

# An Internet-Assisted Experimental Environment Suitable for the Reinforcement of Undergraduate Teaching of Advanced Control Techniques

Ahmet Bugra Koku, *Student Member, IEEE*, and Okyay Kaynak, *Senior Member, IEEE*

**Abstract**—The development of an experimental environment with interactive features, suitable both for the reinforcement of undergraduate level teaching of control theory and also for graduate-level research work in the field of intelligent control is described. The students are prepared for the experiment by an interactive session on the Internet before coming into the laboratory. The hardware and the software components of the experiment are integrated to achieve maximum effectiveness in teaching advanced control concepts, such as fuzzy and neuro-fuzzy control. The environment is also suitable for masters-level research work.

**Index Terms**—Matlab, soft computing, teaching control systems, undergraduate teaching.

## I. INTRODUCTION

**E**XPERIMENTAL study is a very important component in engineering education. It not only acts as a bridge between theory and practice but also solidifies the theoretical concepts presented in the classroom [1]. In the classical approach to laboratory work, students are often prepared to an experiment theoretically in classroom or some handouts may be presented to them for reading prior to a laboratory session. Equipped with a brief theoretical knowledge acquired in such a way, the student either observes a predesigned experiment throughout its course of action or at most modifies some system parameters to observe their effects. This classical way of experimenting clearly has the following shortcomings.

- The classroom lectures or the handouts are generally not sufficient for a student to be fully prepared for a hands-on experiment or to appreciate the significance of the previously explained theory in the experiment to be performed.
- The percentage of student interaction with the experimental system is a measure of what the student gains from an experiment. When the student is a passive observer or a semiactive part of an experiment, s/he will understand neither the correspondence nor the difference between theory and practice.

An experimental setup that activates the student to a higher intellectual degree is therefore essential in increasing the conceptual reinforcement of control system theory.

The recent technological trends in engineering education have evolved around the use of computers, software tools, and multimedia presentations. The use of software simulation tools in place of the prohibitively expensive laboratory hardware has become an integral part of modern curriculum, because computers are able to provide unique visualization and animation capabilities that help in the presentation of abstract subjects and dynamic phenomena. However, simulation studies suffer from the shortcoming that the model used in place of a physical plant often falls short of fully portraying the intricacies of real systems. An environment that enables the student to experiment freely with physical hardware and at the same time utilizes the capabilities of computers and the software would therefore be very desirable. This paper reports the development of such an environment, called the “Experimental Garden.”

The technological developments of the recent decade have increased the use of computers in experimentation and in real-world control applications. It is now possible and economically feasible to use complex, model-based control paradigms in practical applications, using advanced strategies derived from adaptive, nonlinear, and robust control theories. This has resulted in the development of the “intelligent control” field and a host of new control approaches based on fuzzy logic, neural networks, evolutionary computing and other techniques adapted from artificial intelligence. The introduction of such concepts at the undergraduate level is now not uncommon. This has been taken into consideration in the development of the experimental garden. It is designed in such a way that it can be used by undergraduate students for the reinforcement of these advanced methodologies as well as by the graduate level ones for masters degree research studies. This naturally raises the question of availability. In order to decrease the time that the students have to spend in the laboratory with the hardware, an informative web site has been prepared by two undergraduate students as a part of their final year projects ([http://mecha.ee.boun.edu.tr/lab/X\\_Garden](http://mecha.ee.boun.edu.tr/lab/X_Garden)). Students that are going to use the experimental garden are requested to first visit the web site through the campus-wide fiber-optic network and get themselves acquainted with the hardware and software structures of the experimental setup. A graduate student has constructed the Experimental Garden itself as a part of his M.Sc. thesis work and provided the necessary software environment and hardware-software cooperation. Its details are described in the next section.

Manuscript received March 2, 1998; revised November 22, 1999.

A. B. Koku is with Intelligent Robotics Laboratory, Vanderbilt University, Nashville, TN 37203 USA.

O. Kaynak is with the Department of Electrical and Electronic Engineering, Bogazici University, Bebek, Istanbul, Turkey.

Publisher Item Identifier S 0018-9359(01)01255-9.

## II. STRUCTURE OF THE EXPERIMENTAL GARDEN

In dealing with physical systems, the following notions have to be appreciated and fully grasped by every prospective engineer:

- nonideal structure of physical world (e.g., friction);
- presence of noise and nonlinearity at different levels (e.g., backlash);
- physical constraints (e.g., limited displacement range);
- importance of physical construction (e.g., reliable cabling);
- protection against probable malfunctioning;
- appropriate interfacing of different parts to get a “working” system.

A full grasp of above topics cannot be obtained in classroom or through simulation studies but requires hands-on experimental experience. The Experimental Garden designed is aimed at providing this experience through a multifunctional and open-ended architecture [2]. It is composed of the following main parts, as illustrated in Fig. 1:

- a PC with RTI (real-time interface) installed;
- amplifying unit;
- variety of plants.

These parts constitute the hardware structure of the garden. The PC is an HP 486 on which a real-time DSP card procured from dSpace (DS1102) is installed. Different amplifying units are present depending on the plant used in the loop. The plants that are presently integrated for usage are as follows:

- inverted pendulums (two different types);
- linear position servo;
- ball and the beam;
- rotary position servo;
- seesaw.

The garden is open-ended for future development; new plants can easily be added to it.

As it is stated earlier, the use of simulation software in place of physical apparatus is becoming increasingly common in undergraduate laboratories and a number of products are available in the market for this purpose. One of the commonly used ones, especially in controls [3] and signal processing [4], is Matlab, either used alone or with a graphical front end; Simulink. Since these packages were available at the EE Department of Bogazici University too, the software environment of the Experimental Garden is constructed basically around them at the design phase of a control application and on the software provided by dSpace (Cockpit and Trace) for its implementation. The general structure of software interaction and hierarchy is shown in Fig. 2 [5]. First of all, a control system is designed by the use of Simulink blocks. This may be either a simulation or an application file. After the design is complete, the Real-Time Workshop (RTW) converts the Simulink file into C code, compiles it and then downloads the compiled model onto the DSP of the real-time card. Following a successful download, the Cockpit and Trace software provide the user a medium through which s/he can both view or change the value of any parameter of the system. This online interaction with a presently running application is very helpful in observing the effects of various parameters by dynamically changing them.

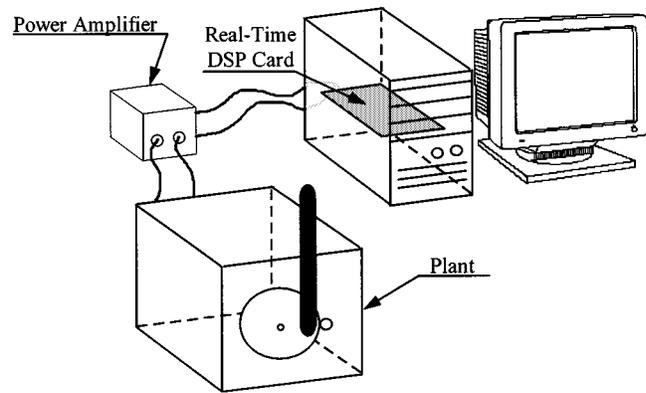


Fig. 1. Schematic drawing of the Experimental Garden.

The RTW also allows the embedding of user developed C codes into the Simulink design for advanced applications. This makes the realization of very complex applications possible.

## III. THE USE OF THE EXPERIMENTAL GARDEN BY STUDENTS

In designing the experimental garden, an important criteria has been to make it as simple as possible for the beginners. However, it is also desired that it enable the experienced users to experiment with advanced “intelligent control” techniques. In this section, how a newcomer student is guided through the usage of the garden from simple experiments to those with advanced fuzzy-neuro algorithms is described.

A student with no previous information about the system is first invited to visit the web site previously stated. This site contains detailed information ranging from the descriptions of individual plants and how appropriate cabling should be done for the correct operation of a chosen plant, to how the software is to be used during the experiments. This visit prepares the student for the subsequent hands-on work on the real system.

The second stage of experimentation commences with hands-on learning. A Matlab menu file routes the student through a number of previously designed simulation studies and real-time applications. A wide range of control algorithms and real-time applications are available on the menu, ranging from open-loop and PID controls to fuzzy and adaptive fuzzy applications. By going through these, a student gets a chance to observe the factors involved in designing a control system from scratch.

As students get more experienced, they can start designing experiments themselves. In order to help them in their design, some Simulink template blocks, Matlab and C codes are provided as a library.

The simplest way to have a walk around the experimental garden is to use the interactive Matlab menu mentioned above. It is very similar to the “demo” program designed to introduce the general structure of Matlab and Simulink. While navigating through the garden using this menu, both simulations and real-time applications can be reached. The student is thus able to observe the difference between a simulation study and a real-time application. By running the simulations and the applications simultaneously, the additional factors that have to be taken into consideration in designing and running real-time

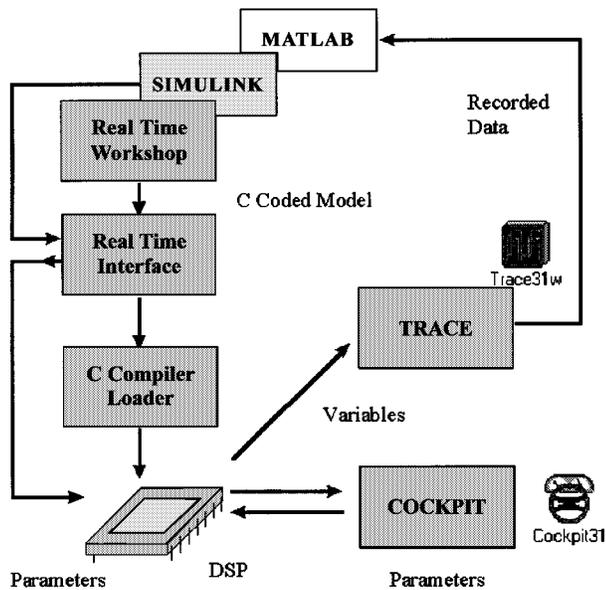


Fig. 2. The compilation and downloading process to download a Simulink control design to a real-time application [5].

control systems can be observed. Considering the fact that some simulations with complex algorithms can take a long time to simulate, their results are saved in data files for different parameter values and, by a simple click on the Matlab menu, these previously saved simulation results can be observed without losing time.

Since the main goal of the Experimental Garden is to facilitate experimental implementation, the main volume of the interactive software is devoted to guide the students through an actual implementation. The control algorithms studied through simulations can all be applied to any plant of the garden. When a student chooses a particular plant to experiment upon, an introduction comes up on the screen. It has two goals. First, it introduces the general structure of the RTW to the student. A very simple Simulink block, which performs no control action, is downloaded onto the microprocessor of the DSP. During the course of this process, the student is expected to understand the operational structure of the RTI, which is shown in Fig. 2.

The second goal of the introduction is to allow the user/student to learn how the system works. No control action is applied to the system at this stage. For example, if the inverted pendulum is the chosen plant, the student plays around with the mechanical system (the pendulum cart or the pole) and observes the changes in the values of the states on screen. The student is thus introduced to the concept of normalization, which is very important in some of the plants where sensed states are not absolute.

After the introduction, the student has sufficient knowledge to download an existing application and normalize the system parameters. It is now time to experiment with an open-loop control scheme and see how the system behaves with respect to control input. A control voltage in the form of a sine or square wave can be applied to the plant. It is also possible to apply control signals manually using a slider. Fig. 3 illustrates such an interface. This experience is especially helpful in designing classical fuzzy controllers. Some practice on open-loop control provides

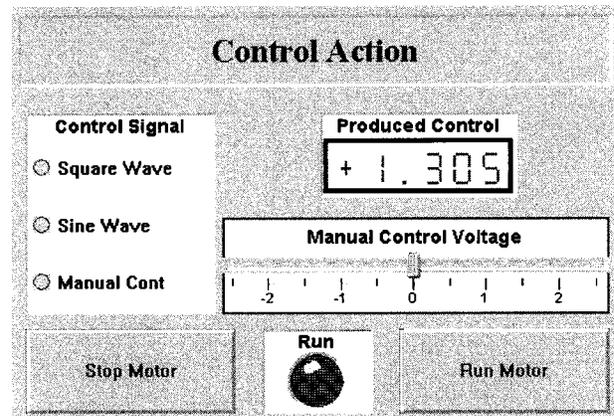


Fig. 3. The open-loop control interface.

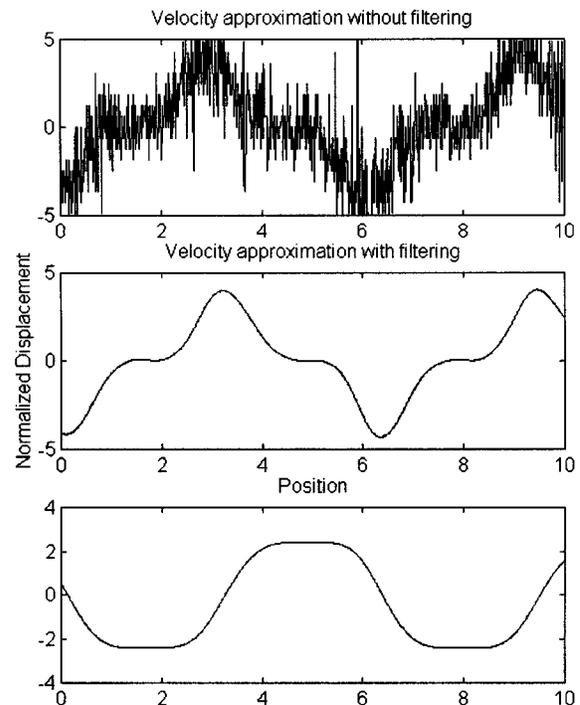


Fig. 4. A noisy and clear cart velocity estimation taken from linear position servo system with the illustrated position profile.

a measure on the meanings of fuzzy adjectives used in the fuzzy control rules (i.e., small, large, fast, or slow).

Experimenting with a PID controller generally constitutes the third stage in the walk through the garden. Following a successful download, the student experiences the first closed-loop application. The reference and the actual trajectories, the errors and other states of the system can be observed by using the Trace software. The parameters of the control system such as gains, digital filter parameters, and more specifically any parameter of a Simulink block used can be both observed and changed while the application is running by using the Cockpit software. This step is not only designed to clarify the concepts of feedback control, but also to enable the student to understand the individual effects of P, I, and D terms by changing their values and

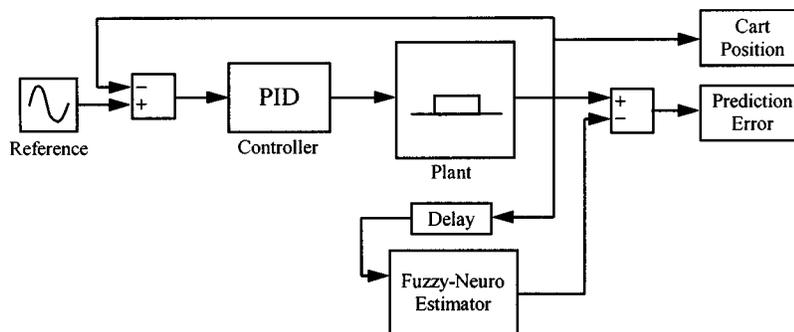


Fig. 5. The control system designed for position estimation.

observing the corresponding response throughout the real-time application.

In the final step of the learning stage, the concept of “noise” is demonstrated. Generally potentiometers are used as sensors and a derivative block approximates the time derivative of a state measured from a potentiometer. Observing the time derivatives of two signals, one being filtered before the derivative block and the other not, helps the student to understand the importance of filtering, the effect of noise and the amplifying effect of derivation on a noisy signal. After downloading the noise demonstration file onto the DSP, the student will observe a plot similar to the one illustrated in Fig. 4, while running a real-time PID control on linear position servo.

At the completion of the steps described above, a student is expected to have attained an adequate understanding of how the closed-loop control systems (previously) designed using Simulink blocks are converted into real-time applications by using the RTW. The student at this stage is now ready to make her/his own design. Therefore, designing a controller from scratch constitutes the third stage of progress in the garden. The students are urged to go through this stage too, designing a controller of their choice using the Simulink blocks available or creating their own blocks and a corresponding controller interface in Cockpit.

The procedure explained above is sufficient for undergraduate students to gain experience on real-time control system design. As mentioned before, not only do they experiment with basic control concepts, but also they get demonstrations on advanced algorithms (such as a fuzzy-neuro position estimator) within the span of the Matlab menu. More advanced algorithms are accomplished by embedding C codes into Simulink blocks. For advanced users, there are some examples in the garden to guide them in building their own controllers. It is also possible to adapt existing controllers to novel applications (e.g., changing the knowledge and rule base of the fuzzy controllers). The fuzzy-neuro position estimator is such an example. It is based on the current control signal applied to the system and it tries to predict how much displacement it will result at the end of the sampling period. The main purpose of this example is to demonstrate the adaptability of the fuzzy-neuro algorithm used [6]. It basically relies on fuzzy logic and defines a fuzzy network structure that can be trained by using backpropagation. To explain this algorithm and to demonstrate its capabilities, the above mentioned position prediction experiment is included in

the Matlab menu. A modified version of this algorithm is also applied to position control [2], [7]. In the experiment, a PID control is applied to the system, which provides a continuous motion on the track, and the estimator block tries to estimate the position of the cart at the next sampling interval. The block diagram of the system is given in Fig. 5. The Simulink file shown in the figure is downloaded onto the DSP and the algorithm is run. Until the end of seventh second, no control action is applied to the system, and at this instant, PID controller starts to act. The first graph in Fig. 6 shows the position profile of the cart. The second graph in the same figure illustrates both the actual and the estimated values for the change of position in one sampling interval. The last graph of Fig. 6 depicts the error occurred in estimating this displacement. The system is run without learning for approximately 10 s, and then the learning is introduced to the system. As can be seen from the figure, the estimated value catches and follows the actual value, soon after the incorporation of learning into the system.

The study of the example on the position estimator teaches the student how C codes can be embedded into Simulink designs to perform complicated tasks that are not realizable by using only the blocks available within Simulink. However, this example is also important in demonstrating the basic difficulties in building fuzzy logic controllers. Converting qualitative linguistic rules to quantitative control actions is not a very easy process, which a student should be aware of.

The walk through the garden described above constitutes a short but basic route. It involves two main phases: getting acquainted with the basics of the garden and the implementation on the garden. A student, after completing the introductory phase and thus grasping the structure of the garden and the main concepts of feedback control, is able to produce her/his novel structures and codes, simply by analyzing the present existing codes. Currently, two groups of students are studying sliding mode and fuzzy-neuro control structures on the garden.

#### IV. CONCLUSION

A recently developed, state-of-the-art experimental environment is described in this paper. Although its main objective is the reinforcement of the teaching of basic feedback control concepts, it is also suitable for use as an experimental test bed for graduate research activities, especially in the field of intelligent control. Currently, two graduate level research projects are

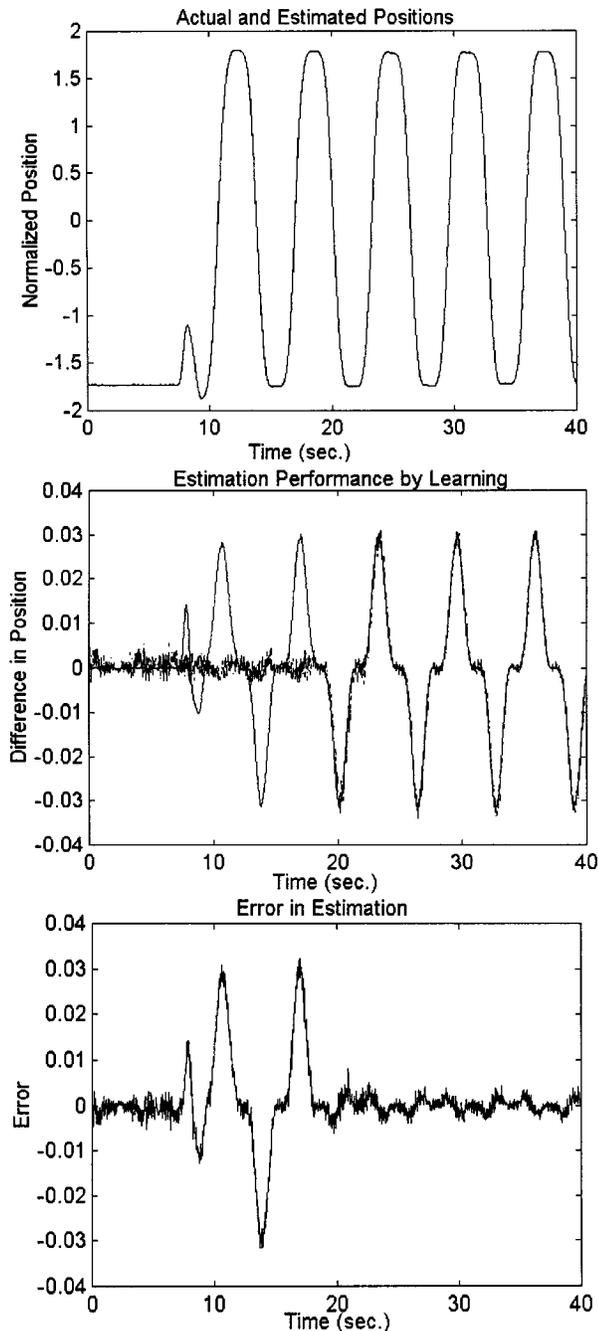


Fig. 6. Real-time position estimation data collected from linear position servo.

being conducted on this environment on the topics of sliding mode and fuzzy-neuro control.

The experimental garden has extensive interactive features. A student can therefore go through a laboratory session at his/her

own pace without requiring the presence and the assistance of a laboratory supervisor.

In order to improve the didactic effectiveness of the garden, students who work on the garden are requested to provide feedback by completing a questionnaire that is available on the web. The initial reactions are very positive. It is hoped that in time the garden will evolve into a very useful and attractive platform for students.

#### REFERENCES

- [1] J. Darnell and P. S. Harold, "Using computer simulations as a bridge between the classroom and laboratory," *Int. J. Elect. Eng. Educ.*, vol. 33, pp. 18–26, 1996.
- [2] A. B. Koku, "Fusion of soft computing techniques in control applications," M.Sc. thesis, Syst. Contr. Eng. Bogaziçi Univ., Istanbul, Turkey, 1997.
- [3] D. Xue, A. Goucem, and D. P. Atherton, "A menu interface to PC-Matlab for a first course on control systems," *Int. J. Elect. Eng. Educ.*, vol. 28, pp. 21–33, 1991.
- [4] R. W. King, "Speech signal analysis, synthesis and recognition exercises using Matlab," *Int. J. Elect. Eng. Educ.*, vol. 34, pp. 161–172, 1991.
- [5] *DSP-CIT eco Hardware Handbook*, Vol. 1.0, dSpace Digital Signal Processing and Control Eng., GmbH, Paderborn, Germany, 1993.
- [6] L. Wang, *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [7] K. Erbaturo, O. Kaynak, and I. Rudas, "An inverse dynamics based robot control method using fuzzy identifiers," in *Int. Conf. Advanced Intell. Mechatron.*, Tokyo, Japan, June 16–20, 1997.

**Ahmet Bugra Koku** (S'97) was born in Erzurum, Turkey, in 1971. He received the B.S. degree in mechanical engineering in 1994 and the M.S. degree in systems and control engineering in 1997, both from Bogazici University, Istanbul, Turkey. Currently, he is pursuing the Ph.D. degree at Vanderbilt University, Nashville, TN, and is working in the Intelligent Robotics Lab (IRL).

His current research interests are memory organization of robots, behavior modeling and learning for robots, and low-cost hand design for humanoid robots.

**Okyay Kaynak** (S'80–SM'90) received the B.Sc. degree with first class honors and the Ph.D. degree, both in electronic and electrical engineering from University of Birmingham, U.K., in 1969 and 1972, respectively.

After spending seven years in industry, in January 1979, he joined the Department of Electrical and Electronics Engineering of Bogazici University, Istanbul, Turkey, where he is presently a full Professor. He has chaired the Computer Engineering Department for three years and of the Electrical and Electronic Engineering Department for two years and served as the Director of Biomedical Engineering Institute for one year at this university. Currently, he is the holder of the UNESCO Chair on Mechatronics and the Director of Mechatronics Research and Application Centre. He has held long-term visiting professor/scholar positions at various institutions in Japan, Germany, the United States, and Singapore. His current research interests are in the field of Intelligent Control and Mechatronics. He has published three books and edited three. He has authored or coauthored almost 200 papers in various journals and conference proceedings.

Dr. Kaynak is presently President Elect of the IEEE Industrial Electronics Society and an Associate Editor of IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.