Bulletin of the *Transilvania* University of Braşov
Series I: Engineering Sciences • Vol. 4 (53) No. 1 - 2011

DASTS: DISTRIBUTED ARCHITECTURE FOR SUN TRACKING SYSTEM MOUNTED ON MOBILE PLATFORMS

D. PUIU[1] D. FLOROIAN[1] F. MOLDOVEANU[1]

Abstract: *In present time, the need for flexibility in different organizational structures, impose top equipment for infrastructure. These mobile platforms are equipped with many electrical appliances and the needs for power limits the consumption of these platforms or impose the rinse of infrastructure with power generators. In this paper we present a sun tracking system mounted on a mobile platform and based on two local intelligent servo drives connected on a CAN network to a motion controller which has implemented a solar position algorithm. The motion controller determines the position of the sun on the site, and then, coordinates the two drives in order to correctly position the photovoltaic panel.*

Key words: *distributed architecture, sun tracking, mobile platform, autonomous robots.*

1. Introduction

The decreasing product and technology life cycles have made fixed automation systems cost prohibitive. As a result of that the automation users need flexible automation systems that can be modified upgraded and reused easily. This trend ensures that automation system have a long term competitive position.

For a multi-axis positioning system the “classical” architecture contains a very powerful processor which has the following tasks: generating the robot trajectory, calculating the position and speed references for the actuators, synchronization of the actuators, information acquisition from sensors and control the actuators. Further more, the architecture requires a lot of wiring from the processing unit to the sensors and actuators. This approach has a series of drawbacks, because: it requires a lot of processing power and it is hard to debug and extend.

As a result of these new requirements, the traditional architecture of the multi-axis electrical drive system, which is costly and hard to maintain, is replaced by an open distributed structure, based on high level communication protocols, dedicated for multi-axis control [3]. Besides the reduced cost and maintaining prices, the multi-axis distributed control structure allows the design of complex applications in a short period of time with high capacity of extension.

In order to solve the problem, the solution is to replace the centralized control architecture based around a single

[1] Centre “Control Process Systems”, *Transilvania* University of Braşov.

host controller with a distributed control architecture where the tasks of the central controller are distributed among the local drives. In this way the local intelligent drives requires less computation power and they are cheap [3]. In consequence this system is also powerless and could be easily mounted in an eco-friendly mobile platform. Furthermore because each local intelligent drive has only a few tasks it is easier to debug it [4], [5].

Even if the intelligence is distributed through the control system, it has to work in real-time. This implies the implementation of a synchronization strategy of the nodes of the industrial communication network [16]. To solve these problems it can be use an industrial communication network that has implemented in its protocol specific mechanisms for synchronizing its nodes such as CAN or PROFIBUS.

The sun tracking systems, in general allow solar panels to collect up to 50% more energy than that can be collected using stationary solar panels. Furthermore, the mobility of the platform excludes the possibility to optimize the orientation based on latitude [13], [14]. Like the majority of sun tracking systems [1], the proposed system moves the solar panel on two axes.

The paper presents the implementation of distributed control architecture for a two axis sun tracking system, mounted on a mobile platform. The axes are actuated with two servos, which receive position commands from two local intelligent drives. The distributed control system is based on a CAN network where are connected the two local drives, a motion controller and a small mobile host computer. This architecture reduces the cost of the control system because even if it uses more microcontrollers they are cheaper because they have fewer tasks to do. Furthermore, the control structure can be in any moment extended or reduced without complex hardware modifications.

2. Distributed Control Architectures

The first industrial distributed systems are based on OSI standard for network protocols. It result a hierarchic architecture which has its components grouped on four layers (see Figure 1).

Each component from the system receives commands from the devices from the superior layer and sends command to the devices from the inferior layers [5], [7].

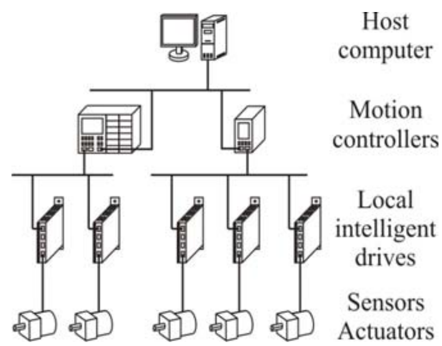

Fig. 1. *Distributed control system with hierarchic architecture* [4]

The highest layer is the management one which ensures a graphic user interface; the downloading and uploading of programs and parameters; the monitoring of the stocks and analysis of the working times [5], [7].

The next layer is the control process one, where the execution times are a critical aspect. The intelligent units from this layer have the following main tasks: the generation of the trajectory and the coordination and the synchronization of the axis [5], [7]. Because the communication time is critical, the network that links this layer with the lower ones must ensure real time communication [2], [8], [10].

The local intelligent drives from the axis control layer receive motion trajectory for the actuator and executes it [5], [7], [15].

This structure has a few disadvantages that make it hard to implement [6], [12]:

- it has several communication protocols with increase the price of the automation and makes it hard to debug;
- from the top of the structure to the bottom and reverse the data have to pass through several processor and the speed of transfer is decreased;
- the combination between the horizontal traffic with the vertical traffic is hard to be implemented for a 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 (see Figure 2), all the intelligent units from all the layers [9], [13].

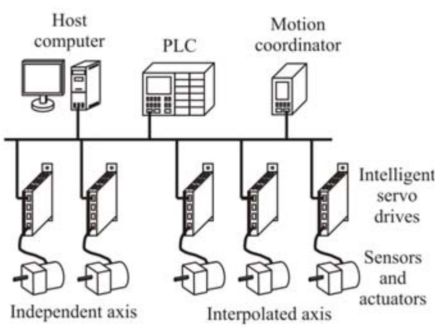

Fig. 2. *Distributed control system with hierarchic architecture* [9]

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

3. The Structure of the DASTS

The intelligence of the control structure is composed of three microcontroller developing boards Dice-Kit from Fujitsu, a Global Positioning System (GPS) module, a host computer and two intelligent servo drives. The boards have their own CAN controllers and the computer has a serial to CAN adapter.

Because the distributed control system with hierarchic architecture has a series of disadvantages the control structure of the DASTS will have a single CAN network. The structure of the system is presented in Figure 3.

The two axis of the sun tracking system are actuated with two servos, which have their own position, speed and current control loops. The first servo actuates the azimuth angle photovoltaic panel and it receives its position reference from the Dice-Kit 2 board. The second servo is for the elevation angle and it is commanded by the Dice-Kit 3 board.

The sun trucking system is mounted on a wheel chair type mobile platform that is actuated with two bipolar stepper motors. The positions of the two wheels are controlled by the two intelligent servo drises.

3.1. The Control of DASTS

The host computer has the role of master of the distributed control system. It receives information about the global coordinates of the DASTS, the local hour and the North Pole orientation from the Dice Kit 1. It, also, sends information about the position of photovoltaic panel and position and speed references of the right and left wheels to Dice Kit 2, to Dice Kit 3 and to the two intelligent drives.

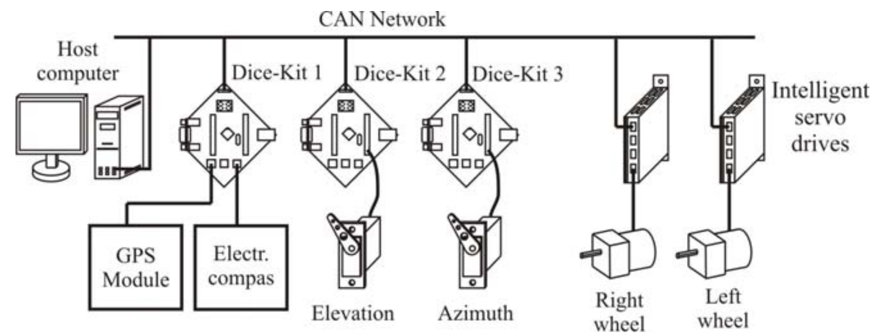

Fig. 3. *DASTS control structure*

The mobile platform has two working programs. The first program controls only the wheels of the mobile platform and can be used to position the autonomous robot in the working space. The second program starts after the robot reaches the destination point and it is used to orientate the photovoltaic panel.

Using the global coordinates of the DASTS and the local hour, the motion coordinator calculates the azimuth and elevation angles of the mobile platform (see Figure 4).

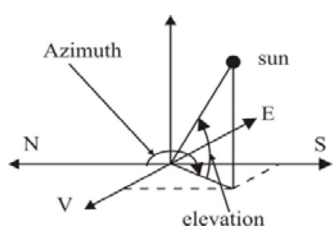

Fig. 4. *The azimuth and elevation angles*

The algorithm used to determine the angles was developed by the National Renewable Energy Laboratory (NREL) from Colorado, USA and is presented in detail in the paper [11].

The host computer determines the orientation angles of the panel using the system of equations:

$$\begin{cases} u_1 = u_{elv}, \\ u_2 = u_{azm} + u_{or}, \end{cases} \quad (1)$$

where u_1 is the angle between the photovoltaic panel and the surface of the mobile platform, u_{elv} is the elevation angle of the mobile platform position, u_2 is angle between the front of the robot and the photovoltaic panel, u_{azm} is the azimuth angle of the mobile platform position and u_{or} is the angle between the front of the robot and the magnetic North Pole.

3.2. The Photovoltaic Position Orientation

The two servos, that actuate the sun tracking system, have their own control loops, including the positioning loop. Servos are composed of an electric motor mechanically linked to a potentiometer. Pulse-width modulation (PWM) signals sent to the servo are translated into position commands by electronics inside the servo. When the servo is commanded to rotate, the motor is powered until the potentiometer reaches the value corresponding to the commanded position.

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 its servo in order to generate a linear move for the axis.

Because the frequency of the PWM signal for each motor is 50 Hz it means that at every 20 ms the local intelligent drive should update the position of the servo. The local intelligent servo drive calculates the immediate command with the following equation:

$$u_k = \frac{\left|u_{ki} - u_{kf}\right| \cdot \tau}{\tau_c}, \tag{2}$$

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 one it will maintain the last command.

3.3. The Mobile Platform Position Control

The wheels of the mobile platform are actuated by two stepper motors, which are controlled by the intelligent servo drives (see Figure 3).

The host computer uses the two intelligent servo drives to change the position of the DASTS mobile platform in the working space.

When the two wheels of the robot are spinning in the same direction but with different angular velocities the trajectory of the mobile platform will be a portion of a circle, as shown in Figure 5. The different angular velocities, implies the fact that a wheel does more steps than the other one.

It is useful for the host computer to know how to calculate the number of steps that each wheel should do, if it wants to move the robot to point defined relatively to the current position.

The distance that the left wheel covers, when the robot moves from the start point to the destination point, can be calculated in two modes:

$$d_l = \left(r + \frac{a}{2}\right)\theta = \frac{b\,\alpha\,n_l}{2}, \tag{3}$$

where d_l is the distance that the left wheel covered; a is the distance between the wheels; b is the diameter of the wheel; r the radius of the trajectory circle; θ the angle of the circle portion, that is covered when the robot moves, α is the angle of the motors step and n_l the number of steps that the left motor made.

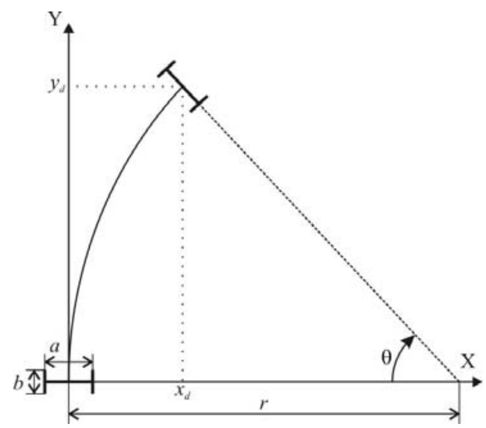

Fig. 5. *The trajectory of the robot when the angular velocities of the two wheels are different*

The equation for the distance that the right wheel covers can be calculated in a similar way:

$$d_r = \left(r - \frac{a}{2}\right)\theta = \frac{b\,\alpha\,n_r}{2}, \tag{4}$$

where d_r is the distance that right wheel covered and n_r the number of steps that the left motor made.

From the Eq. (3) and Eq. (4) can be calculated the number of steps that each wheel should do in order to move the robot to the destination point:

$$\begin{cases} n_r = \dfrac{2r - a}{b\alpha}\theta, \\ n_l = \dfrac{2r + a}{b\alpha}\theta. \end{cases} \tag{5}$$

The values of the angle θ and the radius *r* can be determined using the system of equations:

$$\begin{cases} \sin(\theta) = \dfrac{y_d}{r}, \\ \cos(\theta) = \dfrac{r - x_d}{r}, \end{cases} \quad (6)$$

where x_d, y_d are the relative, to the current point, coordinates of the destination point. Solving this system of equations will result:

$$\begin{cases} r = \dfrac{x_d^2 + y_d^2}{2\,x_d}, \\ \theta = \arcsin\left(\dfrac{2\,x_d\;y_d}{x_d^2 + y_d^2}\right). \end{cases} \quad (7)$$

When to host computes needs to move the mobile platform to a certain point from the working space it has to send to the properly number of steps to the two intelligent servo drives.

4. The Experimental Model

The distributed control structure was tested with the mobile platform presented in Figure 6. The photovoltaic positioning system is a LynxMotion articulated arm robot [17]. Originally the robot had four degrees of freedom, but for this project it had been used only two of them.

The program for the host computer was designed in LabWindows/CVI and it manages the sun tracking system. From here the user has access to control the position of the mobile platform in the working space or can monitor the position and the parameters of the sun tracking system.

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

For the host computer was developed an ANSI C library, which it is useful for the programmer, because it contains functions to create the commands and manages the message from/to the local intelligent drives.

Fig. 6. *The hardware of the sun tracking system*

The program from host computer contains three threads. The first thread is working all the time and it controls the other threads and the graphic user interface.

The second thread stars only when the DASTS mobile platform reaches the destination point and it needs electrical energy. The thread monitors CAN network in order to receive messages from the Dice-Kit 1. At the beginning it needs information about local hour, position and orientation of the mobile platform. After the second thread sets its parameters, it monitor only the messages about the local hour and send the appropriate commands to the Dice-Kit 1 and 2 modules.

The last thread is used only when the user needs to move the DASTS mobile platform and it control the two intelligent servo drives of the distributed control system.

4. Conclusions

The design of the DASTS and the practical tests proved that the distributed architecture is a very efficient one because it allows the designer to create and test small parts of the control structure and then put all of them together.

Furthermore the complexity of the programs from the microcontrollers is smaller because each logical unit of the control structure receives fewer tasks, rather than in centralized control systems where the central processing unit should do all the tasks. This is an important issue in the case of a mobile platform (i.e. a truck) where the hardware resources are very limited.

The design of the distributed control system allows that the same motion controller to coordinate several sun tracking systems. This provides a simple extension to many mobile platforms (i.e. a large number of trucks) for providing more power with fewer resources.

Another fact proved by the experimental implementation is that the communication tasks, among the different modules, are easier because there is a single network and the messages do not have to pass through different processors.

Acknowledgments

This paper is supported by the Sectorial 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. Alexandru, C., Tatu, I.N.: *Modelling and Analising of the Photovoltaic Tracking Systems as Multibody Systems.* In: Bulletin of the *Transilvania* University of Braşov (2010) Vol. 3 (52), Series I, p. 47-54.
2. Alldredge, G.W.: *PID and Model Predictive Control in a Networked Environment.* In: Master's Thesis, Case Western Reserve University, Ohio, USA, 2007.
3. He, F., Tong, W., Wang, Q.: *Synchronization Control Strategy of Multi-motor System Based on Profibus Network*. In: IEEE Inter. Conf. on Automation and Logistics, Jinan, 2007, p. 3029-3034.
4. Jouve, D., Bui, D.: *CANopen Servo Drives Provides High Performance Motion Control*. In: 38th Inter. Intelligent Motion Conf. - PCIM'02, Nuremberg, 2002, p. 1-6.
5. Khalil, A.A., El-Singaby, M.: *Position Control of Sun Tracking Systems*. In: 46th Inter. Midwest Symposium on Circuits and Systems - MWSCAS'03, Cairo, 2003, p. 1134-1137.
6. Lee, K.C., Lee, S., Lee, M.H.: *Worst Case Communication Delay of Real-*

Time Industrial Switched Ethernet with Multiple Levels. In: IEEE Trans. on Industrial Electronics **53** (2006), p. 1669-1678.
7. Lin, S.Y., Ho, C.Y., Tzou, Y.Y.: *Distributed Motion Control Using Real-Time Network Communication Techniques*. In: 3rd Inter. Power Electronics and Motion Control Conf., Beijing, 2000, p. 843-847.
8. Pacas, J.M.: *Drives 2000 End of the Roadmap? - From the State-of-the Art to the Future Trends*. In: 37th Inter. Intelligent Motion Conf. - PCIM'2000, Nuremberg, 2000, p. 41-50.
9. Pereira, C.E., Carro, L.: *Distributed Real-time Embeded Systems: Recent Advances, Future Trends and Their Impact on Manufacturing Plant Control*. In: Annual Reviews in Control **31** (2007), p. 81-92.
10. Pinho, L.M., Vasques, F.: *Reliable Real-Time Communication in CAN Networks*. In: IEEE Trans. On Computers **52** (2003), p. 1594-1607.
11. Reda, I., Afshin, A.: *Solar Position Algorithm for Solar Radiation Applications*. Technical Report, NREL, Colorado, TP-560-34302, 2008.
12. Solea, R.C.: *Sliding Mode Control Applied in Trajectory-Tracking of WMRs and Autonomus Vehicles*. In: Ph.D. Thesis, University of Coimbra, Portugal, 2009.
13. Sun, D.: *Position Synchronization of Multiple Motion Axes with Adaptive Coupling Control*. In: Proc. of the IFAC Fifteenth World Congress, Barcelona, Spain, 2003, p. 997-1005.
14. Taherbaneh, M., Ghafori, H., Rezaie, A.H., Karbasian, S.: *Combination of Fuzzy-Based Maximum Power Point Tracker and Sun Tracker for Deployable Solar Panels in Photovoltaic Systems*. In: IEEE Inter. Fuzzy Systems Conference - FUZZ-IEEE 2007, London, 2007, p. 1-6.
15. Tovar, E., Vasques, F.: *Real-Time Fieldbus Communications Using Profibus Networks*. In: IEEE Trans. on Industrial Electronics **46** (1999), p. 1241-1251.
16. Wu, Q., Li, Y., Qin, Y.: *A Scheduling Method Based on Deadline for CAN-based Networked Control Systems*. In: Proc. of the IEEE Inter. Conf. Mechatronics and Automation, Luoyang, 2006, p. 345-350.
17. *** *AL5B Robotic Arm Combo Kit datasheet*. Available at: www.lynxmotion.com. Accessed: 30.05.2011.