Intelligent Servo Drives Control Based on a Single Fieldbus Network

Dan Puiu and Florin Moldoveanu

Abstract — **Due to the quick evolution of manufacturing processes, the demand for more flexible automation systems is on the rise. To answer these requirements, distributed motion control architecture based on intelligent drives tends more and more to replace the traditional solutions. This paper presents the control of an articulated arm robot with two local intelligent servo drives connected on a CAN network to a motion controller which receives the trajectory of the robot from a computer. The control structure is based on a single CAN network where local intelligent servo drives, a motion controller and a computer are connected.**

Keywords — **articulated arm robot, CAN network, intelligent servo drives, distributed control system, motion coordinator.**

I. Introduction

MANUFACTURING companies at the beginning of the 21st century face frequent, unpredictable market changes. These changes include frequent, rapid introduction of new products, changes in process technology, changes in product demand and mix. To stay competitive, manufacturing companies must possess a new type of manufacturing system that is responsive to all these market changes. This calls for reliable and mobile production units that can be easily rearranged in order to cope with unpredictable and frequent market changes [6], [8], [9].

This trend has driven automation users to require flexible automation systems that can be easily modified or upgraded in order to sustain a long term competitive position. In response to these demands, distributed motion control architecture tends more and more to replace the traditional centralized control architecture based around a single host controller such as a computerized numerical control or a computer motion control board. Rather than including all the control tasks in the central controller, the distributed control architecture is based on fieldbusses communication and Digital Signal Processing technology so that decentralized control tasks can reside in the intelligent servo drives [1], [7], [10].

A distributed control solution not only implies the spreading of intelligent modules on the system, it needs a set of instruments and specific mechanisms which facilitate the efficient exploitation of the distributed resources of the system [6]. These instruments must allow the engineer to design a distributed control system without taking care of distributed resources and of the problems generated by communication protocols. In this way he can concentrate only on specific automation control problems [3].

The benefits of a distributed control structure are: obtaining a more flexible structure that can be extended and which is more reliable and the reduction of installation and maintenance cost [1], [4].

The standardization of communication protocols allows connecting of different control modules made by different manufacturers on the same network [1], [4].

From robotics point of view, a distributed control solution requires a configuration that ensures good multi-axis control performances. In this way a distributed motion control designer should concentrate on choosing a network that ensures robust traffic and on the control rules of actuators drives [2], [11].

This paper presents a distributed control system for an articulated arm robot with four degrees of freedom. The robot has five servos, four used for actuating the axis and one to open and close the gripper of the robot. Servos are connected to two local intelligent drives which are commanded and synchronized with a motion controller. The synchronization of the two drives is important because the robot has to pass a certain trajectory. The motion controller receives the trajectory of the robot from a host computer, generates the appropriate command for each servo and then sends data to the local intelligent drives which execute the commands. The local intelligent servo drives, the motion controller and the host computer use the same CAN network to communicate.

II. Distributed Control Architectures

The first industrial distributed systems are based on OSI standard for network protocols. It resulted in a hierarchical architecture which has its components grouped on four layers [2] (see Fig. 1 (a)).

Each layer communicates with the upper one, from which it receives commands and returns its current state and it also communicates with the lower one, to which it sends commands and monitors it. Network traffic is performed vertically from one level to another (with the possibility of changing the communication protocol), but it is also performed horizontally between the entities of the same layer [2], [3].

The highest layer is the management or the supervision and monitoring one, where real-time tasks do not run. The

Dan Puiu is a Ph. D. Student at *Transilvania* University of Brasov, Romania, System Engineering Department (phone: +40740962037; e-mail: puiudan@unitbv.ro).

Florin Moldoveanu is professor at Transilvania University of Brasov, Faculty of Electrical Engineering and Computer Sience, Automaton Department (e-mail: moldof@unitbv.ro).

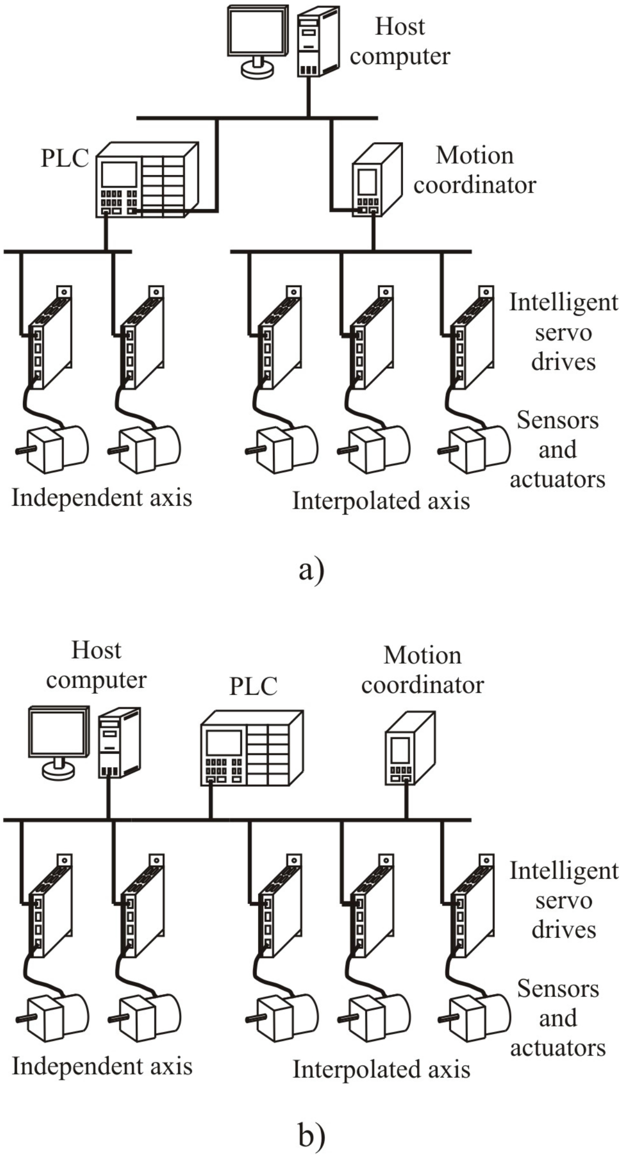


Fig. 1. Distributed control architectures [2]: (a) hierarchical architecture based on layers; (b) architecture with only one communication network.

main services of this layer are: providing user-friendly graphical interfaces; loading and unloading of programs or parameters and alerting the user of errors occurring during operations [2], [3]. Also, on this level it is possible to perform the analysis of working time, inventory and production monitoring. For all these tasks the access time is not a critical parameter and therefore the industrial communication network can be Ethernet.

The next layer is the process control one, where the execution times are a critical aspect. The tasks carried out by digital signal processors (DSP) or by programmable logic controllers (PLC) are numerous: trajectory generation; distribution of position reference for each axis; monitoring and interpretation of data acquired from a lower layer and management of entire network [2], [3]. Communication protocol must be reliable and allow quick and error free transmission, even conditions of electromagnetic disturbance. The most appropriate industrial communication networks for this layer are CAN and PROFIBUS.

The PLC or the microcontrollers from the axis control layer (the third layer) receive motion trajectory for the actuator and execute it. The last layer contains sensors and actuators and it is close to automation hardware.

This structure has a few disadvantages that make it hard to be implemented [6]:

- it has several communication protocols which increase the price of the automation and make it hard to debug;
- from the top of the structure to the bottom and reverse data has to pass through several processors and the speed of transfer is decreased;
- the combination between the horizontal traffic with the vertical traffic is hard to be implemented for real time multi-axis control.

An optimal solution to implement a distributed real time system is to connect, in a network with a single protocol, all the intelligent units from all the layers [10] (see Fig. 1 (b)). Even if all the devices are connected to the same industrial communication network, they maintain their functionalities from the distributed control architecture based on layers. Because the intelligent devices from the process control layer require a reliable, quick and error free transmission, the industrial communication network used must ensure these constraints, even if, for the intelligent devices from the first layer, the transmission time is not a critical factor.

This approach has, comparing it with the distributed control system based on layers, the following advantages:

- only one communication protocol used in the system;
- a simple structure for devices from the process control layer (from hardware point of view because they need only one communication interface and from software point of view because they implement only one communication protocol);
- the host computer can directly monitor the intelligent devices from the third layer.

It is useful for the host computer to monitor the conversation between local intelligent drives and motion controller because in this way it can directly determine the state of local drives.

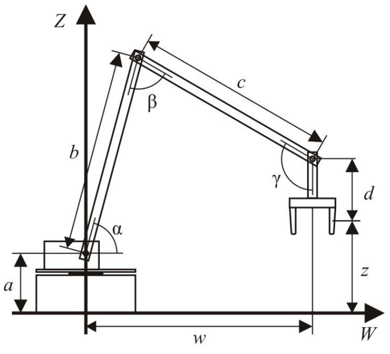
The main disadvantage of a distributed control system with only one communication network is that it increases traffic from the industrial communication network, which has to ensure a reliable, quick and error free transmission.

III. Geometric Model of Articulated Arm Robot

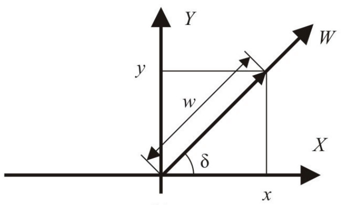
In Fig. 2 is presented the model of the robot, which is actuated by five servos. Because the servos have their own control loops, including the position control loop, the distributed control structure has only to generate the appropriate position trajectory for each servo.

One servo controls the rotation of the articulated arm in X-Y plane, which is the same with the plane where the robot is placed. Three servos move the arm of the robot in the plane Z-W and the last servo is used to control the position of the gripper. The W axis represents the axis from the X-Y plane where the first servo positions the articulated arm.

To simplify the model, it will be considered that the



(a)



(b)

Fig. 2. Robots model: (a) Arm plan view. (b) Rotation base plan view.

gripper of articulated arm will always be perpendicular on the surface of the X-Y plane.

From figure 2 (a) and (b) the direct geometric model of the articulated arm results easily:

$$\begin{cases} x = w\cos\delta \\ y = w\sin\delta \\ z = a + b\sin\alpha - c\cos(\alpha+\beta-90) - d \end{cases} \quad (1)$$

where x, y and z represents the position of the gripper, a, b, c and d are the dimensions of robots arms, α, β and γ are the angles between robots arms, δ is the angle of robot arm from the X-Y plane and w is the distance from the base of the robot and the projection of the gripper in the X-Y plane, which can be determined from figure 2 (a), resulting in the following equation:

$$w = b\cos\alpha + c\sin(\alpha+\beta-90). \quad (2)$$

Solving the system of equations consisting of equations 1, 2 and:

$$w = \sqrt{x^2 + y^2} \quad (3)$$

results in the inverse geometric model:

$$\begin{cases} \alpha = \arccos\left(\dfrac{w(b - c\cos\beta) - c(z - a + d)\sin\beta}{(z-a+d)^2 + w^2}\right) \\ \beta = \arccos\left(\dfrac{b^2 + c^2 - w^2 - (z-a+d)^2}{2bc}\right) \\ \gamma = 180 - (\alpha+\beta) \\ \delta = \arctan\left(\dfrac{y}{x}\right). \end{cases} \quad (4)$$

It can be observed that α depends on β. It is preferred to use this equation, because it is simpler, otherwise the second equation should be inserted in the first one and this will result in a very complex one.

IV. Control structure of robot

The intelligence of the control structure, which is distributed in the system, is composed of three microcontroller developing boards Dice-Kit, from Fujitsu. The boards have their on CAN and serial controllers.

The structure of the system is presented in Fig. 3. The computer, witch is the host computer from Fig. 1, creates the link between the user and the system, because here the user can insert the trajectory of the robotic arm. It transforms the trajectory into an array of segments of line and sends them to the Dice-Kit 1 board (see Fig. 3).

The Dice-Kit 1 board has the role of the motion coordinator from figure 1 and after it receives the array of segments of lines, it sends the appropriate commands to the Dice-Kit 2 and 3 boards which control the five servos of the robot. The last two boards represent local intelligent drives.

For a complete distributed control system, it is mandatory to use five local intelligent drives one for each servo of the robot, but, because this is only a prototype and the robotic arm is a small one, only two will be used. The Dice-Kit 2 board will control the angles a and δ of the robot and the Dice-Kit 3 will control the angles β and γ, plus the position of the gripper.

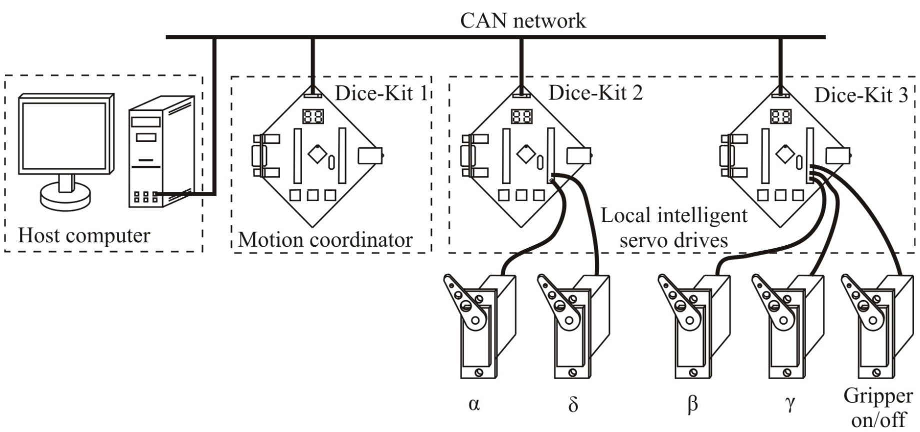


Fig. 3. The distributed control system for the articulated arm robot.

A. The Motion Coordinator

Based on the information received from the host computer, the motion coordinator will generate the commands for each local intelligent drive.

If the motion coordinator needs to move the gripper of the robot from the current point (x_0, y_0, z_0) to (x_1, y_1, z_1) from space, it has to send three broadcast messages containing the x_1, y_1 and z_1 values and then to send a broadcast message with the time in which the gripper should move from a current point to a destination point. The variation of the motion time determines the variation of the speed of the articulated arm robot.

Because the articulated arm robot has speed limitations, the motion coordinator must know how to calculate the minimum time necessary for the move. It can determine this time with the following equation:

$$t_{\min} = \max\left(\left|\alpha_0 - \alpha_1\right|, \left|\beta_0 - \beta_1\right|, \left|\gamma_0 - \gamma_1\right|, \left|\delta_0 - \delta_1\right|\right) \frac{\tau}{90} \qquad (5)$$

where α_0, β_0, γ_0, δ_0 and α_1, β_1, γ_1, x_1 are the angles between robot axes when the gripper is at the point (x_0, y_0, z_0), respective (x_1, y_1, z_1) and τ is the time in which a servo changes its position with 90 degrees at full speed.

B. Local Intelligent Drives

After the local intelligent drives receive the new set of coordinates (x_1, y_1, z_1), they have a break of 10ms and during this time the Dice-Kit 2 microcontroller calculates, using the system of equations 4, the angles α and δ, respective Dice-Kit 3 microcontroller calculates the angles γ and β. When the 10ms break finishes, the two local intelligent drives start to modify the position reference of the five servos.

The position of the rotor of a servo depends on the duty cycle of the command PWM signal. Because the frequency of the PWM is 50Hz, it means that every 20ms the local intelligent drive must update the position reference of the servo. When the driver received a new command and the robot is in transition from the initial point to the destination point it calculates the current command for the servo with the following equation:

$$u_k = \frac{\left|u_{ki} - u_{kf}\right| \cdot \tau}{\tau_c} \qquad (6)$$

where τ is the time that passed from the beginning of the transition; τ_c is the time in which the robot should effectuate the transition and u_k is the angle of the servo at τ moment. If the driver finishes the transition and it does not receive a new command, it will maintain the commands for each servo at the same value.

C. The Management of the CAN network

There are four categories of messages on the CAN network:

- from the computer to the motion coordinator;
- from the motion coordinator to local intelligent drives, message which contains the next position of the servo;
- from the motion coordinator to the local intelligent drives, message that contains the motion time;
- status and error messages.

Because all the devices may need to transfer a piece of information at any moment of time, it is important to implement a table of priorities among the messages.

The message with the biggest priority is the one from the motion coordinator to the local intelligent servo drives with the time of transition. The next message as priority is the message with the next position of the servos followed by the status and error messages.

The messages from the computer to the motion coordinator have the smallest priority because in this case there is no need for deterministic communication.

V. APPLICATION SOFTWARE

The software of the local intelligent drives contains the command strategy, the synchronization algorithm and the communication protocol. It was designed and simulated in Softune, using a C compiler.

The motion coordinator has an ANSI C library, which is useful for the programmer, because it contains functions for creating the commands and manages the message from/to the local intelligent drives.

To test the applications a program had been created for the host computer that has the interface in Fig. 4. The program allows the user to create an array of points from the space in which the gripper of the robotic arm should pass. The application sends the array of points to the motion coordinator and it executes them.

In the interface there are two canvas objects where can be observed the position of the articulated arm robot (see Fig. 4) from the planes X-Y and Z-W.

Because all the intelligent devices are connected on the same network, the host computer has direct access to the messages transmitted by the motion coordinator to the local intelligent servo drives. In this way the host computer can show the exact position of the robot on the two canvases, a task that would have been very difficult in distributed control architecture with a hierarchical architecture.

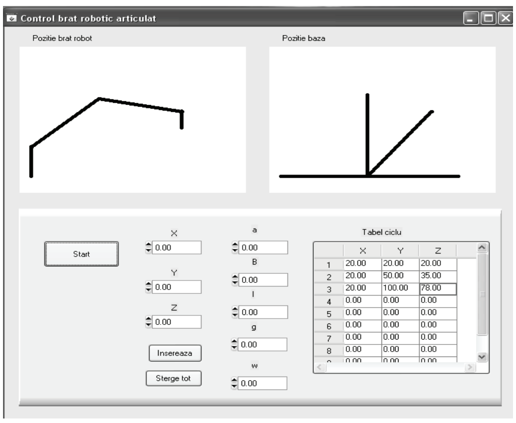


Fig. 4. The interface of program from the host computer.

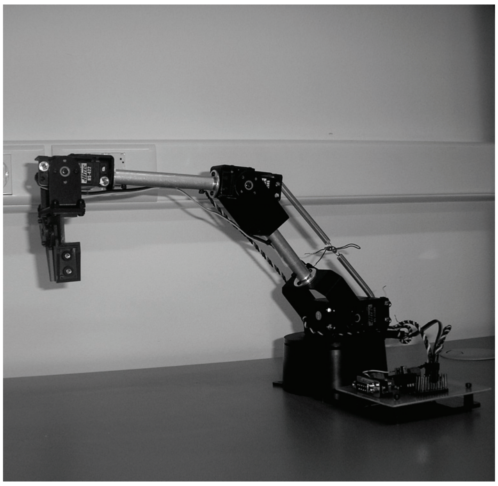


Fig. 5. The articulated arm robot.

VI. Conclusions

The distributed architecture from Fig. 3 was used to control an articulated arm robot from LynxMotion (see Fig. 5). The design of the project and the practical tests proved that the distributed architecture is a very effective one. It allows the designer to create and test small parts of the project and then put all of them together. Furthermore, the complexity of the programs from microcontrollers is smaller because each member of the control structure receives a certain task, rather than in centralized control systems where the central processing unit should do all the tasks.

Another fact proved by practical tests is that the communication tasks, among different modules, are easier because there is a single network and messages do not have to pass through different processors. Furthermore, the microcontroller developing board needs only a communication driver and it is cheaper.

The disadvantage of this control system is that it does not have a global position controller. The motion controller and the local intelligent servo drives ensure only that the position references for the servos are synchronized. This is mainly because servos have their own position control loops and the local intelligent drivers cannot read these values. For this reason when the robot has to position the gripper at a high speed or when the robot needs to move a heavy weight, it produces position errors.

Acknowledgement

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/6/1.5/S/6.

References

[1] D. Jouve and D. Bui, "Distributed Intelligence Allows New Dimension in Manufacturing Processes Control," *in 2000 Proc. Intelligent Motion Conf.*, pp. 343–348.

[2] D. Jouve and D. Bui, "CANopen Servo Drives Provides High Performance Motion Control," in *2002 Proc. Intelligent Motion Conf.* pp. 1–6.

[3] Y. S. Kim, H. S. Kim and W. H. Kwon, "A Design of a Network-based Robot Control System using FIP," in *2000 Proc. Asian Control Conf.*, pp. 2229–2234.

[4] K. C. Lee, S. Lee and M. H. Lee "Worst Case Communication Delay of Real-Time Industrial Switched Ethernet With Multiple Levels," *IEEE Trans. on Industrial Electronics*, Vol. 53, No. 5, pp. 1669–1678 October 2006.

[5] S. Y. Lin, C. Y. Ho and Y. Y. Tzou, "Distributed Motion Control Using Real-Time Network Communication Techniques," in *2000 Proc. Power Electronics and Motion Control Conf.*, pp. 843–847.

[6] L. Molinari-Tosatti, "The ABB Flexible Automation Approach for Developing New Software Tools Dedicated to Advanced Manufacturing and Servicing Technologies," in *2002 Proc. IEEE Inter. Symp. in Industrial Electronics*, pp. 46-51.

[7] J. M. Pacas, "Drives 2000 End of the Roadmap? – From the State-of-the Art to the Future Trends," in *2000 Proc. Intelligent Motion Conf.*, pp. 41–50.

[8] C. E. Pereira and L. Carro, "Distributed Real-time Embeded Systems: Recent Advances, Future Trends and Their Impact on Manufacturing Plant Control," *Annual Reviews in Control*, Vol. 31, No. 1, pp. 81÷92, June 2007.

[9] R. Schönfeld, M. Franke, H. Hasan and F. Muller, "Intelligent Drives in Systems with Decentralized Intelligence," in*1993 Proc. Conf. on Power Electronics and Applications*, pp. 489 of the Fifth European 494.

[10] F. Schewe and J. Jasperneite, "Further Development of Fieldbus Technology to Support Multi-Axis Motion," in *1999 Proc. Intelligent Motion Conf.*, pp. 33–38.

[11] A. Valera, J. Salt, V. Casanova and S. Ferrus, "Control of Industrial Robot with a Fieldbus," in *1999 Proc. IEEE Inter. Conf. on Emerging Technologies and Factory Automation*, pp. 1235÷1241.