5

Networked Control Systems for Electrical Drives

Baluta Gheorghe and Lazar Corneliu
"Gh. Asachi" Technical University of Iasi
Romania

1. Introduction

The use of networks as a media to interconnect the different components in an electrical drive control system is increasing in the last decades. Typically, the employ of a network on a control system is desirable when there is a large number of distributed sensors and actuators. Systems designed in this manner allow for easy modification of the control strategy by rerouting signals, having redundant systems that can be activated automatically when component failure occurs, and in general they allow having a high-level supervisor control over the entire plant. The flexibility and ease of maintenance of a system using a network to transfer information is a very appealing goal. Due to these benefits, many industrial companies and institutes apply networks for remote control purposes and factory automation (Yang, 2006), (Lian et al., 2002), (Bushnell, 2001), (Antsaklis & Baillieul, 2004), (Baillieul & Antsaklis, 2004). Control applications utilize networks to connect to Internet in order to perform remote control at much farther distances than in the past without investing on the whole infrastructure.

The connection between new network based control systems and teaching allows many universities to develop virtual and remote control laboratories (Valera et al., 2005), (Casini et al., 2004), (Saad et al., 2001). For several years, at the Departments of Power Electronics and Electrical Drives (Baluta & Lazar, 2007) and Automatic Control and Applied Informatics (Carari et al., 2003), (Lazar & Carari, 2008) from "Gh. Asachi" Technical University of Iasi, virtual and remote laboratories for electrical drive systems and process control have developed using a Networked Control System (NCS).

This chapter presents the experience of the electrical drives control group at the "Gh. Asachi" Technical University of Iasi in developing remote control laboratory for electrical drive systems. A SCADA environment has been chosen to implement the network based control architecture. This architecture allows the user to remotely choose a predefined controller to steer the electrical drives systems or to design a new one. Using SCADA software facilities, students can develop themselves new networked control systems for the set ups from the laboratory. The main advantage of the network based control structure is the user interface, which allows analyze the electrical drives system performances, to tune the controller and to test it through the remote laboratory. During the experiments, it is possible to change the set point, the operating mode and some typical controller parameters. Experimental results can be displayed showing the real running experiment and can be checked through on-line plots.

Source: New Approaches in Automation and Robotics, Book edited by: Harald Aschemann, ISBN 978-3-902613-26-4, pp. 392, May 2008, I-Tech Education and Publishing, Vienna, Austria

The chapter is organized as it follows. Section 2 illustrates the main features and the architecture of the networked control system laboratory. In Section 3, typical working sessions are described. Section 4 provides conclusions and future developments.

2. Remote control architecture

In order to achieve the remote control of the electrical drive systems, a Web server is used which assures the process distribution for different users (clients) via Intranet and Internet. The Intranet is the local computer network of the Department of Power Electronics and Electrical Drives from "Gh. Asachi" Technical University of Iasi. The electrical drive systems distribution is realized using a NCS architecture, which implements both configurations: direct structure and hierarchical structure (Tipsuwan, 2003).

2.1 NCS architecture

The developed NCS architecture has the layout from Fig. 1.

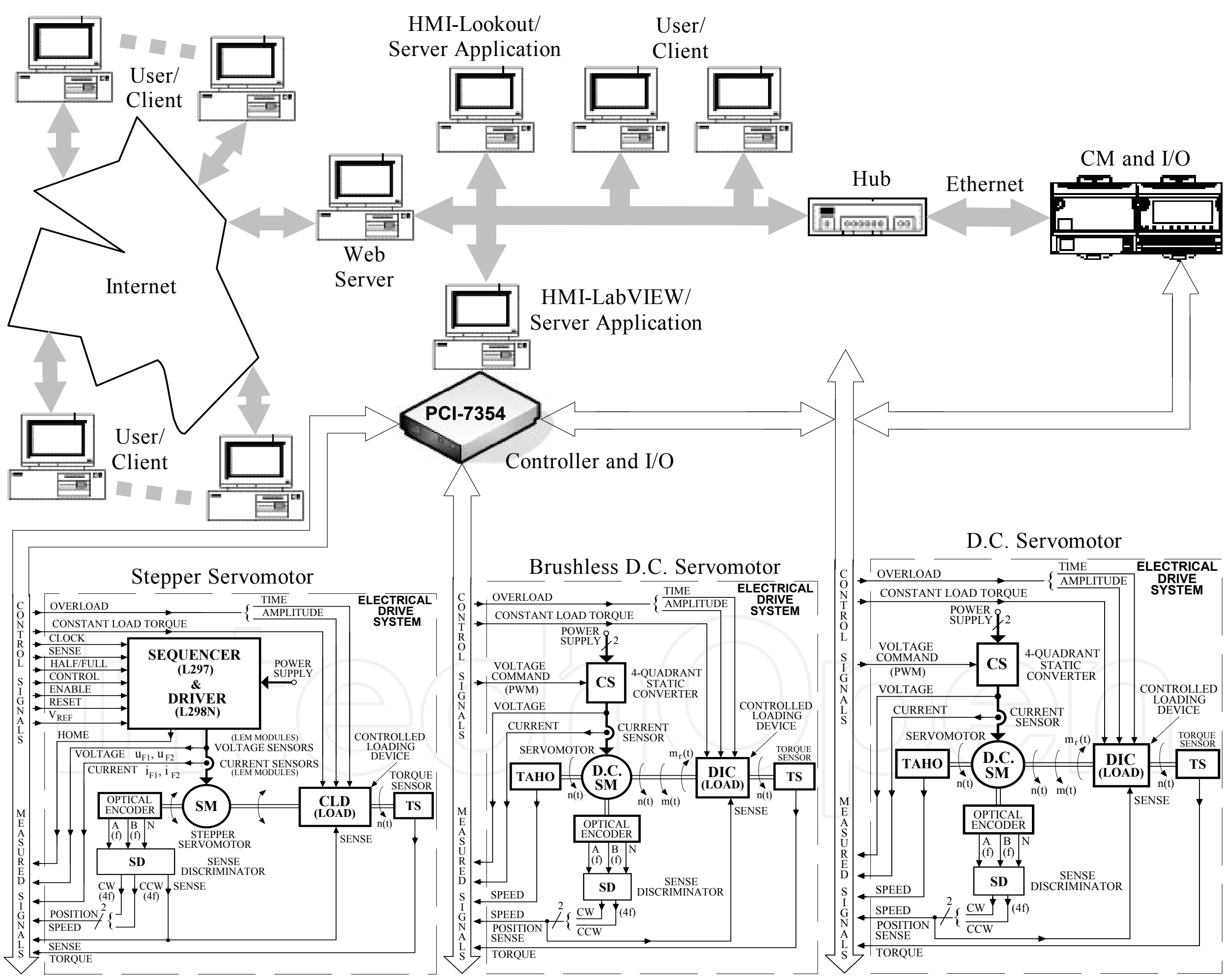


Fig. 1. Remote control architecture.

A user can have access to the process and run an experiment in real time using Intranet and Internet. The user can design and implement different control structures for electrical drive systems employing SCADA software facilities or, for a given control structure, he is able to implement and test PID control algorithms and tuning procedures. SCADA software enables programmers to create distributed control applications having supervisory facilities and a Human-Machine Interface (HMI). As SCADA software, Lookout is used for the direct structure and LabVIEW for the hierarchical structure. All external signals start and arrive at HMI/SCADA computer.

The laboratory architecture allows running experiments while interacting with instruments and remote devices. The I/O remote devices permit data acquisition from sensors and supplying control signals for actuators, using A/D and D/A converters.

The NCS architecture offers the possibility to remotely choose a predefined control structure to handle the electrical drives system variables or to design a new control application, using SCADA software facilities.

In the first case, using the remote control architecture the students have the possibility to practice their theoretical knowledge of electrical drive systems control in an easy way due to process access by a friendly user interface. The second opportunity offered to the students is to design a new networked control architecture which allows creating a new HMI/SCADA application to remotely control a process, using Lookout and, respectively, LabVIEW facilities (Carari et al., 2003).

The software architecture can be split in two parts: one concerns the control of the physical process – server side and the other relates to the user interface – client side. The server runs on the Microsoft Windows NT platform and is based on Lookout for direct structure and, respectively, LabVIEW environment for hierarchical structure.

The server application contains the HMI interface and fulfils the following functions:

- implements the control strategies;
- communicates with I/O devices through object drives;
- records the signals in a database;
- defines the alarms.

The client process contains a HMI interface, similar or not with those from server application, and has the following characteristics:

- allows modifying remotely the parameters defined by application server through a Web site;
- communicates with server application;
- displays the alarms defined by the server application.

The remote control architecture is mainly intended for educational use and it is employed for electrical drive control course. The aim is to allow students to put in practice their knowledge of electrical drives and control theory in an easy way without restrictions due to process availability through laboratory and project works. One of the main features is the possibility of integrating in the control loop of the remote process the user-designed controller. The interface for the controller synthesis is very friendly.

2.2 NCS in the direct structure

The NCS in the direct structure is composed of a computer of the Intranet, called HMI/Lookout that achieves the local communications with the process using Ethernet protocols. The remote electrical drive system, a D.C. brush servomotor, is connected with the communication module (CM) able to transfer data from/to I/O device to/from HMI/SCADA computer via a communication system. The communication module and I/O devices are implemented with National Instruments modules, FP1600 (Ethernet) for communication and, respectively, Dual-Channel Modules for I/O devices. The Lookout environment has been chosen to implement HMI/SCADA application. For D.C. servomotor, a cascade control structure is used in order to control the speed from the primary loop and the current from the secondary loop. The current controller is locally implemented and the speed controller is remotely implemented using Lookout environment. The cascade control structure allows the monitoring of control loops variables and the command of the overload at the servomotor shaft.

2.3 NCS in the hierarchical structure

Hierarchical structure is composed of a computer of the Intranet, called HMI/LabVIEW with a PCI motion controller board (National Instruments PCI-7354) and Analog & Digital I/O devices.
DSP controllers available today are able to perform the computation for high performance digital motion control structures for different motor technologies and motion control configuration. The level of integration is continuously increasing, and the clear trend is towards completely integrated intelligent motion control (Kreidler, 2002). Highly flexible solutions, easy parameterized and "ready-to-run", are needed in the existent "time-to-market" pressing environment, and must be available at non-specialist level.
Basically, the digital system component implements through specific hardware interfaces and corresponding software modules, the complete or partial hierarchical motion control structure, i.e., the digital motor control functionality at a low level and the digital motion control functionality at the higher level (see Fig. 2).

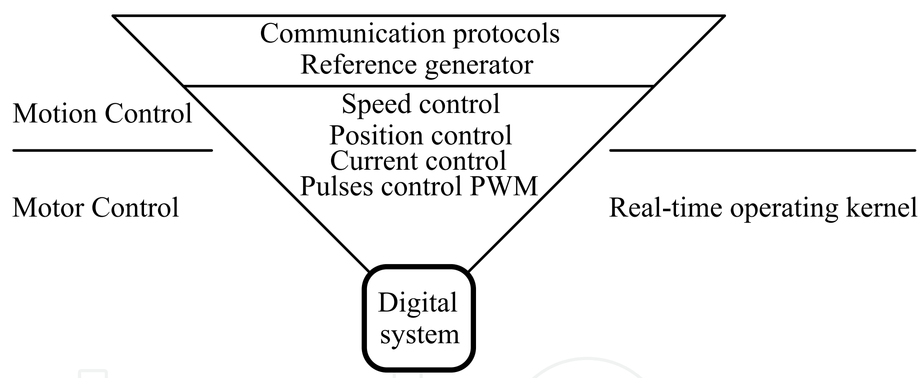


Fig. 2. Motion system structure hierarchy.

The National Instruments PCI-7354 controller is a high-performance 4-axis-stepper/D.C. brush/D.C. brushless servomotors motion controller. This controller can be used for a wide variety of both simple and complex motion applications. It also includes a built-in data acquisition system with eight 16-bit analog inputs as well as a host of advanced motion trajectory and triggering features. Through four axes, individually programmable, the board can control independently or in a coordinated mode the motion. The board architecture, which is build around of a dual-processors core, has own real-time operating system . These board resources assure a high computational power, needed for such real-time control.
Three electrical drive systems, based on a unipolar or bipolar stepper servomotors, D.C. brush servomotors and a D.C. brushless servomotors are linked to the remote control architecture. The connection is achieved with the I/O devices from PCI motion controller

board, which also contains a remote controller implemented using a DSP and real-time operating system, as is presented in Fig. 3.

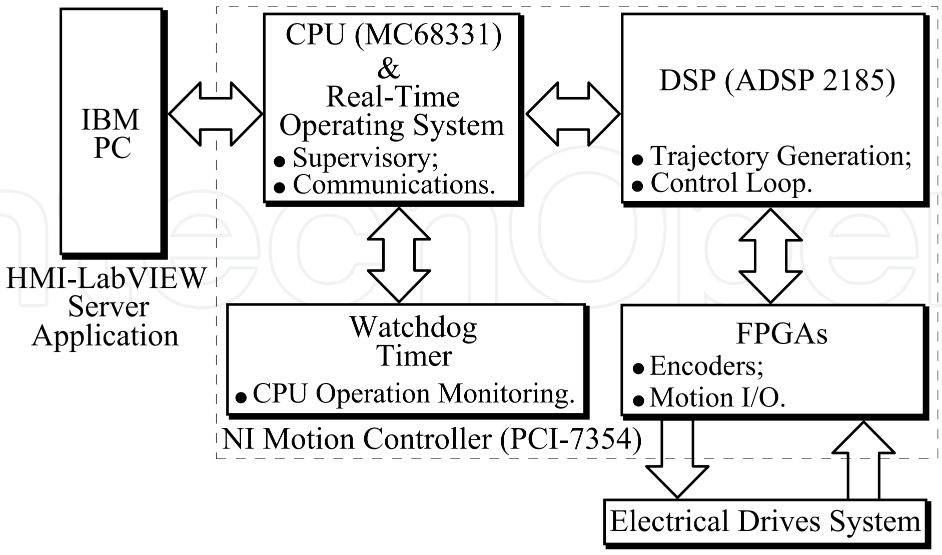


Fig. 3. Motion controller board structure.

Functionally, the architecture of the National Instruments PCI-7354 controller is generally divided into four components (see Fig. 4):

- supervisory control;
- trajectory generator;
- control loop;
- motion I/O.

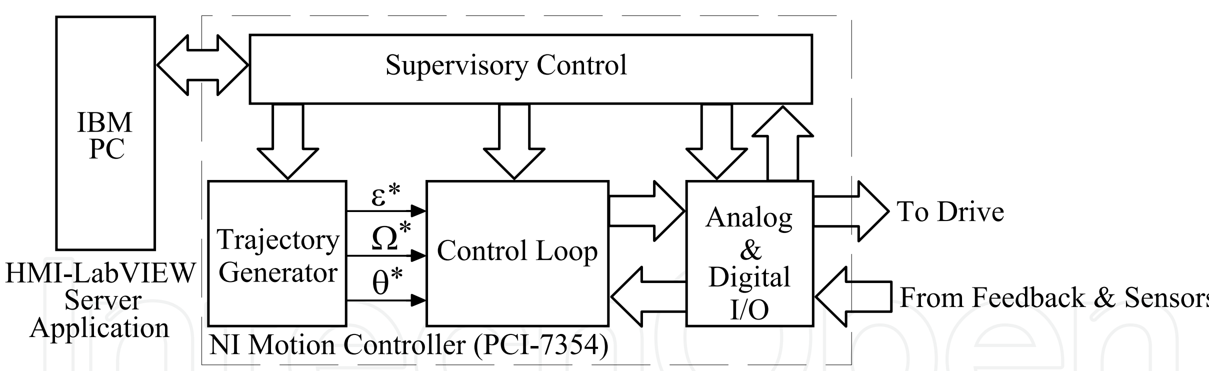


Fig. 4. Functional architecture of the NI PCI-7354.

Supervisory control performs all the command sequencing and coordination required to carry out the specified operation. Trajectory generator provides path planning based on the profile specified by the user, and control loop block performs fast, closed-loop control with simultaneous position, velocity, and trajectory maintenance on one or more axes, based on feedback signals.
The LabVIEW environment has been chosen to implement HMI/SCADA application. The development environment used to complete the applications is LabVIEW 7.0, which beside the graphic implementation that gives easy use and understanding takes full advantage of the networking resources. Using NCS hierarchical structure, control architecture for stepper

servomotors, D.C. brush servomotors and D.C. brushless servomotors supervisory was developed which allows the following functions:
-open-loop control of stepper servomotors using constant frequency or prescribed profile (trapezoidal or S-curve move profile, see Fig. 5);

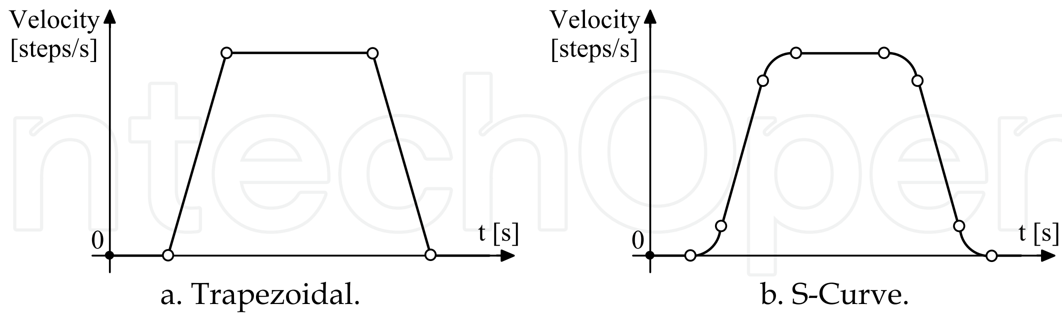


Fig. 5. Move Profile.

- closed-loop control of stepper servomotors (speed and position);
- the operating mode of the stepper servomotors can be with full step, half step or microstepping mode;
- open-loop control or closed-loop control of D.C. brush servomotors;
- self-commutation control of D.C. brushless servomotor speed, which allows very high speeds;
- sensorless control for D.C. brushless servomotor.

3. Session description

This section presents three examples to illustrate the manipulation of instruments and real devices for low power electrical drives education on the Web. These examples exemplify the laboratory works dedicated to control a D.C. brush servomotor electrical drives system and a stepper servomotor electrical drives system.

3.1 NCS in the direct structure: D.C. brush servomotor electrical drives system

The set-up for D.C. brush servomotor control is presented in Fig. 6 and the layout can be inspected in Fig. 1.
The set-up consists in a D.C. brush servomotor, a D.C. brush generator used as load, an incremental encoder and a tachometer as speed transducer, a controlled loading device and a control block which includes an analogical controller for current loop. The set-up is used to illustrate cascade control and how this structure can reject the load disturbances. The primary variable is the motor speed, the secondary is the armature current and the disturbance is an overload introduced with a D.C. brush generator coupled on the servomotor shaft. The user has to tune the two controllers of the speed and current loops in order to obtain optimal performances and disturbance rejection.
When the application is started, the main panel from Fig. 7 is displayed. The main panel contains in the right side, the layout of the control system, real time data from the sensors and actuator and allows the reference setting, the overload command, the choice of operating mode: manual or automatic and the changing of the tuning parameters of the PI speed controller (Baluta & Lazar, 2007).

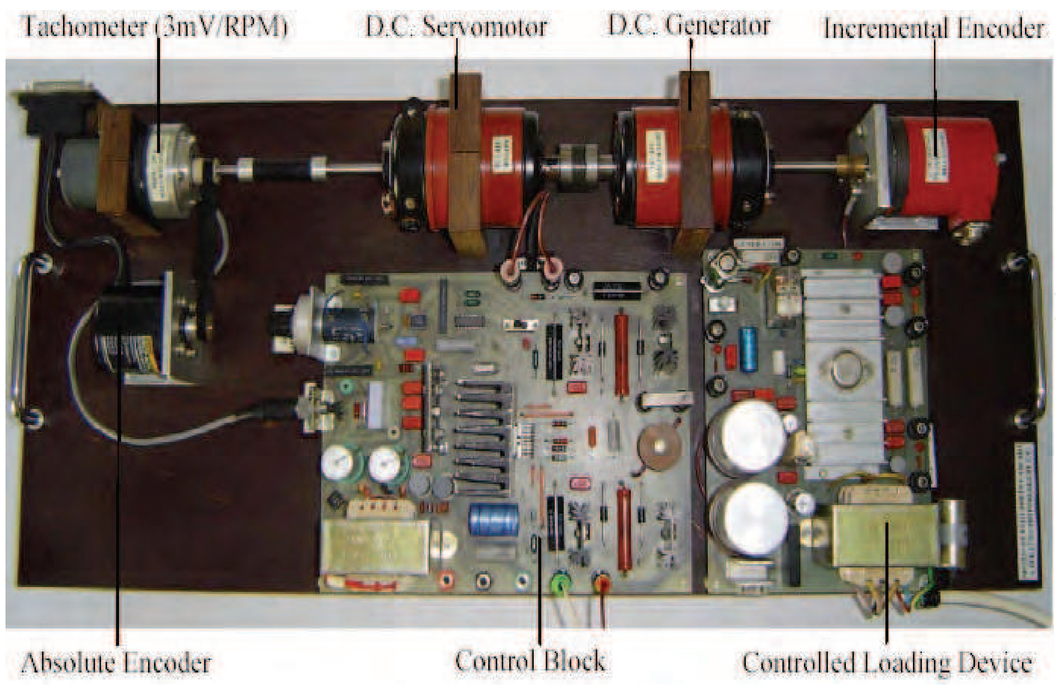


Fig. 6. D.C. brush servomotor electrical drives system.

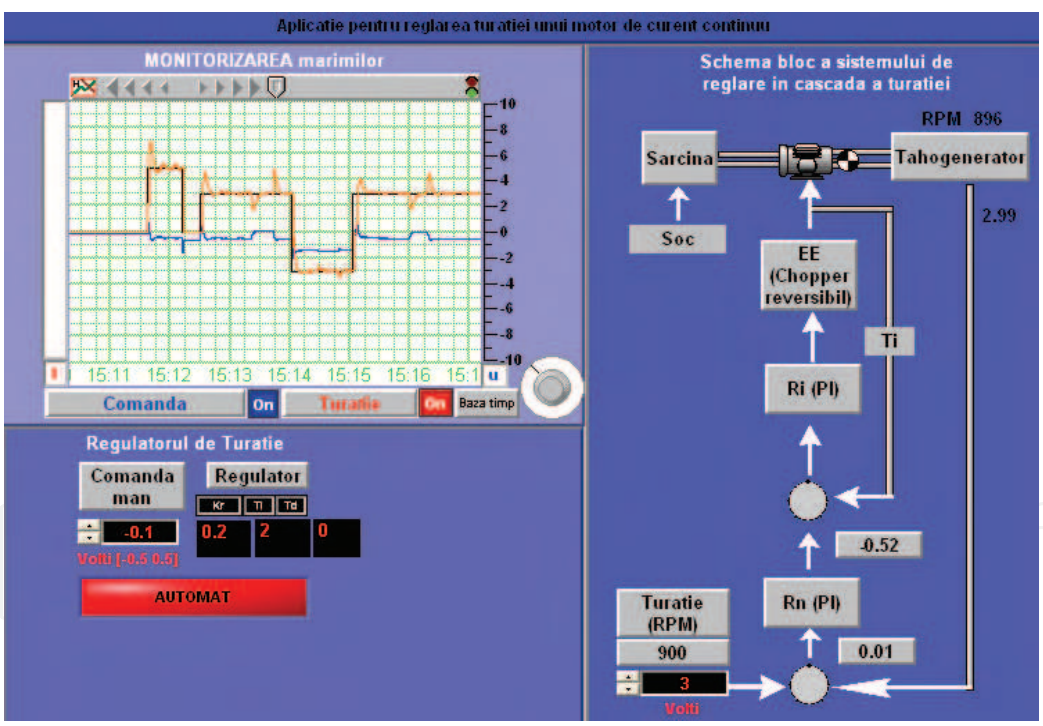


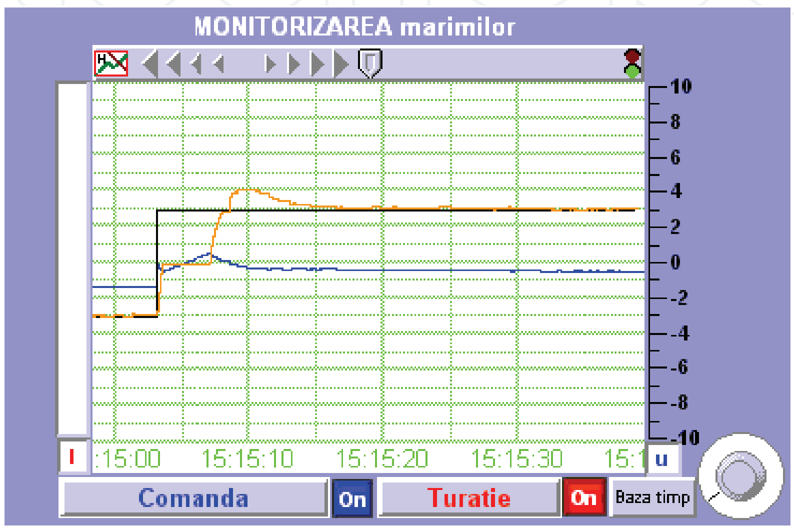
Fig. 7. Main panel for D.C. brush servomotor electrical drives system.

All these commands can be done using the left down side of the main panel. In the left up side, there is a monitor window, which allows the displaying of the main variables of the control loops. The Monitor window has the possibility to represent only the selected signals by using on/off buttons or to stop the variables displaying in order to analyse the signals or to change the time scale through a potentiometer button. All control system real time data are recorded in a database and can be employed for a future analysis (Baluta & Lazar, 2007).

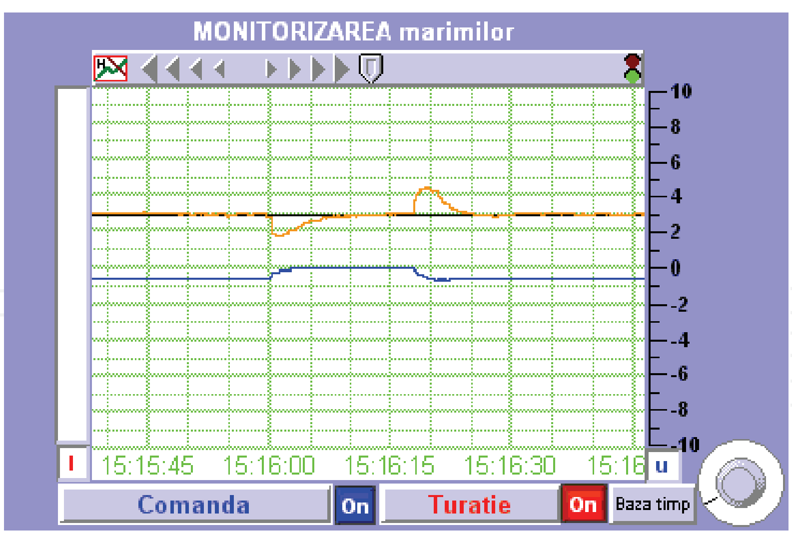
The remote control of the D.C. brush servomotors implements the following functions:

- the reversible speed control;
- the on-line tuning of the parameters of speed controller;
- load command for servomotor shaft.

The monitor window of the main panel from Fig. 7 shows both the speed response for a different set point step variations and the load rejection and the current evolution. Details regarding disturbance rejection are given in Fig. 8. The numerical controller's parameters which have led to such results are Kr=0.2, Ti=2 and Td=0. Due to the well-chosen and adjusted controller, the system's response to a step variation of the set point is very good. At the same time, a good compensation of the disturbance can be noticed (the load torque).



a. Step variation of the set point.



b. Load disturbance rejection.

Fig. 8. Monitoring panel (set point-black, speed-brown, armature current-blue).

3.2 NCS in the hierarchical structure: D.C. brush servomotor electrical drives system

The D.C. brush servomotor has the following features:

- rating armature voltage U_n=20V;

- rating armature current I_n=2A;
- rating speed n_n=2400 rpm;
- armature winding resistance R_a=1.31 Ω;
- armature winding inductance L_a=7.58 mH.

The D.C. brush servomotor is driven by a reversible chopper with the L292 specialized integrated circuit (Baluta, 2004). The motor's shaft is connected with an incremental optical encoder that gives 1000 pulses/rev. (this means 4000 pulses/rev. at the output of the NI PCI-7354 controller). The D.C. brush servomotor is loaded using a controlled loading device. Among the features of the loading system, the authors emphasize (Baluta, 2004):

- the possibility to load the electric drive motor with a reactive load torque in a wide range of speed, including very low speeds;
- the possibility to impose a constant load torque operation mode or an overload operation mode.

The implemented control system uses the algorithm from Fig. 9, which allows to generate in real-time the move trajectory and to change the motion parameters. In order to program the motion control system, the developer employs LabVIEW environment with specific virtual instruments for motion control from the FlexMotion library.

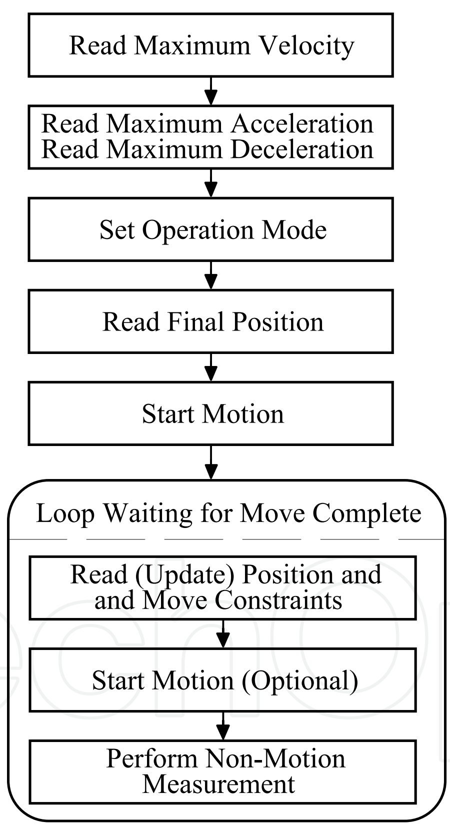


Fig. 9. Position-based straight-line move algorithm.

Within such application, the user has the possibility to study either:

- the command regimes in *speed open-loop*;
- the position control regimes of the positioning system.

The graphic program associated to the first type of study is that shown in Fig. 10. It is structured by means of a single *while ()* loop, preceded by an initialization sequence allowing the configuration of the system (selection of chopper control signal, setting of optical encoder parameters, definition of the speed measurement units). The control elements of the loop creates the possibility that the user could modify the chopper control signal and/or to establish the servomotor loading degree. The measures acquired in view of their graphical representation are the electrical ones (current in the armature winding and supply modulated voltage of the armature winding) as well as the mechanical ones (position and speed of the drive system). Leaving off the application is achieved explicitly, by means the *Stop* push-button, available in the graphical user interface.

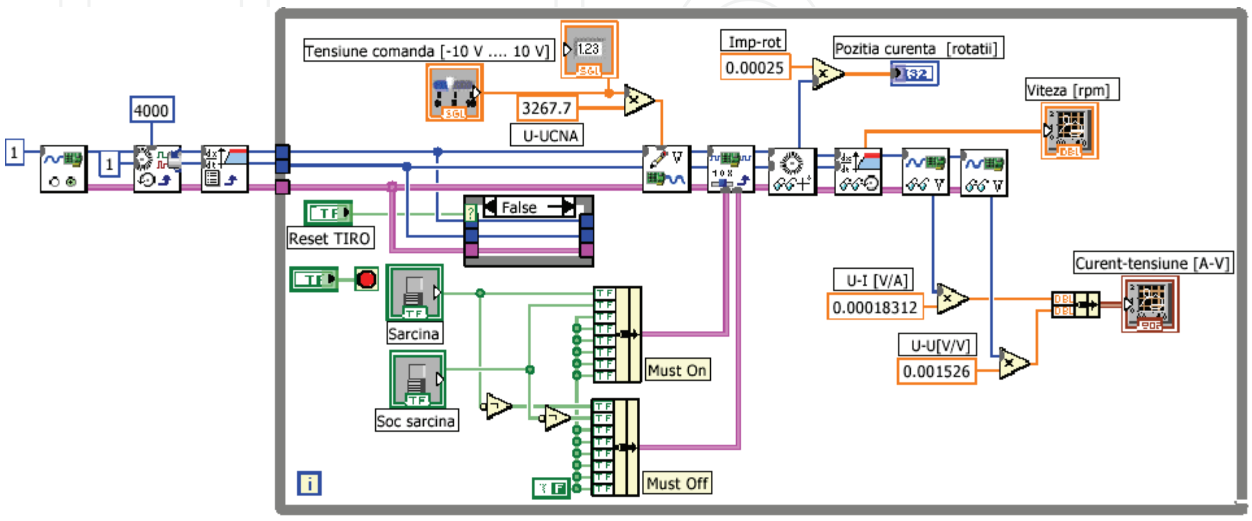
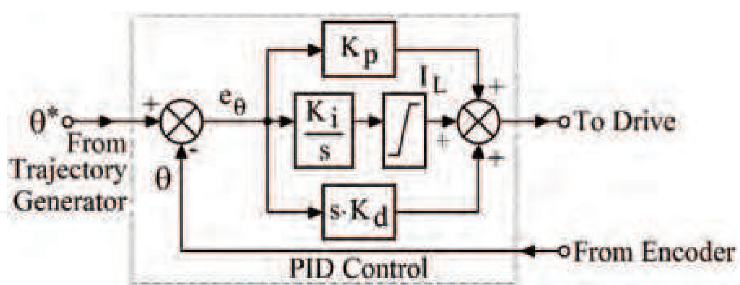


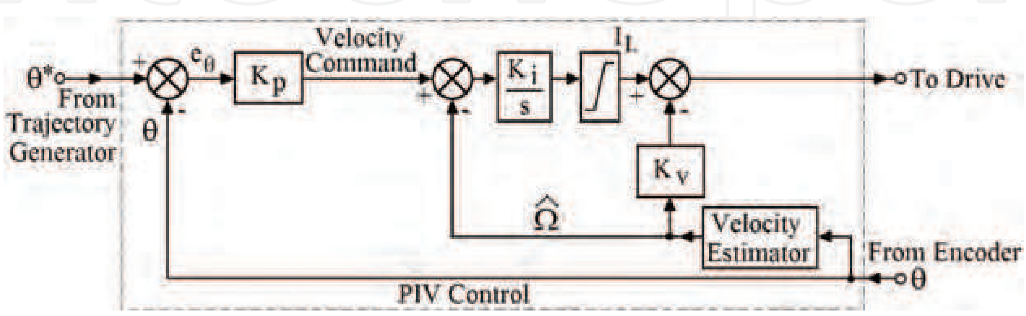
Fig. 10. LabVIEW diagram for the command regimes in speed open-loop.

The vast majority of motion control algorithms employed in industrial applications are of two forms (Ellis & Lorentz, 1999]:

- the well-known PID position loop (Fig. 11a);
- an average velocity loop cascaded with a position loop (Fig. 11b).



a. PID Servocontrol topology.



b. PIV. Servocontrol topology.

Fig. 11. Servocontrol topology.

On the other hand, in order to achieve near zero following or tracking error, feedforward control is often employed. A requirement for feedforward control is the availability of both the velocity, $\Omega^*(t)$, and acceleration, $\varepsilon^*(t)$, commands synchronized with the position commands, $\theta^*(t)$.
An example of how feedforward control is used in addition to the second servocontrol topology is shown in Fig. 12. The National Instruments PCI-7354 controller can be configured for any of the above servocontrol topologies.

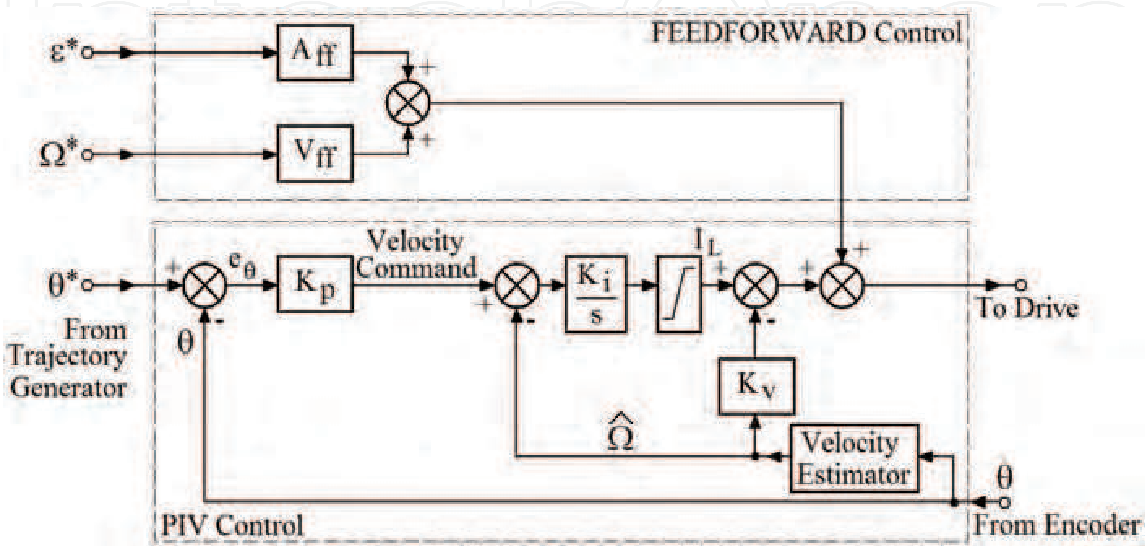


Fig. 12. Basic feedforward and PIV control topology.

In order to build a positioning application, the developer has to follow the stages shown in Fig. 13, which represent the steps required to design a motion application. The learner can also perform the second to fifth stages of motion application design (Bauer & Fedok, 2003) (Fedok & Bauer, 2005), (Saliah et al., 1999).

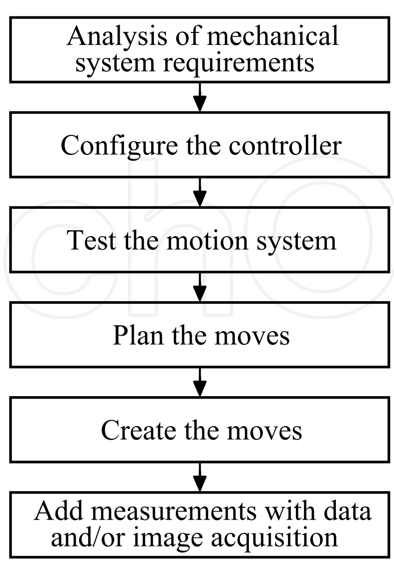


Fig. 13. Generic steps for designing a motion application.

To introduce a student to motor control, the D.C. brush servomotor set-up will provide the opportunity to learn about this motor without ever attending a laboratory session at their

institution. Since the majority of students do not have prior technical experience, this remote set-up is very economical and offers students a great tool for learning about the servomechanism and data acquisition.
To ensure proper performance for servomechanism the motion control system must be tuned and tested. Then the type of move profile is planed. Motion constraints are the maximum velocity, acceleration, deceleration and jerk that the system can handle. Trajectory parameters are expressed as a function of motor shaft revolutions. The trajectory generator takes into account the type and the constraints of motion and generates values of instantaneous trajectory parameters in real-time.
Basically, the student can learn about the parameters of the D.C. servomechanism, by tuning the control loop parameters and setting the motion constrains to make execute the operations, i.e., change the rotation direction, position and speed. Also during the experiment, the student can visualise critical input and output points in the D.C. brush servomotor electronic interface using virtual oscilloscopes. Since this system is in real-time mode, the learner observes changes in position, direction, and velocity as the motor rotates after being commanded. Fig. 14 shows the graphical user interface specific to this application.

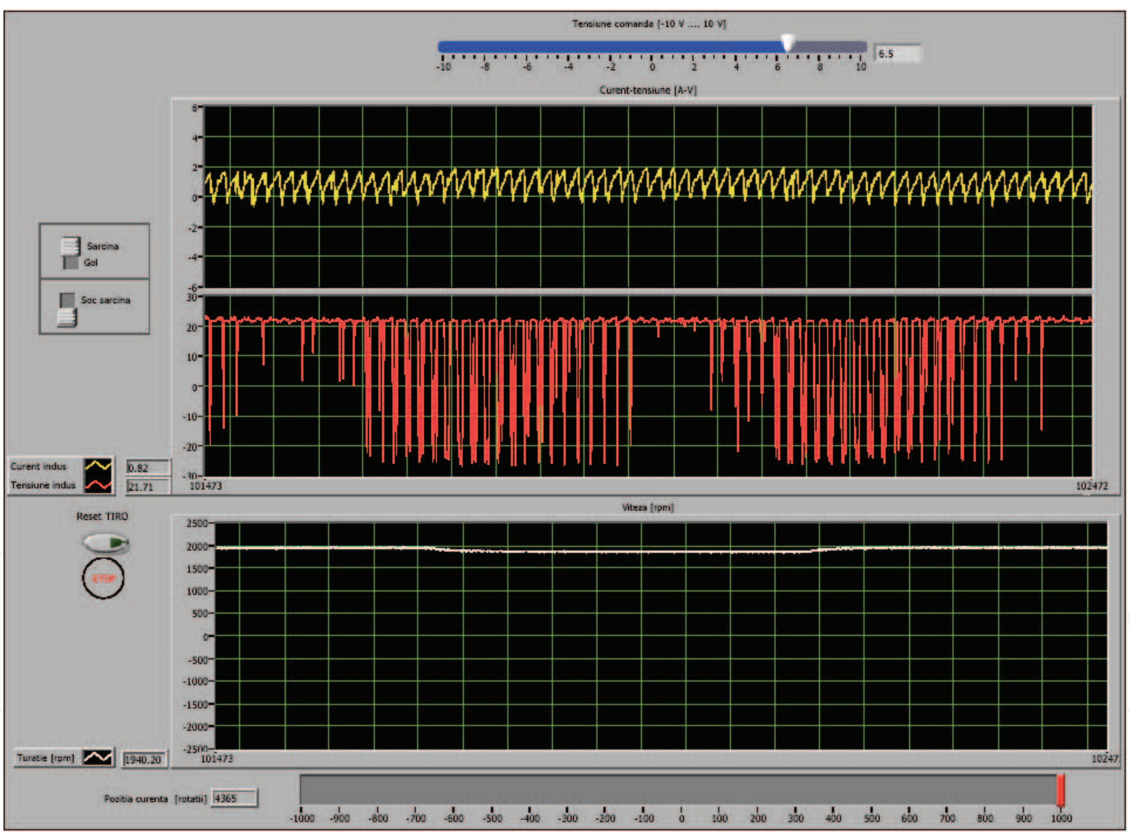


Fig. 14. Main panel for D.C. brush servomotor electrical drives system (open-loop operating mode).

It reveals the experimental results being obtained for the open-loop control of the electrical drives system when the user applies a load torque disturbance to the controlled servomotor shaft. The variations of the armature current and modulated voltage applied to its terminals (for a command signal of 6.5 V) are presented by means of the first virtual oscilloscope. The speed fall caused by the load torque disturbance can be noticed on the second virtual oscilloscope. Thus the user can deduce the causal link between the disturbance acting over the system and its effects over the output measure.

A more complex study can by achieved by means of the second application designed for this stand. This is the control of a positioning system driven by D.C. brush servomotor (the motion control algorithms from Fig. 11 and Fig. 12). The LabVIEW program developed this purpose is shown in Fig. 15. The same as for the application for the position control of the positioning system driven by stepper servomotor, the program is structured into two *while ()* loops (the control algorithm ensures the processing flow described by the same flowchart shown in Fig. 9). The novel element, as compared to the structure of positioning described earlier, is the position and speed controller offering multiple possibilities of configuration for the control topology. Thus, the user can choose to use either a classical structure of position control by means of a *PID controller*, or a *master-slave (cascade)* structure of P-position & PI-speed type. At the same time, either structure can be completed with control elements by *feedforward* in accordance with the speed and/or acceleration references of the trajectory generator. It is obvious that once the topology having been established, the user has to ensure the tuning of the control structure in view of obtaining the desired behavior performances.

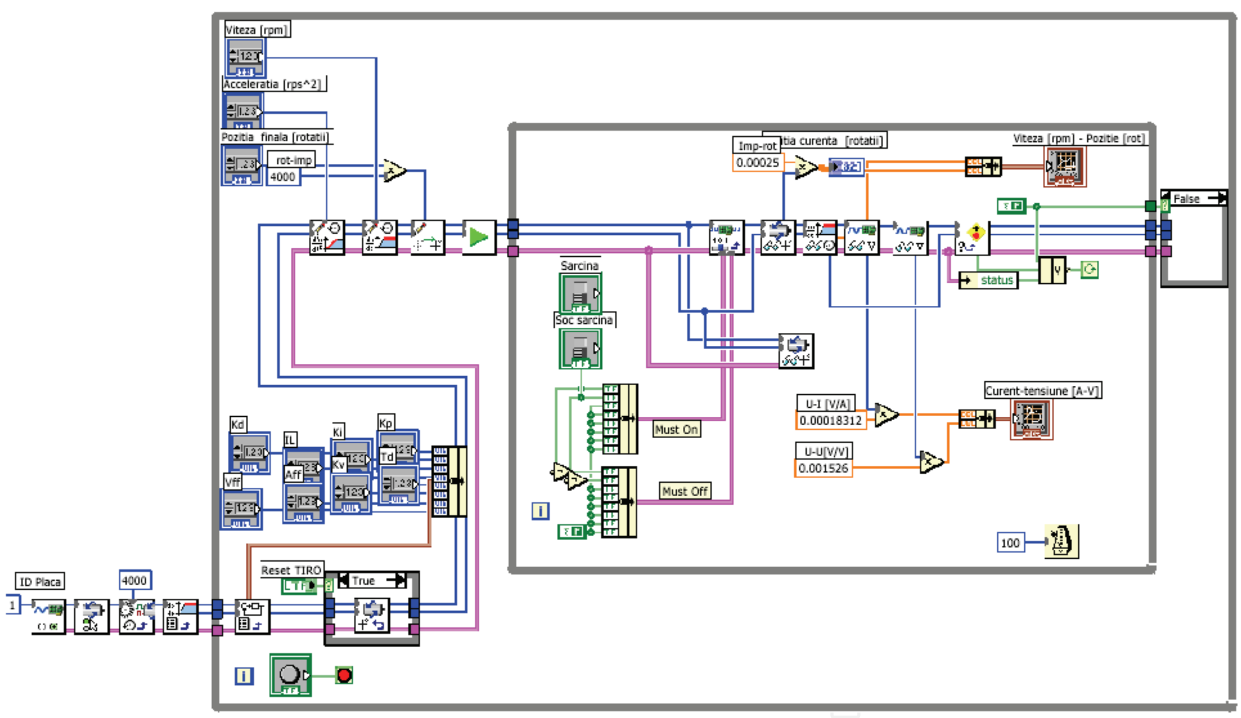


Fig. 15. LabVIEW diagram for single axes positioning system (closed-loop operating mode).

The user graphical interface (see Fig. 16) shows the experimental results obtained for a classical position control structured by using a PID controller. The tuning parameters of the control law are presented in the bottom-left area reserved for the controller settings and those related to the motion in the top-left part. The review of the results certifies the fact that the positioning within the range [0÷1000] rev. is completed in full accordance with the requirements imposed by the motion constraints and with optimal dynamic behavior performances that is little and well harmonized transitory regimes.

Through the graphical user interfaces, the students acquire the following abilities:

- enabling the motor and resetting the control system;
- calibrating a D.C. brush servomechanism;

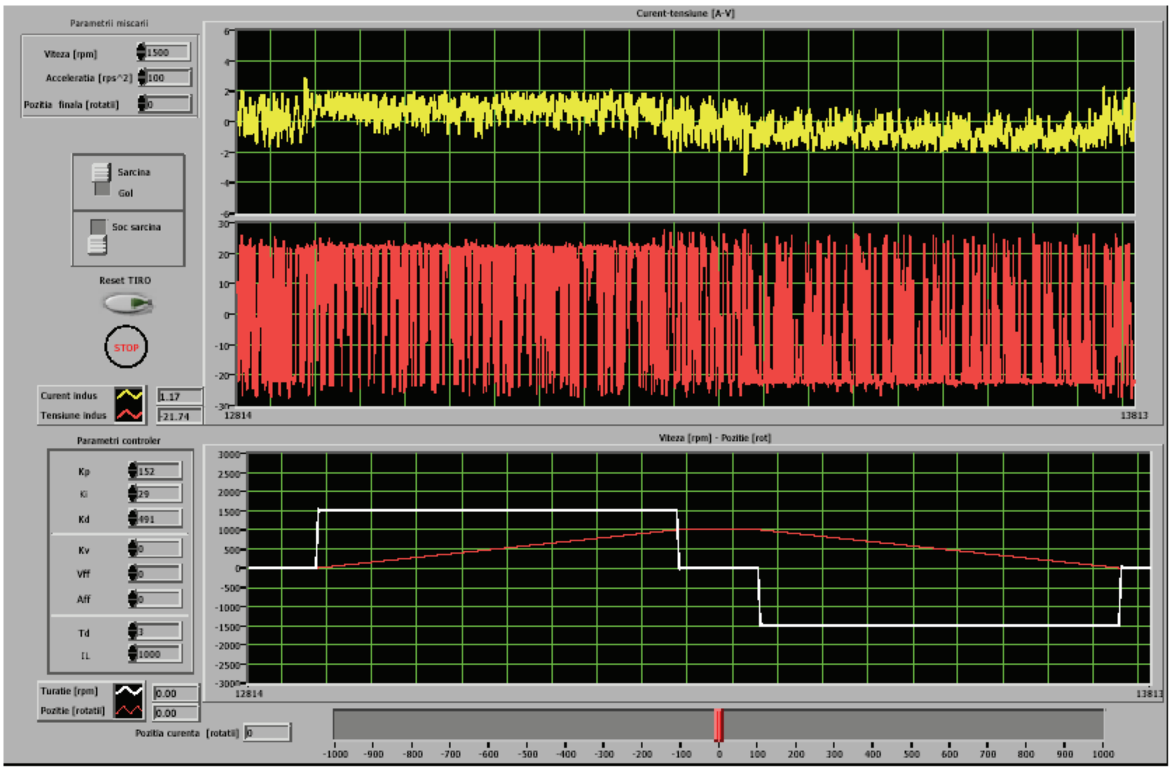


Fig. 16. Main panel for D.C. brush servomotor electrical drives system (closed-loop operating mode).

- controlling the rotation direction of the D.C. brush servomotor;
- moving the servomotor shaft from one arbitrary position to another;
- accelerating the servomotor and maintaining a constant velocity as well as decelerating and bringing it to a complete stop;
- visualizing intermediate motor control and encoder signals;
- understanding the relationship between the external loads and command effort.

3.3 NCS in the hierarchical structure: Stepper servomotor electrical drives system

In order to study positioning application for stepper servomotor electrical drives, the set-up, which is shown in Fig. 17 and has the layer from Fig. 1, is used.
The set-up consists of a stepper servomotor, an optical encoder, a controlled loading device and electronic boards for the delivery of control sequences.
The 2-phase stepper servomotor (made in Italy, type HY 200-2222-100 D 6) has the following features:

- step angle θ_p=1.8 degrees/step (200 steps/revolution);
- rated voltage supply U_n=5V;
- current per phase I_n=1A;
- resistance per phase R_f=5Ω;
- inductance per phase L_m=5.45mH;
- electrical time constant T_e=1.09ms;
- maximum synchronization torque M_{max}=0.3N·m.

The stepper servomotor is driven by two specialized integrated circuits (made by SGS-THOMSON, Microelectronics Company): L297 (stepper servomotor controller) and L298N (dual H-bridge driver). The servomotor's shaft is connected with an incremental optical encoder that gives 250 pulses/rev. (this means 1000 pulses/rev. at the output of the sense discriminator or the NI PCI-7354 controller). Among the features of the loading system, the following ones permit (Baluta et al., 1997), (Baluta, 2003):

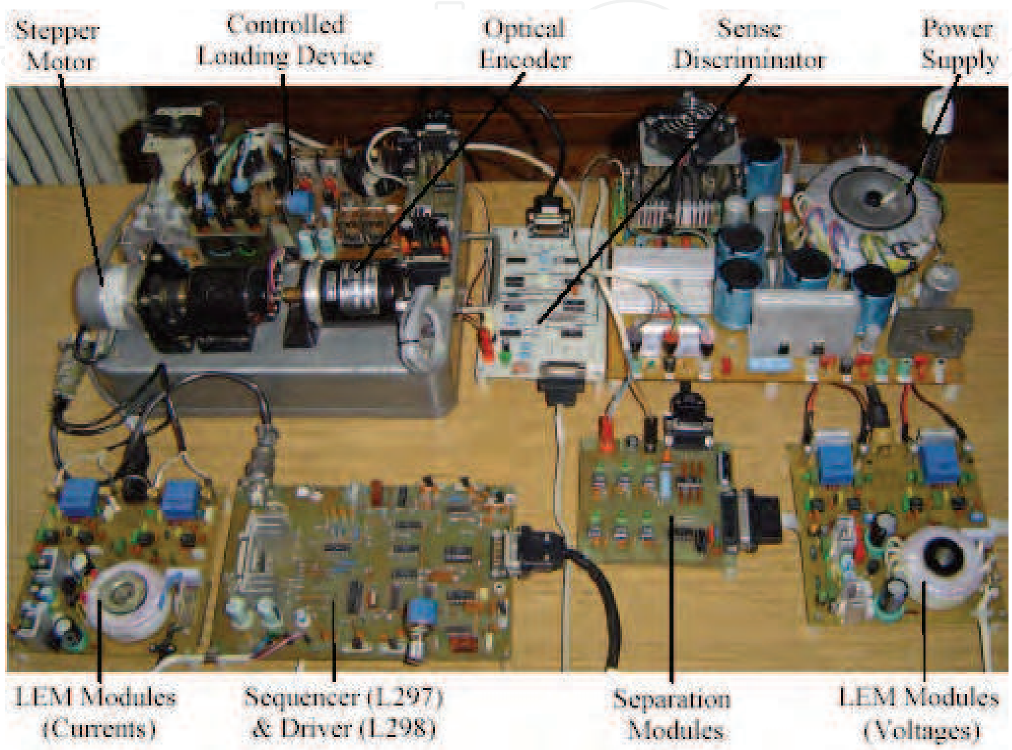


Fig. 17. Stepper motor electrical drives system.

- to load the electric drive servomotor with a reactive load torque in a wide range of speed, including very low speeds;
- to impose a constant load torque operation mode or an overload operation mode;
- to prescribe, analogically or numerically, the amplitude and time length of the load torque.

The driven axes with stepper servomotors can be controlled in open-loop using constant frequency or prescribed profile (trapezoidal or S-curve move profile) or closed-loop. In the second case, the user has to initialise the feedback for position and speed variables and to adequately set the control block of the pulse generation for stepper servomotor command. The operating mode of the stepper servomotor can be with full step, half step or microstepping mode. The main function of a motion controller is to assure the desired motion trajectories.

The implemented control system uses the algorithm from Fig. 9, which allows to generate in real-time the move trajectory and to change the motion parameters. In order to program the motion control system, the user employs LabVIEW environment with specific virtual instruments for motion control from the library FlexMotion. The graphic program associated is shown in Fig. 18 (Baluta, 2007).

The processing flow is structured into two loops of *while ()* type. The inner loop, executed for a time period of 1 ms is used to update the temporal trajectories of the measures of interest: electric (phase currents and phase supply voltages) and mechanic (position and speed). The leaving off of the inner loop done as the prescribed position is attained or the follow up speed exceeds a prescribed level. The control outer loop is executed with a period of 1 s and allows or one hand the configuration of the application in relation with the parameters of the drives system (number of stepper steps, number of pulses optical encoder) and on the other hand the definition of the operating regimes and modes and the motion parameters (full steps/half step, open-loop/closed-loop, final position, maximum speed, maximum acceleration and maximum deceleration).

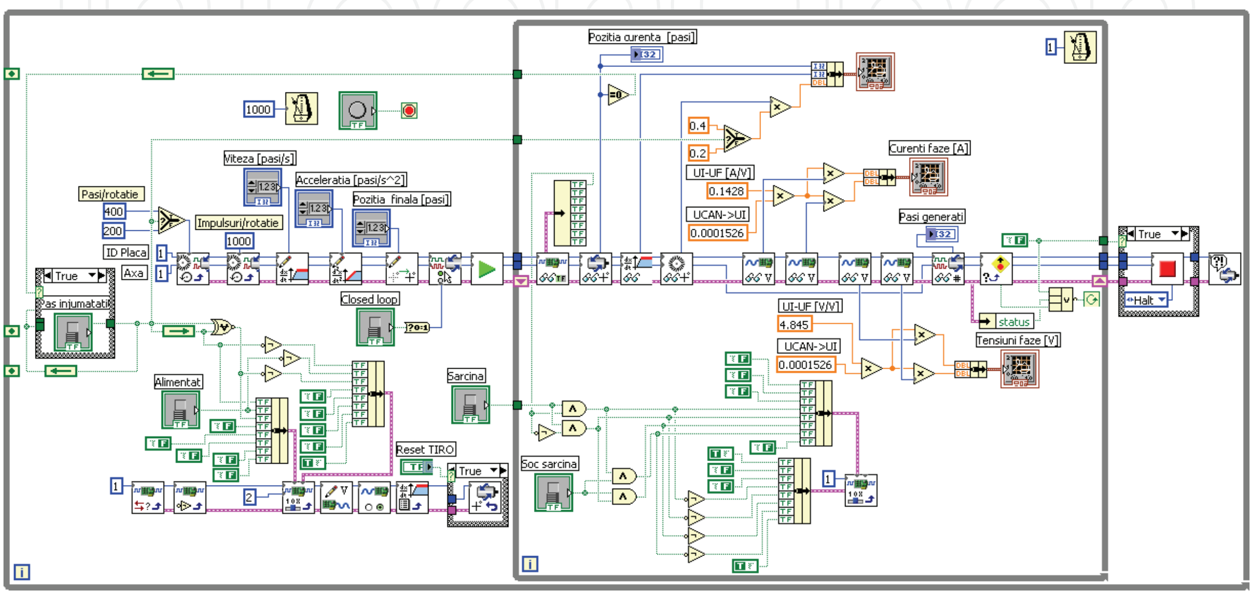


Fig. 18. LabVIEW diagram for single axes positioning system.

The application calculates the error between the prescribed position and real position and, when this error is other than zero, the inner loop is executed until the position error becomes of no value. As one can notice, the program provides two types of configuration:

- implicit configurations, not accessible to user (number of stepper steps, number of pulses/rev. of optical encoder);
- explicit configurations available in the user graphical interface (full steps/half step, open-loop/closed -loop, servomotor loading, position initialisation).

Such configurations are basically achieved when the main loop is executed, therefore after the achieved a motion programmed profile. Nevertheless, in order to ensure to user the possibility of applying an overload when achieving the motion profile, the command of application of the overload was included in the control inner loop.

When the controlling mode of the stepper servomotor (full step/half step) is modified, the settings related of the number of steps that the servomotor is executing during a full revolution must be correlated as well. This is automatically achieved in a transparent manner from the point of view of the user.

The application can be stopped by an explicit control through the *Stop* push-button, involves the leaving off the main loop of the program.

In order to build a positioning application, the developer has to follow the stages shown in Fig. 13, which represent the steps required to design a motion application.

The trajectory generator takes into account the type and the constraints of motion and generates points or instantaneous position in real-time. The control loop converts each instantaneous position to a voltage or to step-and-direction signal, depending on the type of motor is used. Motion constraints are the maximum velocity, acceleration, deceleration and jerk that the system can handle. The trajectory generator creates a velocity profile based on these motion constraints values. Trajectory parameters are expressed as a function of steps numbers. In this case the number of steps depends on the type of the stepper drive and the servomotor that is used.

Basically, the student can learn about the parameters of the stepper servomotor, by changing modes and input settings to make execute the operations, i.e., change the rotation direction, position and speed. Also during the experiment, the student can visualise critical input and output points in the stepper servomotor electronic interface using virtual oscilloscopes. Since this system is in real-time mode, the learner observes changes in position, direction, and velocity as the motor rotates after being stepped through with various frequencies corresponding to increasing or decreasing variations of steps/sec.

The graphical user interfaces for a positioning application of stepper servomotors are presented in:

- Fig. 19 for open-loop control and full step operating mode;
- Fig. 20 for closed-loop control and half step operating mode.

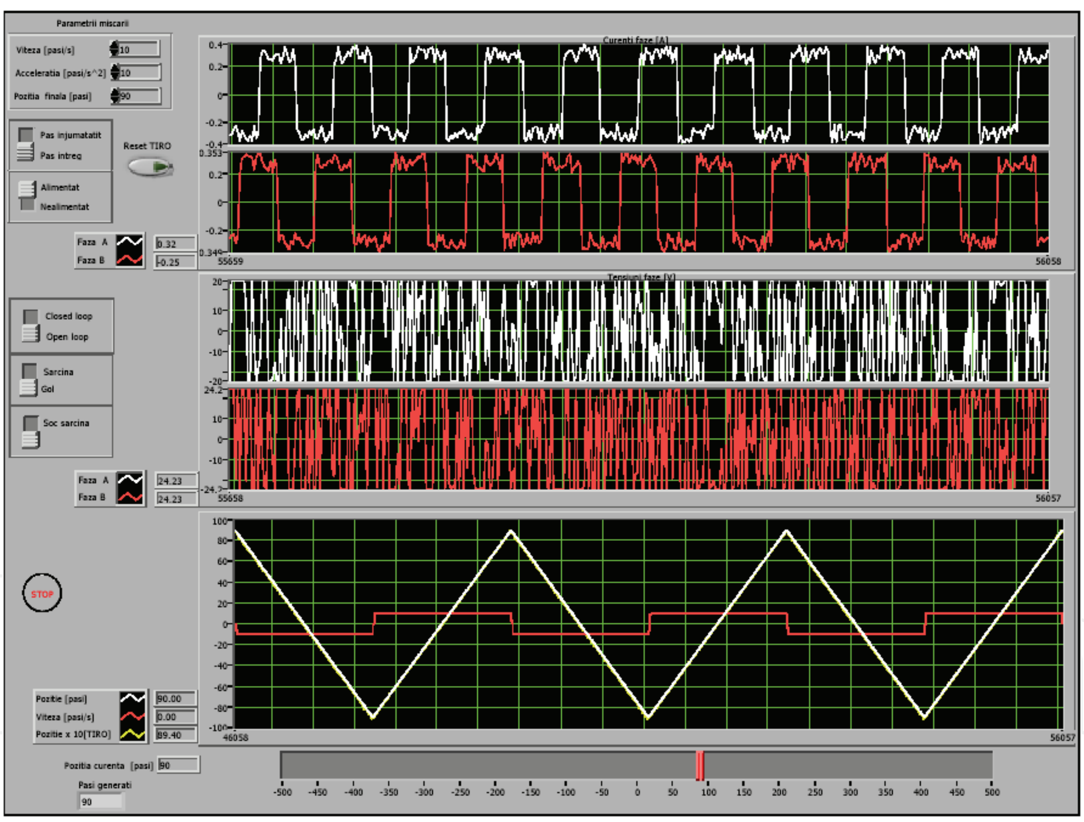


Fig. 19. Main panel for stepper servomotor electrical drives system (open-loop control and full step operating mode).

The graphical interface includes three virtual oscilloscopes allowing the view of the phase currents through the servomotors windings, voltages applied to windings and trajectories of the mechanical measures (prescribed position, measured position, prescribed speed). Additionally, the real position is presented dynamically, by means of a graphical slider too.

The instantaneous numerical values of the above measures are also presented using virtual display elements.

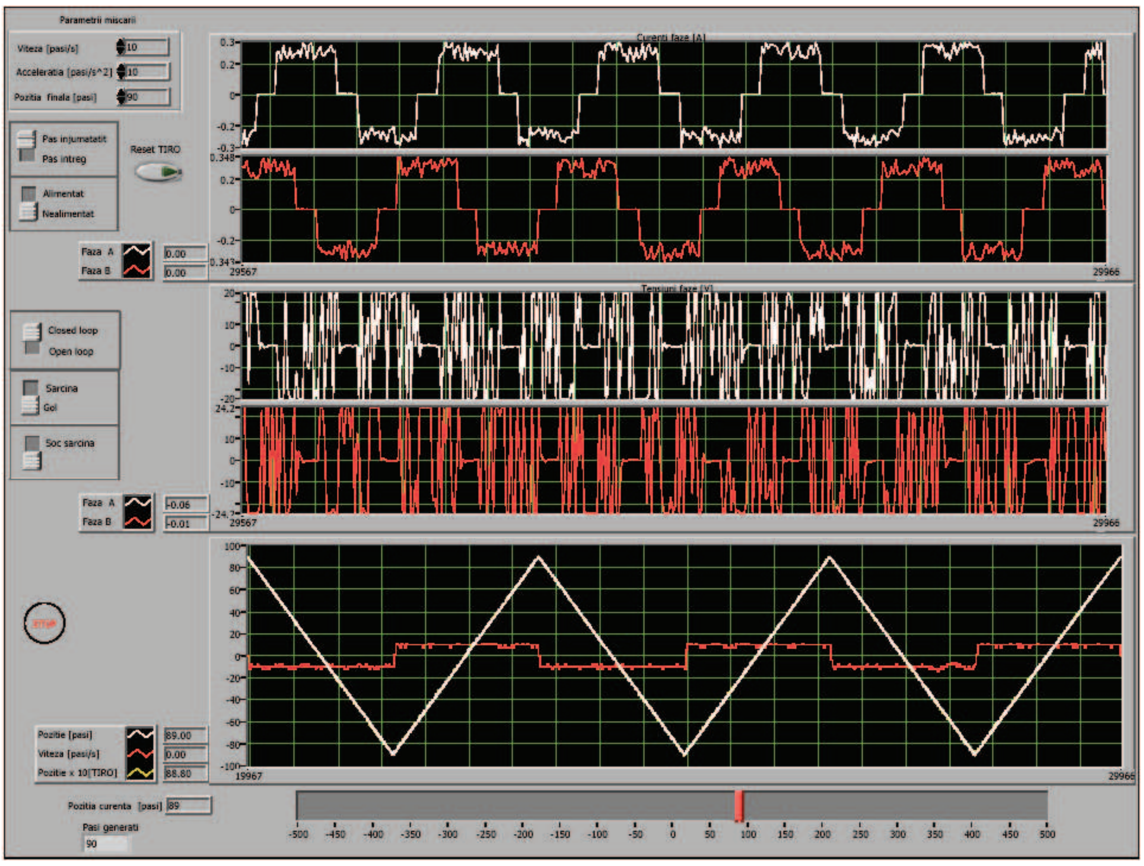


Fig. 20. Main panel for stepper servomotor electrical drives system (closed-loop control and half step operating mode).

The graphical interface offers to the user certain switches, in order to establish the desired operating mode:

- the initialization of the position register of the positioning system is achieved by means of the *Reset TIRO* push-button;
- the *Powered/Unpowered* push-button provides a logical command of activating on the stepper servomotor driver;
- the *Full Step/Half Step* push-button defines the stepper servomotor command mode (full step or half step);
- the positioning system loading is achieved by means of a controlled loading device being controlled through the *Load/Unload* and *Overload* push-buttons.

The parameters and constraints on the motion are defined by means of the introductions elements of numerical values (speed, acceleration, position). After such measures are programmed, the supervision system transfers such measures to the trajectory generator, which, in its turn, supplies suitable references to the control loops. Thus, the user has the possibility to program dynamically the motion parameters, the new values being considered immediately after the completion of the motion current cycle, previously programmed. Fig. 19 shows the experimental results obtained by an *open-loop* control of the position of the positioning system between the extremes [-90÷+90] steps with the maximum speed of 10 steps/s and acceleration/deceleration of 10 steps/s^2. The servomotor command mode is with full step, the positioning system being unloaded.

The temporal trajectories of the electrical and mechanical measures, obtained for three positioning cycles are shown of the virtual oscilloscopes of the user graphical interface. The user has thus the possibility to correlate the command and control measures with the operating regimes and loading modes of the positioning system. Fig. 20 shows, for instance, the operation regime of the positioning system in *closed- loop*. By a comparative study, one can notice that in such case, the reaction measures are easily contaminated by noise. In spite of that, the control block ensures the desired parameters of the motion and guarantees the execution of the positioning without risking of loosing steps.

Through the user interface, the students acquire the following abilities:

- calibrating a stepper servomotor system;
- controlling the rotation direction of the stepper servomotor;
- enabling the motor and resetting the control system;
- understanding the relationship between a step and the number of steps required making one complete 360-degree revolution;
- moving the motor shaft from one arbitrary position to another;
- accelerating the motor and maintaining a constant velocity as well as decelerating and bringing it to a complete stop;
- visualizing intermediate motor control and encoder signals.

The implemented control structure for stepper motors permits the user to get familiarized with solving problems for positioning applications.

4. Conclusions

In this chapter, a remote control laboratory for electrical drive systems is presented. The laboratory allows the students to develop network based control systems using an architecture based on I/O devices, communication modules and server-client application implemented with Lookout and LabVIEW environment facilities and to operate on real electrical drive systems through Intranet and Internet.

Three examples were presented demonstrating the potentiality of the networked control system laboratory to remotely control the electrical drive systems and to develop new network based control system structure.

Work is in progress to upgrade the laboratory with new electrical drive systems: D.C. brushless servomotor electrical drives system.

5. References

Antsaklis, P.J. & Baillieul, J. (2004). Guest Editorial: Special Issue on Networked Control Systems, *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, (September 2004), pp. 1421-1423, ISSN: 0018-9286.

Baillieul, J. & Antsaklis, P.J. (2007). Control and Communication Challenges in Networked Real-Time Systems, *Proceedings of the IEEE*, Vol. 95, No. 1, (January 2007), pp. 9-28, ISSN: 0018-9219.

Baluta, Gh. (2003). *Electrical Drives with Stepper Motors* (in Romanian), Gh. Asachi Publishing House, ISBN: 973-621-034-0, Iasi, Romania.

Baluta, Gh. (2004). *Low Power Electrical Drives. Applications* (in Romanian), Politehnium Publishing House, ISBN: 973-621-072-3, Iasi, Romania.

Baluta, Gh. & Lazar, C. (2007). Remote Control Laboratory Development for Electrical Drive Systems, *Proceedings of the 13-th IEEE/IFAC International Conference on Methods and*

Models in Automation and Robotics, pp. 393–398, ISBN: 978-83-751803-3-6, IEEE Conference Number: 12459, Szczecin, Poland, 27-30 August 2007.

Bauer, P. & Fedak, V. (2003). E-learning for Power Electronics and Electrical Drives, *Proceedings of the 13-th International Conference on Electrical Drives and Power Electronics*, pp. 567-572, ISBN: 80-89061-77-X, High Tatras, Slovak Republic, 24-26 September 2003.

Bushnell, L.G. (2001). Special Section on Networks and Control, *IEEE Control Systems Magazine*, Vol. 21, No. 1, (February 2001), pp. 24-99, ISSN: 0272-1708.

Carari, S.; Chirea, O. & Lazar, C. (2003). Distributed control system for educational training based on Lookout and FieldPoint, *Proceedings of the NIWeek Worldwide Virtual Instrumentation Conference*, Austin, USA, 13-15 August 2003.

Casini, M.; Prattichizzo, D. & Vicino, A. (2004). The Automatic Control Telelab: A Web-based technology for distance learning, *IEEE Control Systems Magazine*, Vol. 24, No. 3, (June 2001), pp. 36-44, ISSN: 0272-1708.

Ellis, G. & Lorentz, R.D. (1999). Comparison of Motion Control Loops for Industrial Applications, *Proceedings of the IEEE IAS Annual Technical Conference*, Vol. 4, pp. 2599-2605, ISBN: 0-7803-5589-X, Phoenix, AZ, USA, 3-7 October 1999.

Fedok, V. & Bauer, P. (2005). E-learning in Education of Electrical Drives and Power Electronics: Opportunities and Challenges, *Proceedings of the 15-th International Conference on Electrical Drives and Power Electronics*, pp. 1-9 (paper E05-L4.pdf), ISBN: 953-6037-43-2, Dubrovnik, Croatia, 26-28 September 2005.

Kreindler, L. (2002). DSP Solution for Digital Motion Control, *Journal of Electrical Engineering*, Vol. 2, (June 2002), pp. 1-9 (paper Kreindler.pdf), ISSN: 1582-4594.

Lian, F.L.; Moyne, J.R. & Tilbury, D.M. (2002). Network Design Consideration for Distributed Control System, *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 2, (March 2002), pp. 297-307, ISSN: 1063-6536.

Lazar, C. & Carari, S. (2008). A Remote Control Engineering Laboratory, *IEEE Transactions on Industrial Electronics*, in press, ISSN: 0278-0046.

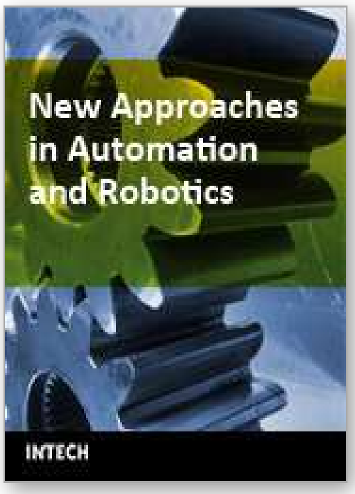
Saad, M.; Saliah-Hassane, H.; Hassan, H.; El-Guetioui, Z. & Cheriet, M. (2001). A Synchronous Remote Accessing Control Laboratory on the Internet, *Proceedings of the International Conference on Engineering Education*, Session 8D1, pp. 8D1_30-8D1_33 (paper 321.pdf), ISBN: 1-588-74-091-9, Oslo, Norway, 6-10 August 2001.

Saliah, H.H.; Nurse, E. & Abecassis, A. (1999). Design of a Generic, Interactive, Virtual and Remote Electrical Engineering Laboratory, *Proceedings of the 29-th ASEE/IEEE Frontiers in Education Conference*, Session 12c6, pp. 12c6_18-12c6_23 (paper 1365.pdf), ISBN: 0-7803-5643-8, San Juan, Puerto Rico, 10-13 November 1999.

Tipsuwan, Y. & Chow, M.Y. (2003). Control methodologies in networked control systems. *Control Engineering Practice,* Vol. 11, No. 10, (October 2003), pp. 1099–1111, ISSN: 0967-0661.

Valera, A.; Diez, J.L.; Valles, M. & Albertos, P. (2005). Virtual and Remote Control Laboratory Development, *IEEE Control Systems Magazine*, Vol. 25, No. 1, (February 2005), pp. 35-39, ISSN: 0272-1708.

Yang, T.C. (2006). Networked control system: a brief survey, *IEE Proceedings of Control Theory and Applications*, Vol. 153, No. 4, (10 July 2006), pp. 403-412, ISSN: 1350-2379.



New Approaches in Automation and Robotics
Edited by Harald Aschemann

ISBN 978-3-902613-26-4
Hard cover, 392 pages
Publisher I-Tech Education and Publishing
Published online 01, May, 2008
Published in print edition May, 2008