The Distributed Control of an Articulated Arm Robot

Dan PUIU, Florin MOLDOVEANU, and Mihai CERNAT

"Transilvania" University of Brasov, B-dul Eroilor nr. 29, 500036 Brasov, Romania
puiudan@unitbv.ro, moldof@unitbv.ro, m.cernat@unitbv.ro

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. The control structure has a pyramidal topology where in the front of the pyramid is a host computer, in the middle is the motion controller and on the base of pyramid, close to the arm robot actuators, are the local intelligent servo drives.

1. Introduction

Today, the decreasing product and technology life cycles have made fixed automation systems cost prohibitive. 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 control tasks in the central controller, the distributed control architecture is based on fieldbus communication and Digital Signal Processing technology so that decentralized control tasks can reside in the intelligent servo drives. The general structure of distributed control architecture is presented in figure 1 [5, 6, 7].

Different communication levels are used according to the control hierarchy. At the higher control level, the factory floor network is connecting the different factory processes to the host computer [6].

At a given process control level, the control fieldbus is connecting the different control components used in the same process to the host programmable logic controller (PLC). At the motion control level, fieldbus communication has also been introduced recently [3, 4].

For interpolated axes control a servo fieldbus is connecting the different servo drives to the motion coordinator, while in simple positioning applications the complete motion control task is performed inside a positioner servo drive. Thus the positioner servo drives are directly connected at the control fieldbus level [6].

This distributed control architecture allows easy building of complex applications in a modular manner with a high degree of reusability. Because of the fieldbus communication facilities and the digital drive technology, the parameters of the complete control system can be easily re set for a new product without any hardware modifications. The process can also be easily modified by adding or removing control components without major modifications to the control system [4, 6].

This paper presents the 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.

The servos are connected to two local intelligent drives which are commanded and synchronized by a motion controller. The synchronization of the two drives is important because the robot has to achieve a certain trajectory.

The motion controller receives the trajectory of the robot via RS232, generates the appropriate command for each servo and then sends the data to the local intelligent drive. Subsequently, it sends a synchronization message to the local intelligent drives that carry out the commands.

The user can view on the interface of the computer program the position of the robot while it is moving.

2. Control architectures

In a distributed control structure, the application programming and the process control are still performed inside the host PLC or motion controller.

However, the servo drives assume more responsibility for motion control, like hardware and software position limit supervision, motor braking modes control and safe low speed operation for the machine commissioning. In regard of this point, it is important to distinguish between interpolated and independent axes applications, respectively [5, 7].

Many robots and machine-tools require interpolated motion control (several axes trajectories must be continuously coordinated). In this case, the axes trajectory must be calculated by the same processor at a high frequency rate in order to maintain the axes coordination [1, 2].

As shown in figure 2, the distributed motion control architecture suitable for multi-axes interpolation is based on intelligent servo drives which perform the complete servo control task including position, speed and current loops plus power conversion.

The host motion controller performs the multi-axes trajectory calculation and sends the digital position set point values to each servo drive via a serial bus communication [4].

Most of the motion control applications in the automation area do not require interpolation between axes (the axes trajectories are independent). In this case a centralized trajectory calculation is no longer necessary and the trajectory calculation can be distributed inside the servo drives [3].

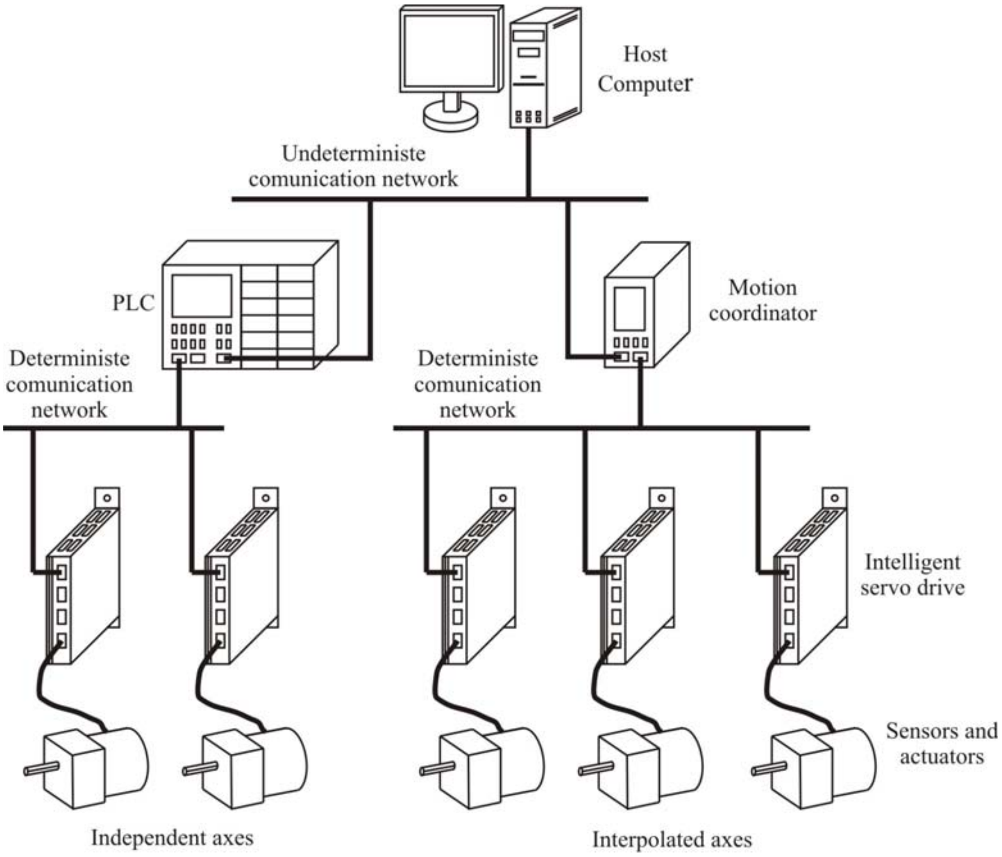


Fig. 1. Distributed control architecture based on fieldbus communications [5, 6, 7].

A host PLC is well suited for providing the application sequences control. This solution based on intelligent servo drives provides a high performance motion control fully integrated in the PLC environment [3, 5].

For a distributed motion architecture, where the axes trajectories are independent, the motion control tasks are distributed inside each stand-alone intelligent servo drive, which can be programmed independently in order to perform a given axis motion control task. The motion control sequences, the trajectory calculation and all the servo loops plus the power conversion are fully integrated in the local servo drive.

The fieldbus communication between the host PLC and the servo drives is only used for the parameter setting and the global process control and monitoring [5, 7].

These solutions are much more effective than the centralized ones in terms of wiring cost reduction, setup and diagnostic facilities, due to the serial bus communication. The servo performances are also greatly improved because the current, speed and position digital servo loops are all inside the servo drive. The complete servo loop gain values are automatically calculated inside the servo drive when the auto-tuning command is executed. Feedforward terms are also implemented in order to follow complex motion profile at high speed with a minimum tracking error value [5, 6, 7, 8].

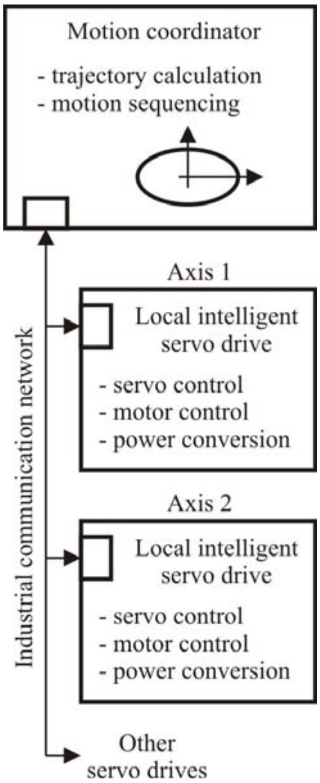


Fig. 2. Interpolated axes motion control structure [6].

3. The geometric model of the articulated arm robot

Figure 3 presents the model of the robot, which is actuated by five servos. Because the servos have their own control loops

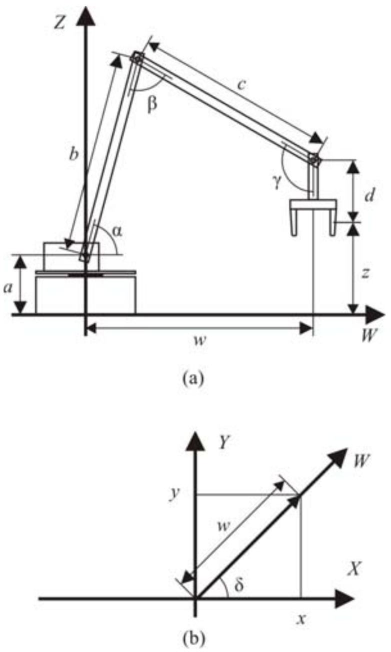


Fig. 3. Robot model: (a) view of the arm plane; (b) view of the rotation base plane.

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

To simplify the model it will be considered that the gripper of the articulated arm is always perpendicular on the X-Y plane.

The direct geometric model of the articulated arm can be easily deducted from figure 3(a) and figure 3(b):

$$\begin{aligned} x &= w\cos\delta \\ y &= w\sin\delta \\ z &= a + b\sin\alpha - c\cos(\alpha+\beta-90) - d \end{aligned} \qquad \text{(1a, b, c)}$$

where: x, y, and z represent the position of the gripper; a, b, c, and d are the dimensions of the robot arms; α, β, and γ are the angles between the robot arms; δ is the angle of the robot arm in the X-Y plane; w is the distance between the base of the robot and the projection of the gripper in the X-Y plane, which can be determined from figure 3a, yielding the following equation:

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

By solving the system consisting of equations (1), (2) and:

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

the inverse geometric model results:

$$\begin{aligned} \alpha &= \arccos\left(\frac{w(b-c\cos\beta) - c(z-a+d)\sin\beta}{(z-a+d)^2+w^2}\right) \\ \beta &= \arccos\left(\frac{b^2+c^2-w^2-(z-a+d)^2}{2bc}\right) \\ \gamma &= 180-(\alpha+\beta) \end{aligned} \qquad \text{(4a, b, c)}$$

$$\delta = \arctan\left(\frac{y}{x}\right). \qquad \text{(4d)}$$

It can be observed that α depends upon β. It is preferred to use this equation because it is simpler, else the second equation should be inserted in the first one yielding a very complex relationship [9, 10].

4. The control structure of the robot

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

The structure of the system is presented in figure 4. The computer, which is the host computer from figure 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 in an array of segments of lines and sends it to the Dice-Kit 1 board (see fig. 4).

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 the local intelligent drives.

A requirement for a complete distributed control system is using five local intelligent drives, one for each servo of the robot; however, the discussed case concerning a prototype with a small robotic arm, only two will be used. The Dice-Kit 2 board will control angles α and δ of the robot, while the Dice-Kit 3 board will control angles β and γ, plus the position of the gripper.

4.1. The motion coordinator

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

If the motion coordinator needs to move the robot gripper to a certain (x, y, z) point in space, it has to calculate, using the system of equations 4, the angles (α, β, γ, δ) and send these to the local intelligent drives.

In order to modify the speed of the gripper, the motion coordinator sends at the end of the command messages the time in which the gripper should move from the current point to the destination point.

The variation of this time determines the variation of the speed of the motors. Because of the speed limitation of the articulated arm robot, it is useful for the motion coordinator to know, after it determines the angles for each axis, how to calculate the minimum time necessary for the move.

This time can be calculated with the following equation:

$$t_{\min} = \max_{k=1}^{5}\left(\frac{|u_{ki}-u_{kf}|\cdot\tau_k}{90}\right) \qquad (5)$$

where: $t_{\min}$ represents the minimum time; k is an index for each servo motor; u_{kl} is the initial angle of the servo motor k; u_{kf} is the destination angle of the servo motor k; τ_k is the time that servo motor k needs to change its position by 90 degrees, at full speed.

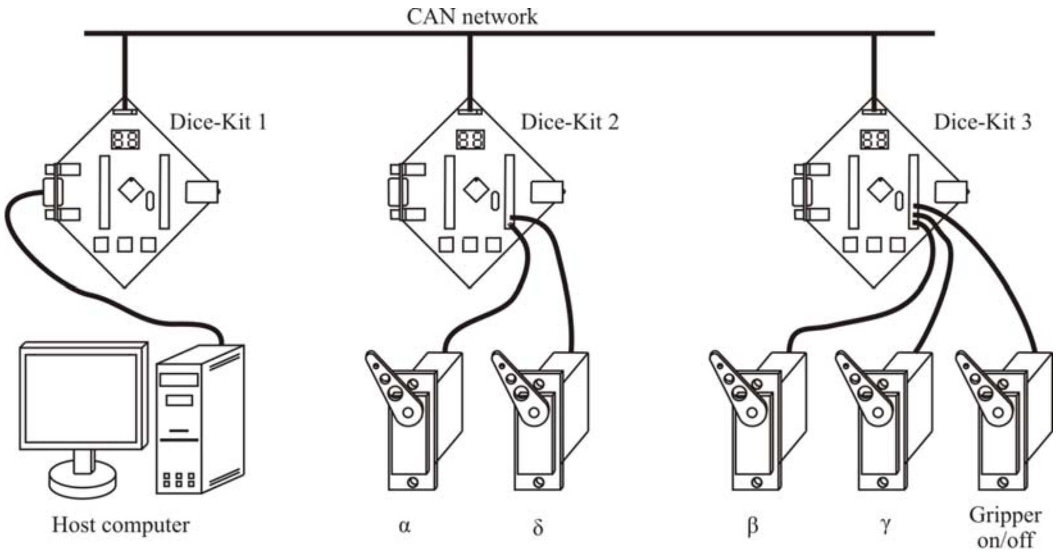


Fig. 4. The distributed control system for the articulated arm robot.

4.2. The local intelligent servo drives

The five servos, that actuate the robot, have their own control loops, including the positioning loop, and can position their rotor under an angle of 180^o. The position of the rotor varies linearly with the duty cycle of a PWM signal that has a frequency of 50 Hz.-

When a local intelligent drive receives a new set of angles for the axis that it controls and the time in which the transition should take place, it will gradually change the reference of each servo in order to generate a linear move for each axis.

As the frequency of the PWM signal for each motor is 50 Hz, the local intelligent drive should update the position of the servo every 20 ms. Once the driver has received a new command and the robot is in transition from the initial to the destination point it will calculate 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 passed from the beginning of the transition; τ_c is the time in which the robot should complete the transition; u_k is the angle of the servo at moment τ.

Once the driver has finished the transition and does not receive a new one, it will maintain the commands for each servo at the same value.

4.3. The synchronization of the intelligent servo drives

When the motion coordinator needs to move the robot gripper to a specific point in space it has to calculate the positions for each axis and send the data to the local intelligent drives. After that, it calculates the time of the transition and sends a broadcast message including this time. In this way all the local intelligent drives receive the time of the transition at the same time. After that, local intelligent drives have a break of 10ms, time in which they prepare for the transition and initiate it. As the local intelligent drives have the same type of microcontrollers, the same type of quartz oscillator and their programmes are optimized in order to start the 10 ms timer when they receive the broadcast message, the moment when the two local drives initiate the transition is the same.

Further more, as the local intelligent drive respects the time received from the motion coordinator for the transition, the movement of the four axes of the articulated arm robot is a synchronised one.

5. Application software

The software of the local intelligent drives contains the command strategy, the synchronisation algorithm and the communication protocol. It was designed and simulated in Softune, using a C compiler. The motion coordinator has an ANSI C library, useful for the programmer, because it contains functions for creating the commands and manages the message from/to the local intelligent drives.

The most important functions of the library are those transforming direct to inverse cinematic, and reverse, using equations (1) and (4). Other functions calculate the message that should be sent to the local intelligent drives in order to move the robot arm gripper to a specific point in space.

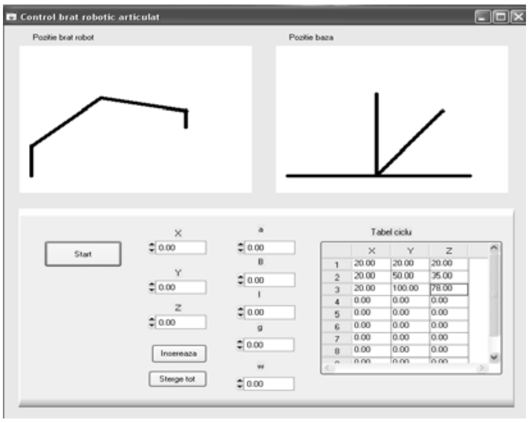
In order to test the application a programme for the host computer was developed with the interface shown in figure 5.

The programme allows the user to create an array of points from the space in which the robotic gripper arm should pass. The application sends via RS232 the array of points to the motion coordinator that executes them.

The interface includes two canvas objects where the approximate position of the articulated arm robot (see fig. 6) can be observed from planes X-Y and Z-W. The position is approximate because the application estimates it.

6. Conclusions

The design of the project and the practical tests proved the effectiveness of distributed architecture. It allows the designer to create and test the project by small parts and subsequently



X: 0.00

Y: 0.00

Z: 0.00

a: 0.00

B: 0.00

I: 0.00

g: 0.00

w: 0.00

Tabel ciclu

	X	Y	Z
1	20.00	20.00	20.00
2	20.00	50.00	35.00
3	20.00	100.00	78.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00

Fig. 5. The interface of the control program.

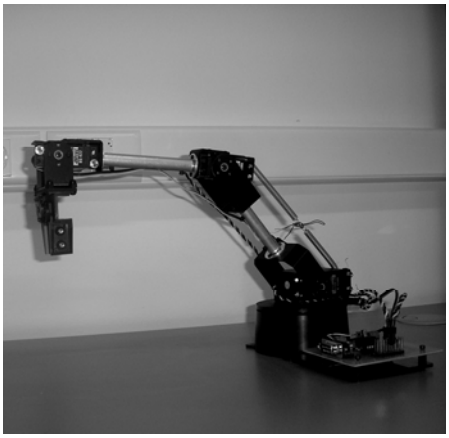


Fig. 6. The articulated arm robot

assemble these. Further more the complexity of the programmes from the microcontrollers is smaller because each member of the control structure receives a certain task, rather than in centralised control systems where the central processing unit should do all the tasks.

The practical experiments proved the benefits of the synchronization method, because the robot axes move in an interpolated way.

An inconvenient of the selected distributed control architecture is the fact that it has two different communication networks, making it difficult for the host computer to communicate with the local intelligent drive. For this reason it was decided to estimate the position of the articulated arm robot in the interface of the host computer. To allow direct communication between the host computer and the drives the motion coordinator needs a new task, hence the complexity of its programme increases.

The solution for this problem is to use a control architecture with the intelligent systems distributed on the same network. In this way, while the traffic on the network increases, every member of the discussion can take all the information it needs from the network.

The further development of the project includes changing of the distributed control architecture with a same network distributed intelligence one.

7. 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.

8. References

[1] M.Y. Cheng, C.C. Lee, “On real-time contour error estimation for contour following tasks.”, in *Proc.* 2005 *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Monterey, California, USA, 2005, pp. 1047-1051.

[2] M.Y. Cheng, C.C. Lee, “Motion Controller Design for Contour-Following Task Based on Real-Time Contour Error Estimation.”, *IEEE Trans. on Industrial Electronics*, vol. 54, no. 3, pp. 1047÷1051, 2007.

[3] F. He, W. Tong, Q. Wang, “Synchronization Control Strategy of Multi-motor System Based on Profibus Network.”, in *Proc.* 2007 *IEEE Int. Conf. on Automation and Logistics*, Jinan, China, 2007, pp. 3029÷3034.

[4] F. He, S. Zhao, “Research on synchronous control of nodes in distributed network system.”, in *Proc.* 2008 *IEEE International Conf. on Automation and Logistics*, Qingdao, China, 2008, pp. 2999-3004.

[5] D. Jouve, D. Bui, “Distributed Intelligence Allows New Dimension in Manufacturing Processes Control.” in *Proc.* 5*th Int. Intelligent Motion Conf.*, Nuremberg, Germany, 2000, pp. 343-348.

[6] D. Jouve, D. Bui, “CANopen Servo Drives Provides High Performance Motion control.” in *Proc.* 7*th Int. Intelligent Motion Conf.*, Nuremberg, Germany, 2002, pp. 1-6.

[7] J.M. Pacas, “Drives 2000 End of the Roadmap? – From the State-of-the Art to the Future Trends.” in *Proc.* 5*th Int. Intelligent Motion Conference*, Nuremberg, 2000, p. 41-50.

[8] J.M. Pacas, “Drives in the Automation”, in *Proc.* 2000 *IEEE International Symposium on Industrial Electronics*, Cholula, Mexic, 2000, pp. TU24÷TU25 (B).

[9] G.E. Chatzarakis, “Nodal Analysis Optimization Based on the Use of Virtual Current Sources: A Powerful New Pedagogical Method,” *IEEE Trans. Education*, vol. 52, no. 1, February 2009, pp. 144-150.

[10] D.Ch. Karamousantas, G.E. Chatzarakis, G.N. Korres, P.J. Katsikas, “Obtaining State Equations for Planar Non-degenerate Linear Electric Circuits using Mesh Analysis with Virtual Voltage Sources," *IJEEE* (*International Journal of Electrical Engineering Education*), Manchester University, Vol. 45, No. 3, July 2008, pp. 239-250.