SENSORING FOR RETROFITTING OF AN INDUSTRIAL ROBOT [1]

Eduardo José Lima II [*]
Guilherme Campelo Fortunato Torres [*]
Carlos Alberto Carvalho Castro [*]
Alexandre Queiroz Bracarense [*]
Renato Ventura Bayan Henriques [*]
Walter Fetter Lages [**]

[*] *Federal University of Minas Gerais*
School of Engineering, Mechanical Engineering Department
Av. Antnio Carlos, 6627 - 31270-901 Belo Horizonte, MG, Brazil
ejlima2@hotmail.com,
guilhermefortunato@hotmail.com,
carlosjapao@hotmail.com, bracarense@ufmg.br
bayan@demec.ufmg.br
[**] *Federal University of Rio Grande do Sul*
School of Engineering, Electrical Engineering Department
Av. Osvaldo Aranha, 103 - 90035-190 Porto Alegre, RS, Brazil
w.fetter@ieee.org

Abstract: Retrofitting of an industrial robot consists on verifying the usability conditions of each component of the robot, replacing obsolete ones, especially electronics and control. This work describes the replacement of the analog sensors (resolvers and tachometers) of a retrofitted ASEA IRB6 robot by digital sensors (incremental optical encoders). The advantages of such replacements are assessed through the use of tachometers and encoders for speed feedback. The control architecture based on a CAN-bus is also presented.

Keywords: Retrofitting, encoders, closed loop control, industrial robots, robot control

1. INTRODUCTION

The Laboratory of Robotics, Welding and Simulation (LRSS) from Federal University of Minas Gerais (UFMG) works currently in a project of retrofitting of an ASEA IRB6 robot, with 5 degrees of freedom, manufactured in 1977 (Figure 1).

Retrofitting of industrial robots consists on verifying the utility of each component of the robot and replacing what is obsolete, especially the electronics and control. The objective of retrofit an old robot is, besides bringing up to date its control technology, to make it available for academic

[1] Authors thank FAPEMIG, FAPERGS, CNPq, CAPES and MANET for financial support.

Fig. 1. ASEA IRB6 robot.

studies, in special studies about robotization of the welding process.

An academic initiative for generating an open standard for robot control is the OROCOS (Open RObot COntrol Software) project (OROCOS, 2002). This project is in its early stages and does not produced a working prototype yet. Also, it is more oriented towards the software architecture of the whole robotic system and does not properly address the supporting hardware architecture.

Open hardware and software architectures for robot control were defined by the PINO project (Yamasaki *et al.*, 2002), but since they were designed with small legged mobile robots in mind, they do not seems adequate for industrial manipulator robots.

There are also commercial robot controllers that can control robots from any manufacturer (URC, 2003). Such a controller eliminates the need to learn different robot programming languages. However, its architecture is not open. Also, its programming language `RobotScript` (Lapham, 1999) can be used for all robots supported by the controller. Nonetheless `RobotScript` is based on `VBScript`, a proprietary language tightly coupled to the Windows operating system. Besides Windows has not been designed with real-time control in mind, it is known for proprietary protocols that change from time to time. Hence it does not appear to be appropriate to serve as a basis for a standard open architecture for robot control.

Fortunately, the mechanics of industrial robots has not changed too much. The main differences from old to newer robots reside in the actuator power drives and controller, including the software. This fact enables us to upgrade old robots to current technology by retrofitting the robot controller.

The first step in robot retrofitting involved the maintenance of the mechanical and electric parts. The robot were disassembled and a verification of all the motors/axis and its electric contacts was made. It was verified that the mechanical structure of the robot was in good state, thus enabling us to reuse it. Power source and motors were in good state as well.

Originally, the ASEA IRB6 used resolvers for position feedback and tachometers for speed feedback to make the closed loop control. Resolvers and tachometers are analog sensors, and threrefore need an analog processing. In order to avoid analog processing, it was chosen to use incremental optical encoders in retrofitting for position feedback. Such sensors are made by light sources, light sensitive devices and a disc in which black and translucent divisions alternate (Jones and Flynn, 1993; Everett, 1995). See Figs. 2 and 3. When the disc turns, the light receivers generate square shaped waves with $\pm 90^{o}$ phase (what allows to determine the turning direction). The frequency of these waves is proportional to the turning speed, as shown in Figure 4.

Thus, using the encoder is possible to get the relative position and speed of the axle by counting signal pulses ocurring in each time interval.

Fig. 2. Incremental optical encoder.

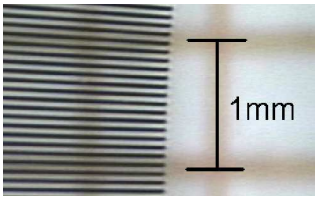

Fig. 3. Detail of an incremental optical encoder.

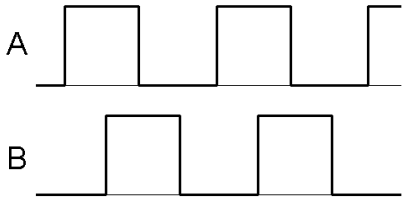

Fig. 4. Signals from channels A e B of an incremental optical encoder.

This work presents an experimental study comparing the use of tachometers and encoders for speed feedback. The hardware implementation of a distributed control system used for robot retrofitting is also described.

2. EXPERIMENTAL PROCEDURE

Before deciding for the replacement of the tachometers by encoders as speed sensors, experiments had been made to verify its accuracy. Basically, the experiment consisted in vary the power supplied to the DC motor and measure its speed using the encoder and the tachometer (Lima II *et al.*, 2003). The speed obtained with the encoder signal was considered in this work as a reference without error. The signals read from the encoder and the tachometer were processed by a digital oscilloscope. Figure 5 shows the experiment basic scheme and Figure 6 shows the adjustable power supply used to drive the motor.

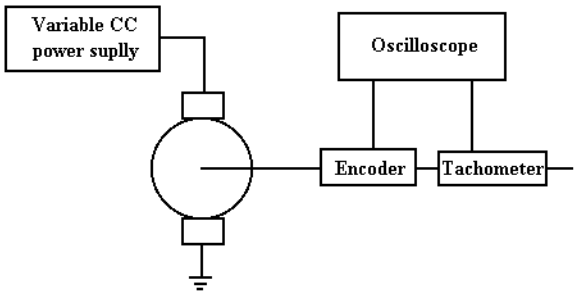

Fig. 5. Experimental setup diagram.

Fig. 6. Experimental setup.

The oscilloscope measures the frequency of the square wave with a precision of $0.1\mu s$ (Figure 7) Since the pulses number (N) of encoder wave is known, the speed of rotation of the motor can be calculated by:

$$\omega = 2\pi \frac{f}{N}$$

where f is the frequency.

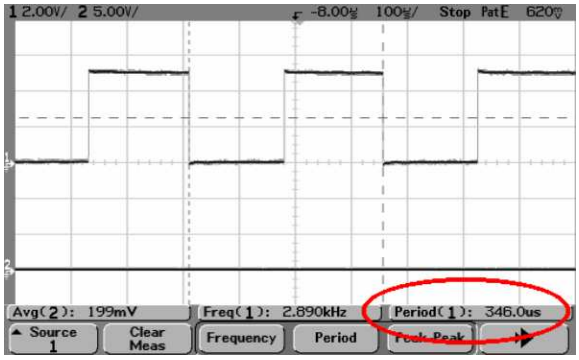

Fig. 7. Period as measured by the digital oscilloscope.

The tachometer voltage was also measured within the use of the oscilloscope. The oscilloscope shows the average voltage with a precision of 1mV (Figure 8).

Fig. 8. Average voltage given by the tachometer.

Then, considering the speed ω obtained by the encoder as a referential without error, a linear regression was made between the values of ω and the tachometer voltage (V_{tacho}). Since the motor speed is not exactly a constant with respect to the applied voltage (V_{motor}), it was chosen to use an average of three ω and V_{tacho} measurements for each V_{motor} value.

The coefficients of equation (1) were obtained by linear regression.

$$\omega = a_0 + a_1 V_{tacho} \tag{1}$$

By comparing equation (1) with the characteristic curve of the tachometer:

$$\omega = K V_{tacho}$$

it is expected a value for a_0 close to 0 and a_1 close to K.

The values for a_0 and a_1 were obtained by:

$$a_0 = \frac{\sum x_i \sum x_i y_i - \sum x_i^2 \sum y_i}{(\sum x_i)^2 - N \sum x_i^2}$$

$$a_1 = \frac{\sum x_i \sum y_i - N \sum x_i y_i}{(\sum x_i)^2 - N \sum x_i^2}$$

with x_i the i-th value of ω and y_i the i-th value of V_{tacho}. The correlation coefficient r was obtained by:

$$r = \sqrt{1 - \frac{S_{xy}^2}{S_y^2}}$$

where:

$$S_y^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \overline{y})^2$$

$$S_{xy}^2 = \frac{1}{N-2} \sum_{i=1}^{N} (y_i - y_{ci})^2$$

where $\overline{y}$ is the average value of V_{tacho} and $y_{ci} = a_0 + a_1 x_i$.

The obtained values were: $a_0 = 6.687$ and $a_1 = 31.706$. Figure 9 shows the comparison between the obtained curve ($\omega = 6.687 + 31.706 V_{tacho}$) and the experimental data. Value of $a_0 = 6.687$ is close to zero, as expected, considering the scale chosen and the ω variation up to 250 rad/s (a_0 is about 2.7% of 250).

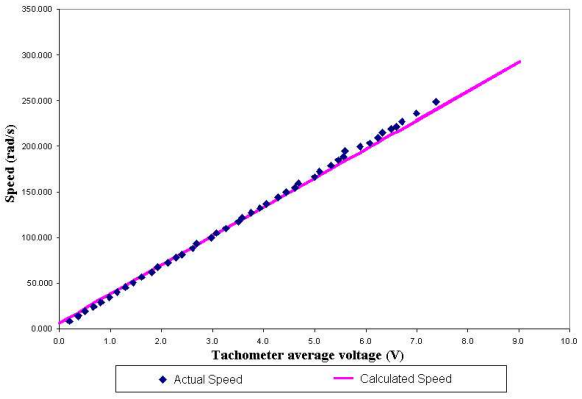

Fig. 9. Linear regression between values of ω and V_{tacho}.

It was found $r = 0.989570$. The factors that make the value of r to be far from 1.000000 are only the tachometer measurement errors, since the motor speed variations are the same measured by the encoder and the tachometer.

Therefore, it can be concluded that besides having the advantage of producing digital signals, that can be directly processed by the digital robot controller, encoders also offer higher precision in measuring speed and better noise immunity when compared to tachometers.

By using the same experimental data, the motor response curve in steady state without load was also obtained. A linear regression relating the voltage applied in the motor (V_{motor}) and the speed computed through the frequency of the encoder signal (ω) was obtained. The resulting values were: $a_0 = -24.731$ and $a_1 = 10.838$, hence:

$$\omega(V_{motor}) = -24.731 + 10.838 V_{motor}$$

Figure 10 shows the related curve. A value of $r = 0.999720$ was obtained, revealing an excellent linearity of the original robot motor, despite being 26 years old.

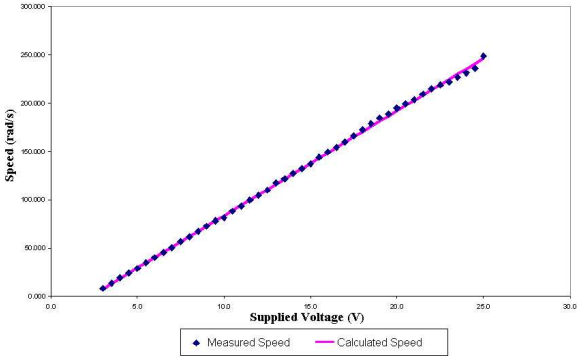

Fig. 10. Steady state response of the motor without load.

Another interesting function is $V_{motor}(\omega)$, which determines the voltage to be applied to the motor to in order to get the desired angular speed in steady state:

$$V_{motor}(\omega) = 2.282 + 0.092\omega \qquad (2)$$

Through equation (2) it can be observed that the motor has a dead zone voltage of 2.282V (below this voltage the motor does not move).

3. ENCODERS READING BY ROBOT CONTROLLER

After choosing the encoders and mounting them on the motors, it was necessary to develop a structure to receive and process the signals and send the feedback information to the controller. The original control architecture was analyzed and a new architecture was proposed.

Despite the independent control of each joint, the control was originally centered in only one machine (the robot controller) for each manipulator, bringing the problem of a single point of failure. By considering the possibility of redundant degrees of freedom that the robot can have in its

structure (for a given task), a considerable increment in the tolerance for global system failures would be reached by a distributed control. The devices to be arranged in a distributed architecture are those ones responsible for the I/O in each joint (i.e., position sensors, motors and brakes) and, eventually, processors with sufficient performance to implement control strategies.

In order to build a flexible, scalable and lower costing structure, a distributed I/O network based on microcontrollers and CAN-bus was designed (Lages and Bracarense, 2003) (Figure 11). There are also Ethernet connections, so that management tasks and programs can be uploaded to controllers by FTP.

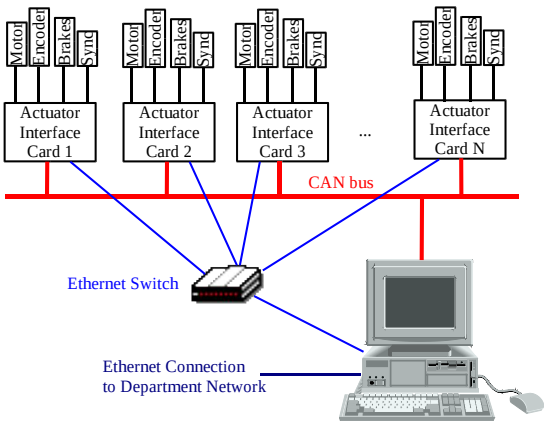

Fig. 11. Distributed I/O architecture over CAN-bus.

What makes CAN-bus an attractive base for the project of flexible real time communication protocols is the possibility of establishing global priorities for the nodes access to the bus, which is made through the following rules: (i) one of the states, 0 or 1, is assigned as dominant and the other as recessive, (ii) in the case of a collision, the node which writes a recessive bit in the bus and reads a dominant bit (i.e., a message with bigger priority than the one which is being transmitted) interrupts the transmission, preventing wasting band with posterior retransmissions of both nodes. This access method to the bus is called CSMA/CA, where CA means Collision Avoidance, although the arbitration mechanism would be better described by "collision solving", because the collision is not prevented, but solved.

The basic idea behind the use of a CAN-bus is simple: each electric device, sensor, actuator, or combination of these in the subsystem is connected to a small computerized module. This represents a total rupture in the concept of centered control connected to modules by a complex wiring system. The CAN protocol is based on simple modules interconnections by a serial data bus, offering a tremendous reduction on electric wiring in the robot.

In the case of ASEA robot, each joint controller can act as a local controller implementing, for instance, a simple PID control law (Fu *et al.*, 1987; Sciavicco and Siciliano, 1995) based on setting points received from the network or as an I/O processor, sending sensors readings and receiving actuators values through the network. When operating in I/O processor mode, the control loop is closed at the controlling PC, enabling the implementation of complex control laws.

Each joint controller is implemented by an custom designed Actuator Interface Card (AIC), implementing PWM amplifiers to drive a motor, an quadrature decoder to interface with optical encoder, a sync switch and brake interfaces. In figure reffig:TINI, it is shown: #1 TINI microcontroller board, which has a microcontroller 8051 with one Java interpreter and a CAN controller on-chip; #2 print circuit board, which has CAN, RS232C, Ethernet and transceivers connectors; #3 Ethernet connector; #4 CAN and RS232 connectors; #5 H bridge.

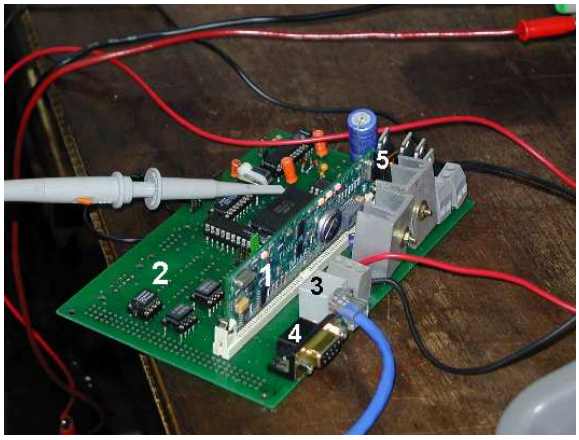

Fig. 12. Actuator interface card (AIC).

The main computer software is based on RTAI (Cloutier *et al.*, 2000; Dozio and Mantegazza, 2003), a real-time variant of the Linux operating system and communicates with the AICs. The control software is extensively based on the use of libraries written in C++, to define objects that represent each robot joint. The motors driving, the brakes releasing and the encoders reading are carried through functions defined in these libraries. Figure 13 shows some program lines which calls these functions.

```
aic.motor.on();
aic.brake.release();
aic.motor=10; //applies 10V to motor
cout<<"Encoder="<<aic.encoder.read();
```

Fig. 13. Examples of program lines which communicates with the AICs.

It is expected, when the project is concluded, an educational controller which may be used in

graduate and undergraduate robotic courses. The program will show in real time values of set-points and actual values of position, speed and tool acceleration and each robot joint through graphs. It will allow functioning characteristics learning and other control algorithms development.

Moreover, the controller, with opened architecture, will virtually allow the programming and execution of any robot trajectory, even complex ones, making possible more simplicity in the programming of trajectories.

4. CONCLUSIONS

Comparative tests has shown that encoders produces more accurate speed measurements than tachometers that originally gave information of angular speed to the robot controller.

The experimental data had been also used to obtain the steady state response curve of the original motors without load. The linearity of the curve stated that the motors still are in good functioning condition, enabling their use in the robot retrofitting process.

The distributed control architecture based in a CAN network, and how each node can locally close the control loop or just receive voltage values to be applied in the motors and to inform feedback values to the controller were shown.

The use of an open control architecture makes possible studies in robotics area and development of new control algorithms for the robot and real time welding parameters calculation, specially using covered electrode, by developping advanced control laws that adapt the control system behavior as a function of welding parameters.

REFERENCES

Cloutier, Pierre, Paolo Mantegazza, Steve Papacharalambous, Ian Soanes, Stuart Hughes and Karim Yaghmour (2000). DIAPM-RTAI position paper. In: *Proceedings of the Real-Time Operating Systems Workshop*. Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano. Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano. Milano, Italy. pp. 1–28. http://www.rtai.org.

Dozio, L. and Paulo Mantegazza (2003). Linux real time application interface (RTAI) in low cost high performance motion control. In: *Proceedings 0f the Motion Control 2003 Conference*. Associazione Nazionale Italiana per l'Automazione. Milano, Italy.

Everett, H. R. (1995). *Sensors for Mobile Robots: Theory and application*. A K Peters.

Fu, K. S., R. C. Gonzales and C. S. G. Lee (1987). *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw Hill College Div, Inc.

Jones, Joseph L. and Anita M. Flynn (1993). *Mobile Robots: Inspiration to Implementation*. A K Peters.

Lages, Walter Fetter and Alexandre Queiroz Bracarense (2003). Robot retrofitting: A perspective to small and medium size entreprizes. In: *Proceedings of the 3rd Austrian Brazilian Automation Day*. RECOPE/MANET. RECOPE/MANET. São Bernardo do Campo, SP, Brazil. pp. 1–11.

Lapham, John (1999). RobotScript the introduction of a universal robot programming language. *Industrial Robot* **26**, 17. http://www.rwt.com/main/articles/robotscript.html.

Lima II, Eduardo J., Carlos A. C. Castro and Roberto M. Andrade (2003). Levantamento da curva de resposta em regime permanente de um motor de corrente contnua e determinao da constante de um tacmetro. Trabalho da disciplina tcnicas da pesquisa experimental. Universidade Federal de Minas Gerais.

OROCOS (2002). Open robot control software. http://www.orocos.org.

Sciavicco, Lorenzo and Bruno Siciliano (1995). *Modeling and Control of Robot Manipulators*. McGraw Hill, Inc.

URC (2003). Universal robot controller. http://www.rwt.com/main/urc.html.

Yamasaki, F., K. Endo, M. Asada and H. Kitano (2002). Energy-efficient walking for a low-cost humanoid robot, PINO. *AI Magazine* **23**(1), 60–61.