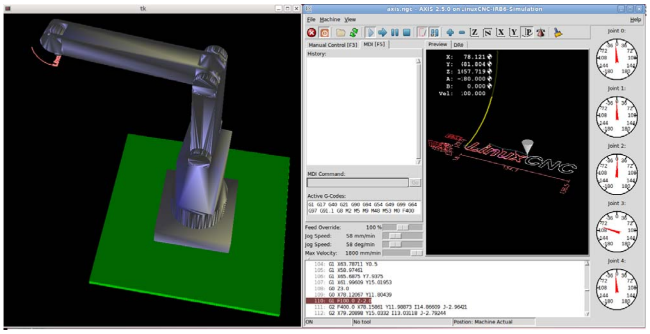

Fig. 3. ASEA robot tracing in G code with the Virtual Manipulator

With the completed simulation, the correct performance of the robot movement is verified through the homogeneous equations in the manipulator inserted in LinuxCNC. It is possible to generate specific trajectories for the positioning of raw and finished parts in LinuxCNC with instructions based on G code, allowing the effective integration with the FMC for the desired positioning of parts.

C. Robot Controller Upgrading

To allow control through LinuxCNC controller, it was necessary to implement digital technology, because the original operation of the robot was by means of totally analog technology. To allow this transition from analog to digital, it was necessary to disassemble the actuators of each DOF, harmonic-drive gears, as well as the resolvers that allowed the control of rotation of each one. When dissemble the robot was made, it was possible to verify the good state of the mechanical structure, thus allowing the integration of new actuators, encoders, drives and parallel breakout board compatible with digital technology, confirming the retrofitting methodology flexibility for manipulators with Linux and RTAI operating systems, where robots technological evolution has focused on improve accuracy of joint sensors, as well as on the hardware and software control systems to improve the final positioning accuracy of the robot end-effector [1].

VII. ASEA IRB6-S2 Robot Results After Retrofitting Implementation

A. LEGO parts manipulation with teach-in programming - LinuxCNC

To validate the robot performance after successful implementation of retrofitting technique was used a on-line robot programming method, known as teach-in for manipulation of LEGO parts. In the Figure 4 is shown a robot operation to carry a LEGO part from initial to end point. That task was performed before retrofitting process, when robot was still operational.

Fig. 4. Parts manipulation after retrofitting process

It was used for part manipulation experiments a Python teach-in module to record desired points, moving the robot in LinuxCNC manual mode. After that was generated a NC program that was successfully executed, allowing to follow a proposed trajectory for part positioning. In this way was performing a basic robot task, validating a retrofitting process through original FMC robot role [3].

B. Capability analysis and geometrical errors - LinuxCNC

Based in NC program generated by CAD/CAM and interpreted by LinuxCNC, was possible estimate a repeatability of robotic arm after retrofitting doing 10 continuous traces with same trajectory, in relation with an unique pen trace thickness. The proposed experiment is presented in Figure 5 where are shown 1 (top) and 10 executed same NC program (bottom).

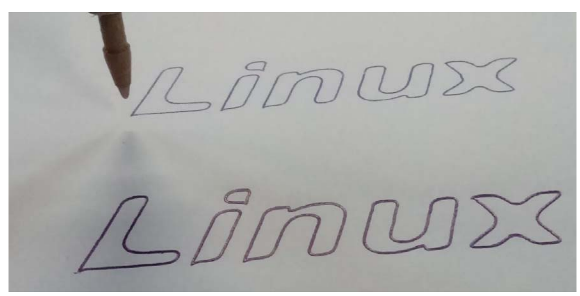

Fig. 5. Experiment for repeatability estimation

After thickness analyses was possible to obtain an approximately repeatability error of 0.28 mm in relationship with original ASEA robot error with 0.2 mm [16]. Being a acceptable error for a old robot that was built 40 years ago. The previous results allow to do a compassion with the actual robot performance, for example the KUKA KR6/2 robot with 0.1 mm of repeatability error [17] and the ABB 6660 robot with 0.07 mm of same performance parameter [18]. Aiming to reduce the repeatability error is necessary to apply a advanced calibration techniques.

With the retrofitting was possible identify parameters that affect the robot performance with repeatability experiment, for example table instability with sheet of paper when robot did the marks generating larger thickness than calculated average (0.532 mm). It is possible improve the obtained results with a better experiment setup [3].

Through geometrical analyses was possible to know the straightness error between commanded distance and executed distance of robot over X and Y axis. In the Figure 6 is presented a proposed experiment with robot and adapted pen, as a tool. The straightness error was among 0.2 and 0.4 mm approximately, assuring the robot after retrofitting has capabilities to do straight lines with a controlled and low error in compassion with modern robots.

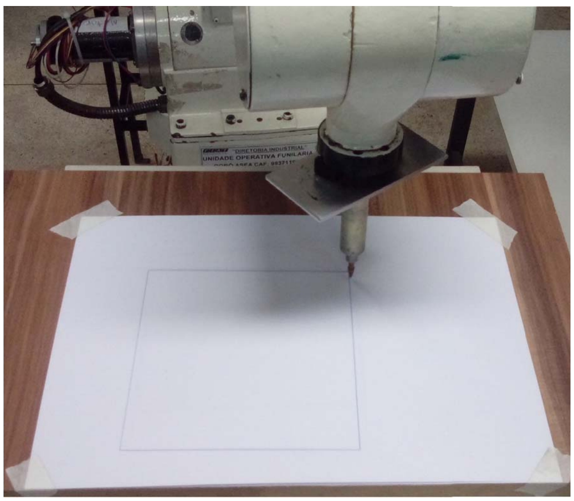

Fig. 6. Experiment for geometric error analysis in X/Y (table)

With the same proposed experiment (drawn square), other evaluated parameters over robot performance after retrofitting process was the perpendicularity between two drawn lines by the tool (fixed pen), obtaining a approximately error of 3 % in comparison with 90° degrees as ideal value in that kind of experiments. The obtained error value is a reference to reduce the perpendicularity error with specialized measurement instruments.

C. Reliability, accuracy and new functionalities for ASEA Robot - MatLab/Mach3

After a retrofitting process the ASEA robot simulated 250 weld bead displacement in 8 hours. Each bead had 100 mm and the velocity was 200 mm/min with a approximately repeatability error of 0.5 mm [2].

The robot now does not have a closed architecture. In this way, all of its hardware and software components can be replaced and new systems and control architectures can be analyzed and validated, with their immeasurable advantages for research projects. Thus is possible to implement new functionalities like machine vision and communications protocol, for instance and the control cabinet can be used in robotic mechanism with up to 6 DOF.

During experimental phase, the ASEA Robot showed excellent accuracy of trajectories in execution and implementation of weld beads by FCAW (Flux Cored Arc Welding) process. Performing a cost analysis, it was observed that retrofitting represents only 5% of a new robot value with similar characteristics to market price of 55,000.00 dollars. Thus, the re-manufacturing process provides a benefit/cost ratio very high.

VIII. COMPARATIVE ANALYSIS AMONG THE LINUXCNC AND MACH3/MATLAB

The LinuxCNC controller does not require the acquisition of licenses due to the fact that the Linux operating system (OS) has open source license. That makes the platform very attractive for industrial applications where operating costs must be low. Additional care should be given with the updates of the operating system, as they are not guaranteed due to the fact that they are carried out by communities that often do not receive help from development institutions. It is necessary to clarify that LinuxCNC has to be installed next to the Linux operating system (Ubuntu 10.04), so the capacity of the hard drive has to be 8GB.

Already using the Windows platform requires the acquisition of licenses for operating system, MatLab and also the numerical command controller Mach3. In this way, a high initial cost can be verified for application of Windows operating system for robotic manipulators control. What often mitigates these costs and the fact that academic centers already have the license of Windows and MatLab.

It should be noted that MatLab can be easily replaced by the GNU free license, version for Linux, in relation to matrix calculations involving transformation matrices. The Mach3 software has a free license for up to 1000 lines of code, or it can also be replaced by micro-controllers dedicated to the task of generating pulses, for example.

Table III is a proposed decision table that compares the two solutions (LinuxCNC vs. Mach3/MatLab) using criteria of comparison and a degree of adherence to the criterion (0 to 5). When the platform has zero (0) is because it doesn't support the criteria requirements, or it needs an integration with a additional system. Three points (3) when exists one same feature with the evaluated criteria, but must have a specific setup into platform environment. Finally if the evaluated criteria has five points (5) is when the controller totally support the criteria according to platform features.

After comparing the criteria it is noted that the solution based in LinuxCNC with integrated kinematics to CNC based controller itself is more robust and efficient solution, since the environment is fully integrated with the RS-274 standard. This platform can be used to control the ASEA Robot position without any editing; meanwhile LinuxCNC itself calculates direct and inverse kinematics in real time due to RTAI kernel. It is necessary to clarify that proposed criteria is only to compare two solutions [2], [3], because original robot performance criteria are not available in literature .

TABLE III

USED NC ARCHITECTURE SOLUTIONS COMPARISON ([1], [2])

Criteria	LinuxCNC	Mach3 & MATLAB
Kinematic integrated to platform	5	3
G and M Code standard RS274	5	5
Offline programing with CAD/CAM	5	0
Programming by teaching	3	0
Low cost hardware	5	5
Open source software	5	0
MDI, manual and automatic	5	5
Integration with TCP/IP	5	5
Remote access	5	5
OS in Real time	5	0
Robustness and Integration	5	3
Usability	5	3
Maintainability	5	5
Available Documentation	3	5
Repeatability (Error)	5 (.3 mm)	3 (.5 mm)
STEP-NC Compatibility	3	0
Total Points (Ranking)	74/80 (1)	47/80 (2)

IX. CONCLUSIONS

To allow the implementation of the robotic manipulators remanufacturing, or even for construction of new control systems, trending will be use open architectures controllers (OAC) in order to create non-dedicated systems that make operation and maintenance costs more accessible.

The hardware requirements for the implementation of proposed solutions are considerable variable when choosing the software platform to be used, as well as the costs according to the characteristics of the computer where the operating systems and the applications of each of the solutions will be installed. According to what has been described, LinuxCNC becomes an interesting solution because it needs few hardware resources in comparison with Mach3/MatLab alternative, in addition the license does not has cost.

By the analysis of the present work, it can be concluded that the use of MatLab/Mach3 or LinuxCNC will basically depend on the application and response time required to perform the task. In academic activities the MatLab® software would be more interesting in what concerns the presentation of how and dynamics of which are the steps for the kinematic representation to manipulator control trajectory. Already, LinuxCNC is more interesting for industrial applications, where it is not necessary to acquire licenses or a complex systems less susceptible to failures or locking of the operating system, for example, besides being a complete environment with direct and inverse kinematics of the Robot ASEA integrated in the numerical control platform, controlling and calculating the orientation of each joint from a generic NC program created by a commercial CAD/CAM system.

Using the same hardware devices, LinuxCNC solution was better scaled, being a more robust and integrated solution than Mach3/MatLab alternative, also it is real-time based solution for immediate response time applications and reducing operating failures.

After LinuxCNC retrofitting the repeatability error was on the order of 0.3 mm, the straightness error was 0.4 mm and the perpendicularity error was 2.8 degrees [3].

The results and detailed progress of retrofitting technique, as technical documents of generated designs for control, power supply subsystems and selected devices for joint sets, allowing that ASEA robot comebacks to be operational, they were available to anyone who wants to contribute, use and develop new applications based in obtained results with present work, encouraging a dynamic consolidation of an Open Architecture Controller (OAC) for industrial robotic arms. The all related information of project is available at `https://tinyurl.com/yc3u7kkj` or `https://tinyurl.com/y7vro3yz`.

ACKNOWLEDGMENT

The authors would like to thank the financial support of the Brazilian government agencies MEC/CAPES/CNPq/FAPDF.

X. REFERENCES

[1] J. S. Toquica and J. A. Alvares, "Implementacion de la tecnica retrofitting para el robot asea irb6-s2 usando linuxcnc," in *Congresso Nacional de Engenharia Mecânica - CONEM2016*, 2016, p. 10.

[2] M. H. S. Bomfim, "Remanufatura de manipuladores robóticos com arquitetura aberta," Master's thesis, Jan. 2013.

[3] J. Toquica, "Retrofitting do robô asea irb6-s2 baseado em tecnologias de comando numérico usando linuxcnc," Master's thesis, Universidade de Brasilia, 2016.

[4] A. De Lima, L. Poubel, F. Lizarralde, A. C. Leite, and G. Gleizer, "Atulização de hardware e software de um robô industrial," *Congresso Brasileiro de Automática*, vol. XVIII, pp. 4403–4410, 2010.

[5] D. Milutinovic, "Reconfigurable robotic machining system controlled and programmed in a machine tool manner," *The International Journal of Advanced Manufacturing Technology*, vol. 53, pp. 1217–1229, April 2011.

[6] R. Preez, "3d 6-dof serial armn robot - kinematics and implementation in linuxcnc," *ASM*, 2014.

[7] E. Lima II *et al.*, "Sensoring for retrofitting of an industrial robot," *Information Control Problems in Manufacturing*, vol. 1, pp. 545–550, April 2004.

[8] D. Ž. Jokić and S. D. Lubura, "Comparative analysis of the controllers for puma 560 robot," *IFAC-PapersOnLine*, vol. 49, no. 25, pp. 98–103, 2016.

[9] G. K. Cheruiyot, X. Zhu, and Q. Cao, "Orocos-based generic control system for a 6 dof industrial manipulator," in *Advanced Robotics and its Social Impacts (ARSO), 2016 IEEE Workshop on*. IEEE, 2016, pp. 174–179.

[10] A. Barrientos, *Fundamentos de Robótica*, segunda ed. McGraw-Hill, 2007.

[11] T. Szkodny, "Forward and inverse kinematics of irb-6 manipulator," *Mechanism and machine theory*, vol. XXX, pp. 1039–1056, 1995.

[12] M. Gutierrez, *Desenvolvimiento de uma Fresadora CNC Aderente á Norma STEP-NC Baseado no Controlador de Máquina Avançado (EMC2)*. Universidade de Brasilia, 2013.

[13] T. Staroveski, D. Brezak, T. Udiljak, and D. Majetic, "Implementation of a linux-based cnc open control system," *International Scientific Conference on Production Engineering*, vol. XII, pp. 209–216, June 2009.

[14] J.-Y. Hascoet and M. Rauch, "Enabling advanced cnc programming with opennc controllers for hsm machines tools," *High Speed Machining*, vol. 2, no. 1, pp. 1–14, 2016.

[15] M. Glavonjic, D. Milutinovic, S. Zivanovic, Z. Dimic, and V. Kvrgic, "Desktop 3-axis parallel kinematic milling machine," *International Journal of Advanced Manufacturing Technology*, vol. XLVI, pp. 51–60, January 2010.

[16] P. Duysinx and M. Geradin, "An introduction to robotics: mechanical aspects," 2004.

[17] P. Abreu, "Robótica industrial: Especificação de robôs e células robotizadas," Slides, Universidade do Porto, 2002.

[18] L. C. P. L. Oliveira, "Maquinagem de superfícies complexas com recurso a sistema robótico," Master's thesis, Universidade do Porto, 2013.