Embedded Control System for a 5-DOF Manipulator by Means of SPI Bus

Angel Flores-Abad and Adrian Arpídez
Universidad Autónoma de Ciudad Juárez
Instituto de Ingeniería y Tecnología
anflores@uacj.mx, al68918@alumnos.uacj.mx

Abstract

This paper presents a SPI based embedded control system for a 5-DOF industrial robot manipulator. A set of microcontrollers is used to perform the control task; each one computes its own control signal to achieve joint movements. The embedded system has an 8-bit microcontroller master-slave scheme, with one master driving 5 slaves, in order to exchange control information by means of SPI network. Functionality tests are carried out by implementing a digital PID control algorithm on a Scorbot robot and results are presented. The developed embedded system provides a low cost, flexible and high effective solution for implementation of control algorithms.

1. Introduction

Robot manipulators are very used in industry to move and position materials or products. The robot manufacturers generally also provide the control system, which is programmed with some specific language depending on the brand. These controllers are expensive due to the characteristics of reliability, precision and safety that they have to satisfy. Besides, most of times, the control algorithm is not wholly inside the electronic devices, but also in the computer. Hence, in spite of the possibility of use a hand-held robot control, the necessity of a computer is inherent.

An embedded system has programmable hardware, with embedded software as one of its more important components. Microcontrollers and digital signal processors (DSPs) are the main engines of the deeply embedded development world. Both of them are wide spread in many industrial control systems [1]. Embedded systems are classified in small, medium and sophisticated scale regarding its application. The small scale embedded systems are designed with a single 8- or 16-bit microcontroller; they have little hardware and software complexities and involve board-level design. They are low-power consumption and may even be battery operated. When developing embedded software, it has to fit within the available memory and keep in view the need to limit power dissipation when the system is running continuously [2]. Those microcontroller systems generally are provided with built-in modules to help to reduce the amount of external hardware, such modules usually are: PWM, ADC, serial communication (SPI, I2C, USART), eeprom memory, counters and timer modules.

A serial communication offers big advantages of circuitry size reduction, several serials protocols exists to achieve information exchange, such as I2C, SPI, CAN, LIN, and USB. Unlike a standard serial port, SPI is a synchronous protocol in which all transmissions are referred to a common clock, generated by the master (processor). The receiving peripheral (slave) uses the clock to synchronize its acquisition of the serial bit stream. Many chips may be connected to the same SPI interface of a master. A master selects a slave to receive by asserting the slave's chip select input [3].

With advances in digital technology, the science of automatic control now offers a wide spectrum of choices for control schemes. But the PID controller is by far the most dominating form of feedback in use today. More than 90% of all industrial controllers are PID. It is surprising how much can be achieved with such a simple strategy. It is the first solution that should be tried when deal with control [4].

In the case of robots the control system has to deal with 4, 5, 6 or more joints, the control of which cannot be done with a single controller. If we would like to use a microcontroller for robot control, each joint should be controlled with one local unit. However in the case of the robot control algorithms, an uncoupled control method could be used, in this case, the influence of the other joints over the controlled one, are represented as a perturbation, those the control law

978-0-7695-3799-3/09 $26.00 © 2009 IEEE
DOI 10.1109/CERMA.2009.62

for the ith joint may depend on the other joint variables [5]. Hence it is necessary that in every control period the joint information (measured position) to be exchanged between the master and slave controllers. Accordingly fast communication between the local controllers is necessary. It can be done by organizing the controllers in a distributed system around a fast communication bus, which can guarantee that the joint information is exchanged between the local units during a single control period.

1.1 Related Work

Distributed control systems permits to develop parallel and multitasks schemes. They have been used in power systems, autonomous vehicles, manufacturing processes and other applications where several controllers need to cooperate with access to different information and with bounds on the communication between them. The work presented in [6] is related with a distributed control for a dual arm mobile robot, the architecture is based on servers and multiple clients in different computers. The study in [7] deals with the available software frameworks for distributed control systems/robotics and describe a novel agent-based software architecture that helps in the development, testing, and deployment of distributed controllers, results show that the systems are capable to synchronize an in door group of robots. In [8] a DSPs based control system is developed to control a 4-DOF manipulator, each controller calculates the control signal for each joint and the trajectory planning is implemented in a computer.

The main advantage of embedded control system proposed in this work is that it can achieve good real-time and control performances with only 8 bit microcontrollers and no control and calculation tasks are in the computer, all of them are computed inside the microcontrollers.

The outline of the paper is as follows. Section 2 presents a detailed description of the embedded system architecture. In section 3 experimental results are shown with the developed embedded control system for trajectory tracking control task. Finally section 4 gives some conclusions and further works.

2. Embedded Controller System

Considering the positioning of only one joint, characterized by the polar moment of inertia J and a viscous damping coefficient k_b, where a torque τ is applied, as shown in figure 1.

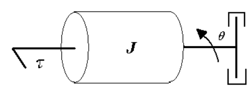

Figure 1. Rotational joint.

τ and the angle θ are related by

$$\tau = I\dot{\theta} + b\ddot{\theta} \tag{1}$$

If the torque τ is generated by a DC motor, a classical linear armature-controlled could be used as given in [9].

$$J\ddot{q} + \left[f_v + \frac{k_a k_b}{R_a} \right] \dot{q} = \frac{k_a}{R_a} v \tag{2}$$

Where

q : output angular position of the joint
v : input voltaje
J : inertia of the joint
f_v : viscous friciton

The constant k_a, k_b and R_a are electrical characteristics of the motor. All these parameters are strictly positive constants. The position regulation of each joint consist of ensuring

$$\lim_{t \to \infty} q(t) = q_d \tag{3}$$

Where q_d is the desired angular position of the joint. The basic structure of the PID control law driven by the joint position error defined as $\tilde{q} = q_d - q$ is given by

$$u = \left[k_p^{'} + \frac{k_i^{'}}{p} + k_d^{'} p \right] \tilde{q} = k_p^{'} + k_i^{'} + \int_0^t \tilde{q}(\sigma) d\sigma + k_d^{'} \dot{\tilde{q}} \tag{4}$$

Where, $p = d/dt$ is the differential operand, $k_p^{'}$, $k_i^{'}$

and $k_d^{'}$ are the proportional, integral, and derivative gains, respectively.

PID control (4) can be implemented as depicted in figure 2, where p_i is the perturbation on the ith joint due to the influence of the other joints. In this work an uncoupled PID feed-back position error were implemented for each articulation, as shown in fig. 3. Generally all the joint variables and the prescribed trajectory of each joint should be known to determine the ith control signal u. If the control law is implemented separately for each joint (on local controllers), before the implementation of the control law the ith controller should receive the joint information and prescribed trajectory of all the other joints. However for some specific control tasks and robot architectures the ith control signal will not depend on all of the joint variables and prescribed trajectories of all of the joints.

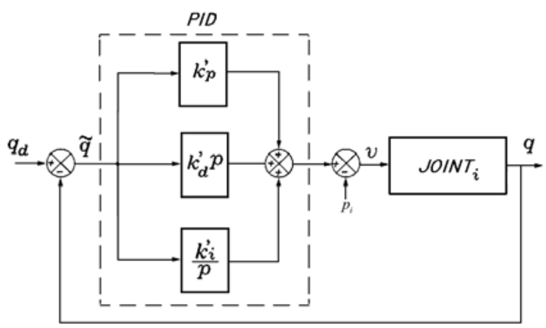

Fig. 2. PID control based on feed-back of posit ion error.

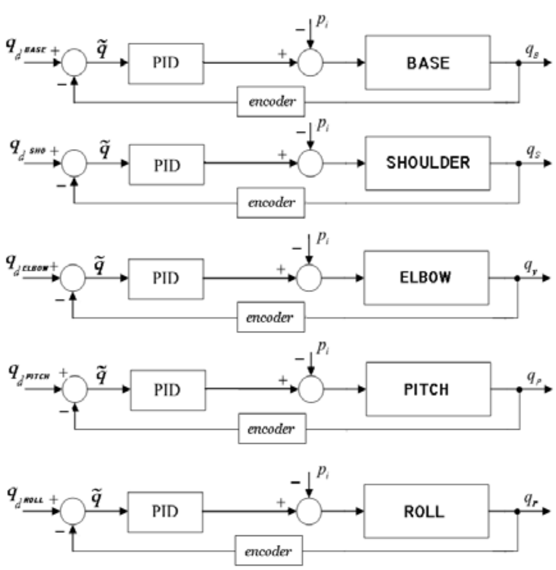

Fig. 3. Control scheme for the 5-DOF robot.

2.1. Embedded system.

Distributed control demands task division to get common objective. A set of 6 microcontrollers, one master (μ Core) and 5 slaves (SubCores) are arranged as shown in figure 4 to control the 5-DOF manipulator.

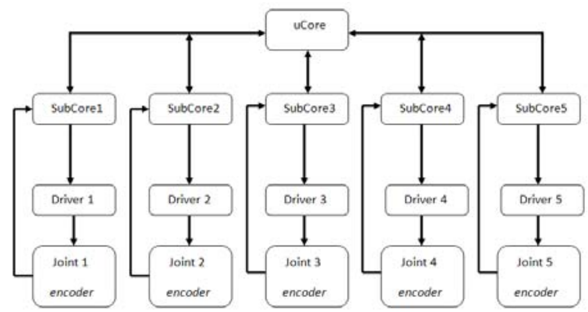

Fig. 4 Block diagram of comunication using a bidirectional SPI Network.

As shown in figure 4, the μCore microcontroller receives and interprets the user instructions, then it sends the information to the corresponding SubCores, where the trajectory and control algorithm are computed. In order to be identified for the μCore, each SubCore will adopt a hardware-ID configuration. This ID will auto assigned a firmware to each SubCore.

It is necessary to establish an own protocol, regarding 16-bytes string that strictly needs to be sent, notwithstanding SPI offer us not limits to expand our protocol, that is one of the reasons why SPI were chose regarding to I2C. The message protocol adopts the form: 1,00,X, 5,-500. Which means, SubCore **1**, executes instruction **0**, modify the **X** position of the profile **5** in **500** encoder counts. Each Subcore sends the control signal u to the drivers for giving the necessary PWM voltage to the DC motors, which causes the movement of the corresponding joint at the desire position q_d. Now the μCore and SubCore routines will be briefly described.

2.1.1 µCore

void setup(void) // initializes the register modules for communication (USART, SPI,GPIO).
void Checksys(void) // Analyzes the entire system communication, in order to prevent failures on the communication.
void loader(void) // prompts a graphical display using RS232.

void PutUSART(void) // communicates the µcore to the external computer.
void PutSPI(void) // sends the information string to the subcores.
void Changestateslaves(void) // safety mechanism to communicate the adequate subcore.
void Openloopctrl(void) // drives the system on open loop control.
void main(void) // main function.

2.1.2 SubCore

void servoisr(void) // interrupt service routine, the vector is called every 50 µs.
void UpdTrajectory(void) // updates the trajectories that shall be follow.
void UpdPosition(void) // updates the position over all the trajectory path.
void CalcError(void) // performs the operation to calculate the actual error.
void CalcPID(void) // computes the PID algorithm given the values of the previously 3 function.
void setup(void) // initialize the register modules (PWM, SPI,GPIO).
void CheckSPI(void) // attends the µcore call interrupt
void GetSPI(U8 rdptr) // get the parameters given by µcore.
void GetID(void) // adopts an ID configuration given by a hardware module.
void ProcessSPIstring(void) // interprets the message received from µcore.
void main(void) // main function

3. Experimental results

The embedded control system developed was tested in the Scorbot-ER VII arm (figure 5), which is a 5 degree of freedom robotic arm designed for academic and industrial purposes [10]. The base/shoulder/elbow have standard industrial anthropomorphic manipulator configuration. The wrist mechanism is a two degree of freedom (Y-Z co-located axis) design. The gearing system is timing belt gear reduction followed by a harmonic drive gear driven by brush-commutated permanent magnet motors rated to 18A at 24VDC. This configuration ensures the system is relatively safe for operation outside a safety enclosure.

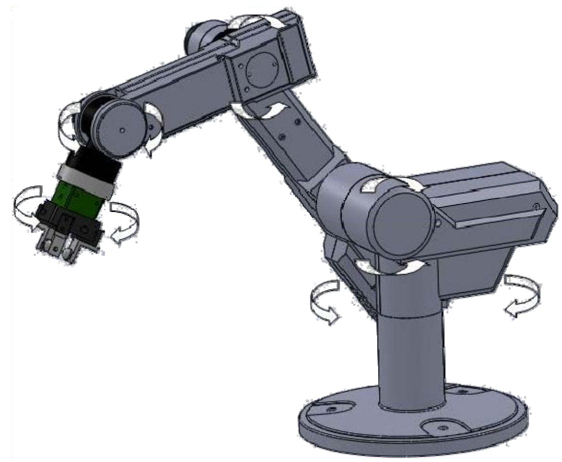

Fig. 5. CAD design of the Scorbot-ER VII.

Accordingly, to control the robot with the proposed distributed control system, one master and five slave controllers are necessary. The Microcontrollers used as slave and master processors are from the PIC18 family with 20 MHz clock frequency. All controllers are connected to a common SPI bus with 1.25 MHz clock frequency.

The encoders are connected to the encoder interface of the controllers through a low pass signal conditioning circuit. The counting of the encoder pulses are performed inside the controller. The control signal is sent out through the PWM interface of the slave controller and drives a L6203 Dmos H bridge amplifier. The motors are connected directly to the H bridge drive.

For better path performance and avoid overshoots, acceleration and deceleration should be implemented at the beginning and the end of the of the motion trajectory. One widely used profile of control that met these characteristics is the LSPB (Linear Segment Parabolic Blends) signal as shown in figure 6. The LSPB begins in the initial position q_0 increasing the speed from $t_0 - t_b$ as a quadratic polynomial and then continues with a line until $t_f - t_b$, finally speed goes down to the final position q_f.

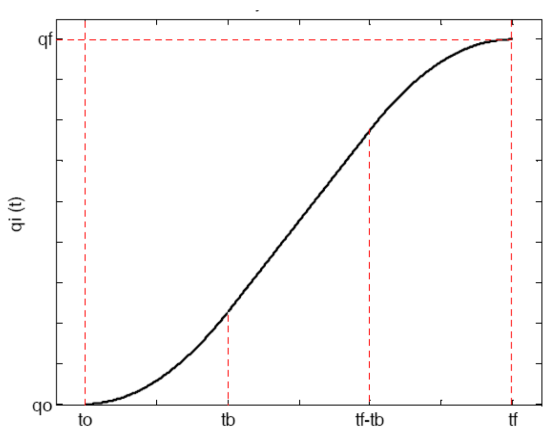

Fig. 6. LSPB signal as a soft motion trajectory.

In figure 7 the Scorbot with the embedded control system is depicted. In order to have a more economical system, the same case box of the original control system is used but, the number and size of electronic board were reduced as well as the number of electronic components in around 20% and as a consequence power consumption is also improved.

Fig. 7. Scorbot robot with the developed embe dded control system.

In figure 8 the driver, µcore and one SubCore board are shown. Once the electronic design was done and well tested, the communication/control boards were manufactured by the company PCBCART. The welding process was done manually.

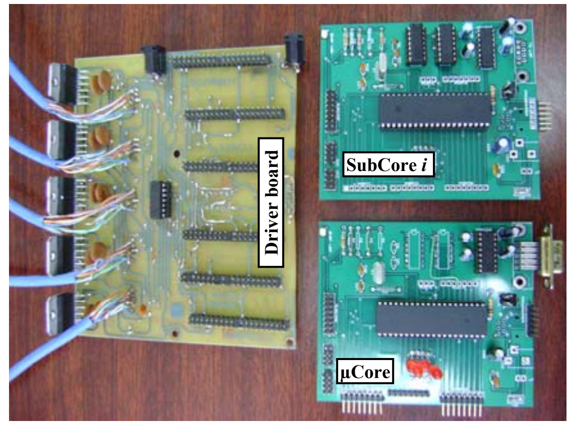

Fig. 8. Driver, µCore and one SubCore boards.

3.1 Communication measurements

During communication, µCore receives the position reference from a remote computer for all joints. Then it distributed the information to the corresponding SubCores. In figure 9 the 16-byte Synchronous strings are shown and figure 10 depicts a zoom to a single byte. The codified message is shown in figure 11, which shall match with the previous signal. Frequency depends on software configuration, from 312.5 KHz to 1.25 MHz.

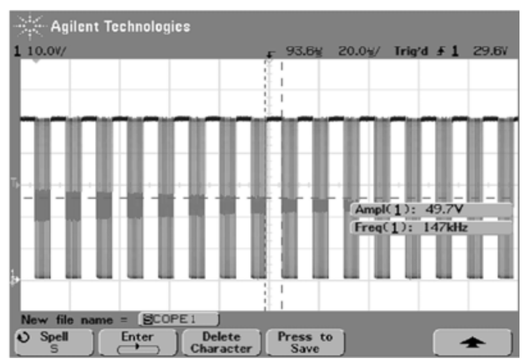

Fig. 9. 16-byte synchronous string.

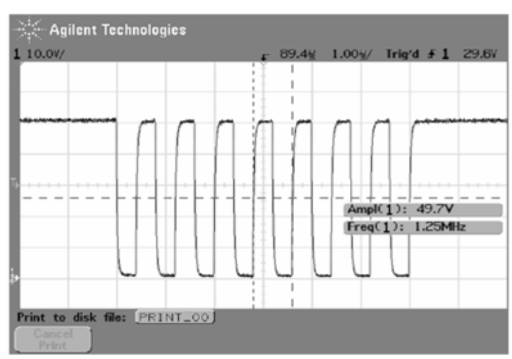

Fig. 10. Single byte synchronous string.

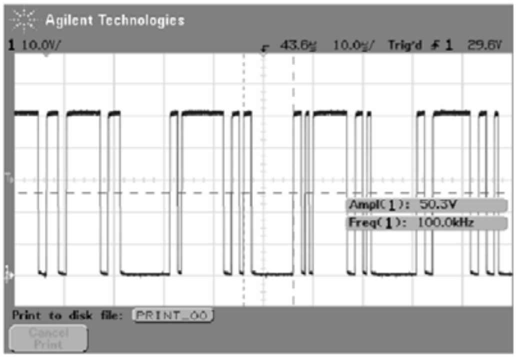

Fig. 11. Codified message.

3.2 Control measurements

A PID control algorithm was implemented on the developed embedded system. The algorithms were tested for trajectory tracking tasks: the reference signal is time varying LSPB. This was planed in such way that the motion of the robot to be time optimal with respect to a given maximum motor velocity and acceleration. In figure 12 the position reference signal and the real velocity are depicted. It could be seen that the control algorithm permits the correct tracking of the trajectory.

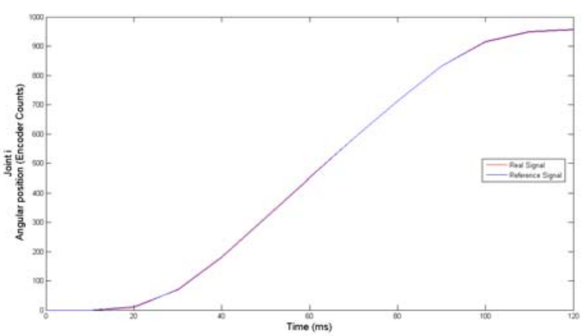

Fig. 12. Reference and real position signal.

4. Conclusions

A distributed embedded control system was presented in this paper. Low scale embedded systems with 8-bit microcontrollers demonstrated to be enough for control algorithms and reduce cost, power consumption and space. An uncoupled PID control law was used in each joint to achieve the position control task; it is considered that the others join affect only as perturbations. SPI bus has adequate speed and transference characteristics for distributed control applications. Experimental results show that real time position control systems for robot manipulator could be implemented in the embedded system developed with excellent results. An open platform for programming control algorithms is useful for academic as well as for research purposes. In the future a more sophisticated control scheme could be implemented in the developed embedded system.

5. References

[1] H., Dimitrios, Levine, William S., *Handbook of networked and embedded control systems*, Springer, Boston, 2005.

[2] R. Kamal, *Embedded systems: architecture, programming and design*, McGrawHill, USA, 2008.

[3] J. Catsoulis, *Designing embedded hardware*, O'Reilly, 2008.

[4] K.H. Ang, G. Chong and Y. Li, "PID Control System Analysis, Design and Technology", *IEEE Transactions on Control Systems Technology*. Vol. 13, No. 4 July 2005.

[5] A. Ollero*, Robótica, Manipuladores y Robots Mόvile*s, Alfa Omega-Marcombo, España, 2007.

[6] C. Qixin, Z. Zhen and G. Jiajum, A distributed Control and Simulation System for Dual Arm Mobile Robot. *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, JackSonville, FL, USA, June-20-23, 2007.

[7] H. Jia, W. Zhuang, Y, Bai, P. Fan and Q. Huang, "The Distributed Control System of a Light Space Manipulator", Proceedings of the IEEE International Conference on Mechatronics and Automation, August 5-8, Harbin, China, 2007.

[8] L. Márton, "Distributed control architecture for advanced robot control" *Proceedings of the international symposium on industrial electronics*, June 30-july 2 2008, Cambridge, UK, 2008.

[9] R. Kelly and Javier Moreno, "Learning PID Structures in an Introductory Course of Automatic Control" *IEEE Transactions on Education*, Vol. 44, No. 4, November 2001.

[10] Eshed Robotec, *Scorbot-ER VII User Manual*. USA, 1991.