EXPERIMENTAL PLATFORM FOR SIMULATION AND ANALYSIS OF SYSTEMS STATES AND FAULT DIAGNOSIS

Konstantin DIMITROV, Dimitar NURKOV
Technical University - Sofia, Bulgaria

Abstract. The present paper describes the development of an advanced experimental platform, designated for process modeling, simulation and analysis of technical conditions and fault diagnosis in a real and/or simulated industrial processes and systems states. The developed experimental platform was applied in the research and development studies as well as in the teaching programs of research students and professors in Technical University – Sofia, Bulgaria.

Keywords: experimental platform, simulation and analysis of systems states, fault diagnosis, reliability enhancement

1. Introduction

The successful completion of the so much sophisticated tasks, like *simulation and analysis of systems states, during the Fault Diagnosis (FD) procedures,* requires also a *development of specific **Experimental Platforms*** [2, 6, 7].

In general, such type of experimental platforms must be equipped with an incorporated and/or exterior, ***modular algorithmic and programming systems,*** designated for *process modeling, condition simulation and evaluation, fault diagnosis and reliability evaluation* [1, 3, 4].

The general purpose for the creation of specific experimental platforms must always be ***object-oriented***, i.e., these platforms should be capable ***to create*** various kinds of *initial data bases,* containing the necessary and the specific diagnostic information, (applied during the pattern recognition procedures), ***to simulate*** different technical conditions and systems states, ***to develop*** various representative process models (even under stochastic and/or fuzzy operating conditions), ***to detect and to isolate*** faults and failures and ***to overcome*** some possible measuring errors [3, 5, 6, 7].

The present paper describes the development of an advanced experimental platform, designated for process modeling, simulation and analysis of technical conditions and systems states during the development of fault diagnosis procedures in real and/or simulated industrial processes and systems conditions.

The created Experimental Platform was applied in the research and development activities, as well as in the teaching programs of PhDs students, assistants and professors from Technical University – Sofia, (and more specifically at the Department of Mechanical Engineering, the French Faculty of Electrical Engineering, Automation, Electronics and Informatics the Laboratory of Reliability and Diagnosis etc.).

2. Design and development of the experimental platforms mainframe – general modules, operational devices and control systems

The general structure of the experimental platform is shown at Figure 1.

The main modules of the developed platform respectively are:

- Platforms Main PC;
- Central Processing Unit (CPU Board);
- Electrical Motors (from I thru V);
- Incremental Encoders (sensors);
- Inductive Sensors;
- Sensor Electronic Device (SED);
- Control System and Electronics;
- PSU for the Motors and for the Control System;
- Platforms steel base and cable lines.

The ***CPU-programmable controller,*** i.e., "**The Board**" (see Figure 2), represents in fact the ***core*** of the platforms control system. The board is designed *to perform all activities and specific calculations,* needed for analysis and treatment of the specific data bases (supplied by the sensors), *to generate the necessary control actions,* (developed under the specially designed control algorithms), and *to apply the analytic and decision-making procedures,* generated in the platforms main computer.

The CPU-controller (the Board) is incorporated in the platforms structure, but can also be used independently, i.e., when performing some research on an industrial system, (under real operation conditions).

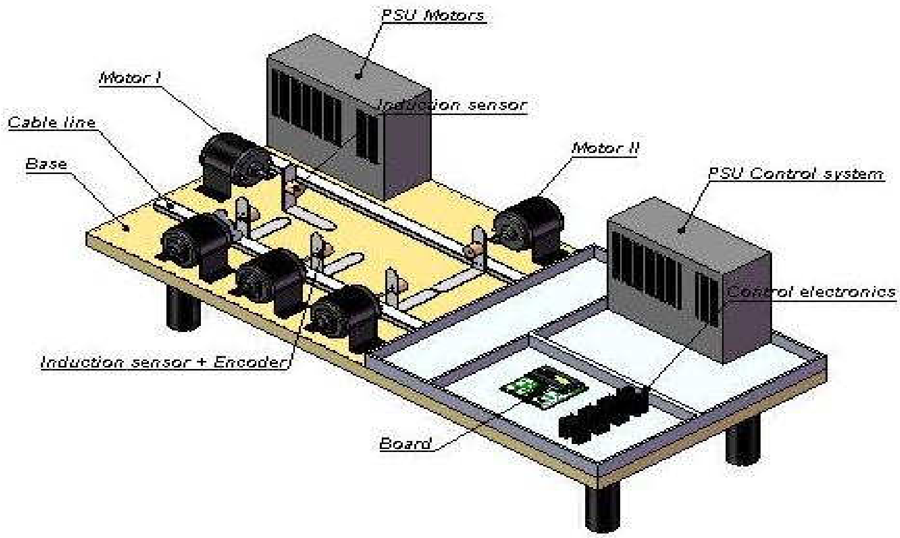

Fig. 1. Structure of the developed experimental platform

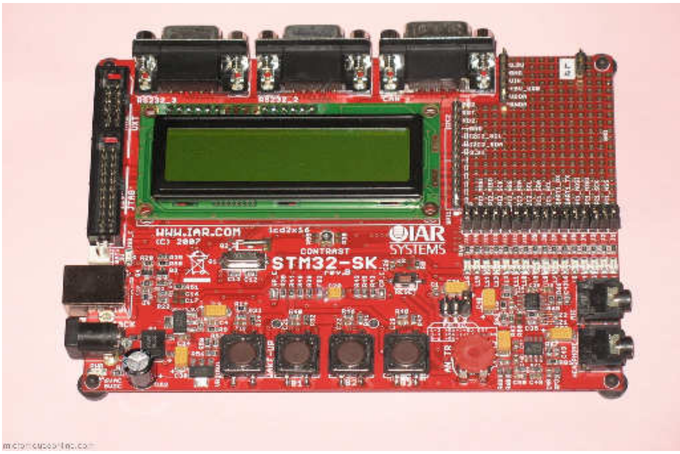

Fig. 2. CPU-programmable controller: STM32 – SK

Fig. 3. Sensor Electronic Device (SED)

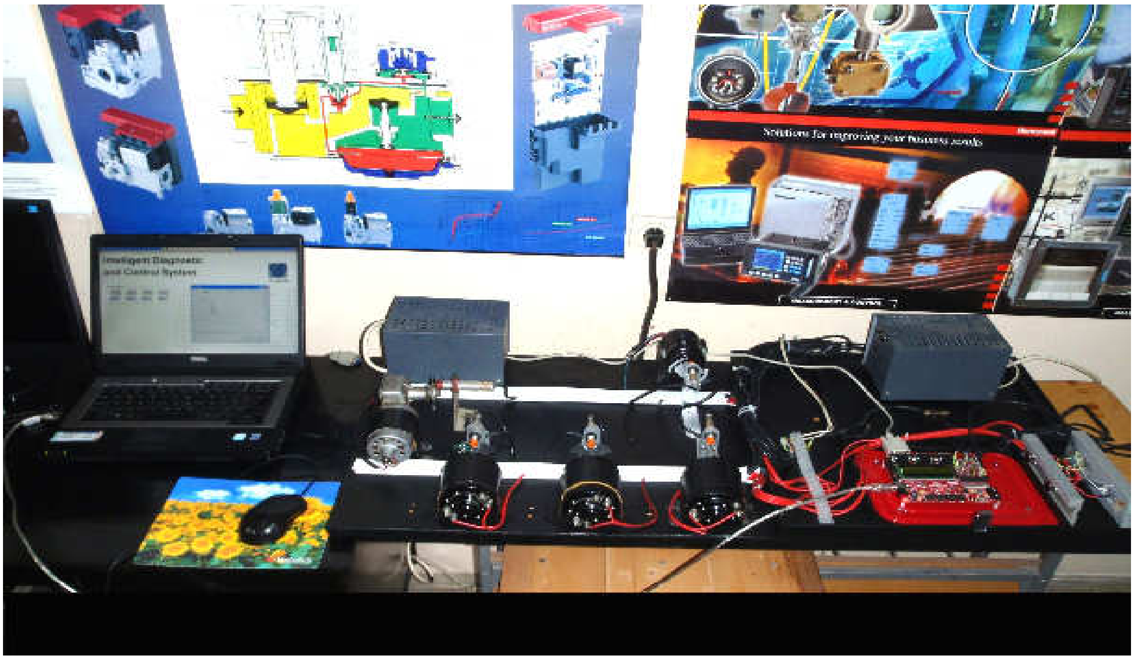

Fig. 4. The developed experimental platform – entirely completed and ready for experiments

The CPU-controller is of the STM32-SK model, and is manufactured by *ST Microelectronics Corp*. It is of an APM 9-generation, and possesses an ***uCOS Micrium Operating System***, which can provide a real time processing of the information, with a maximal frequency of 72 MHZ, thru 16 channels, 7 timers, and 80 high-speed entry/exit devices.

The incremental encoders are of the model KUBLER, series 2400, with maximal impulse frequency of 160 kHz, 12 000 rotations per minute (maximal), and a signal level of generated logical zero 0.5 V.

The induction sensors are manufactured in TU – Sofia, and are designated to generate digital signals, during the creation of stochastic and fuzzy process realizations (applied respectively in the learning procedures of the Pattern Recognition algorithms).

The developed Sensor Electronic Device – SED (Figure 3), was specially created for detection and recognition of different types of movements (rotation in this case). The SED is based on a microcontroller with 8051-core and is equipped with a wireless communication module. The sensors part of the SED consists of a tri-dimensional accelerometer, a tri-dimensional magnetic field sensor, a two-dimensional gyroscope and a microphone. All data, that are collected by the sensors and treated by the core of the SED are transmitted to the sensors main PC.

The real status of the developed Experimental Platform (entirely completed and ready for experiments) is shown on Figure 4.

3. Control of the Experimental Platform – development of Board-Control and Motors-Control Algorithms and Software Systems

A multi-layer General Control Algorithm (GCA) is specially created for the Experimental Platforms control and its structure is presented at Figure 5.

The superior (i.e., the higher) level of the GCA represents the “Main Application” block, which is designed as a ***software Intelligent Diagnostic and Control System (DIACON)***.

The second level (from the top) of the GCA is developed as a “Main Task” algorithmic module, which is designated to perform coordination, supervision, and execution of the command actions (i.e., commands), generated by the platforms “Board” (i.e., the programmable micro-controller).

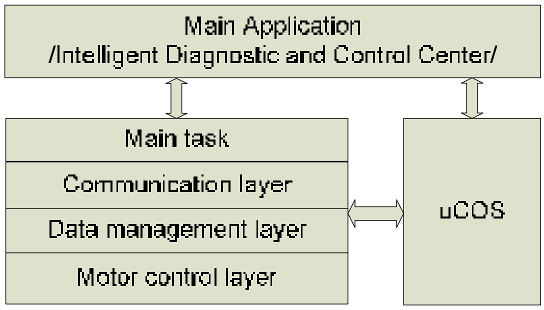

Fig. 5. Multi-layer, Hierarchical Structure of the General Control Algorithm (GCA)

The third level (always from the top) of the GCA is the “Communication Layer”. It is designed to perform the reception, the transfer and the follow up of the generated commands (and the treated data as well) between the platforms “Board” and the main PC.

The purpose for the creation of a “Data Management Layer” (i.e., the GCA fourth level) lies in its capability to realize and to follow up the data-transfer between all developed GCA modules.

The lower level of the GCA is the “Motor Control Layer”, which respectively is developed as specialized Motor Control Algorithm, designed to control the platforms motors.

A rather sophisticated Board-control Algorithm (“BOACON”), designed to execute the interactions and synchronization between the main layers of the GCA and uCOS-II kernel was developed and applied.

The complex structure of BOACON-algorithm is shown at Figure 6.

The initial three levels of the Board-control algorithm are designed to perform the initialization of the control peripherals i.e.:

- UART initialization (with Baud Rate of 115200 Bauds/sec);
- Initialization of the Analogue to Digital Conversion (ADC) Unit – the module uses 12 bits ADC and 6 channels;
- Initialization of the Basic Peripherals Settings (BPS) – PWM, Timers, Input/Output (I/O) – selection, etc.

The kernel configuration and the Task Creation are developed on the fourth level of the BOACON-algorithm.

Once the Task Creation activities are fulfilled, then – the Multitasking and the Main Task blocks (designed respectively as a fifth and a sixth level) begin to perform.

The Main Task block is developed as an ***infinite loop***, of a While/True type, designed to fulfil the task management, task synchronization and task delay options.

The Data Receive Task block handles the accepted (the transferred) data, in cases, when the Call-back function is being activated. This block also verifies the Protocols, and does the separation between the information data-bases (the sources and/or the generated experimental sets) and the really (i.e., the operational) generated command actions (i.e., the commands).

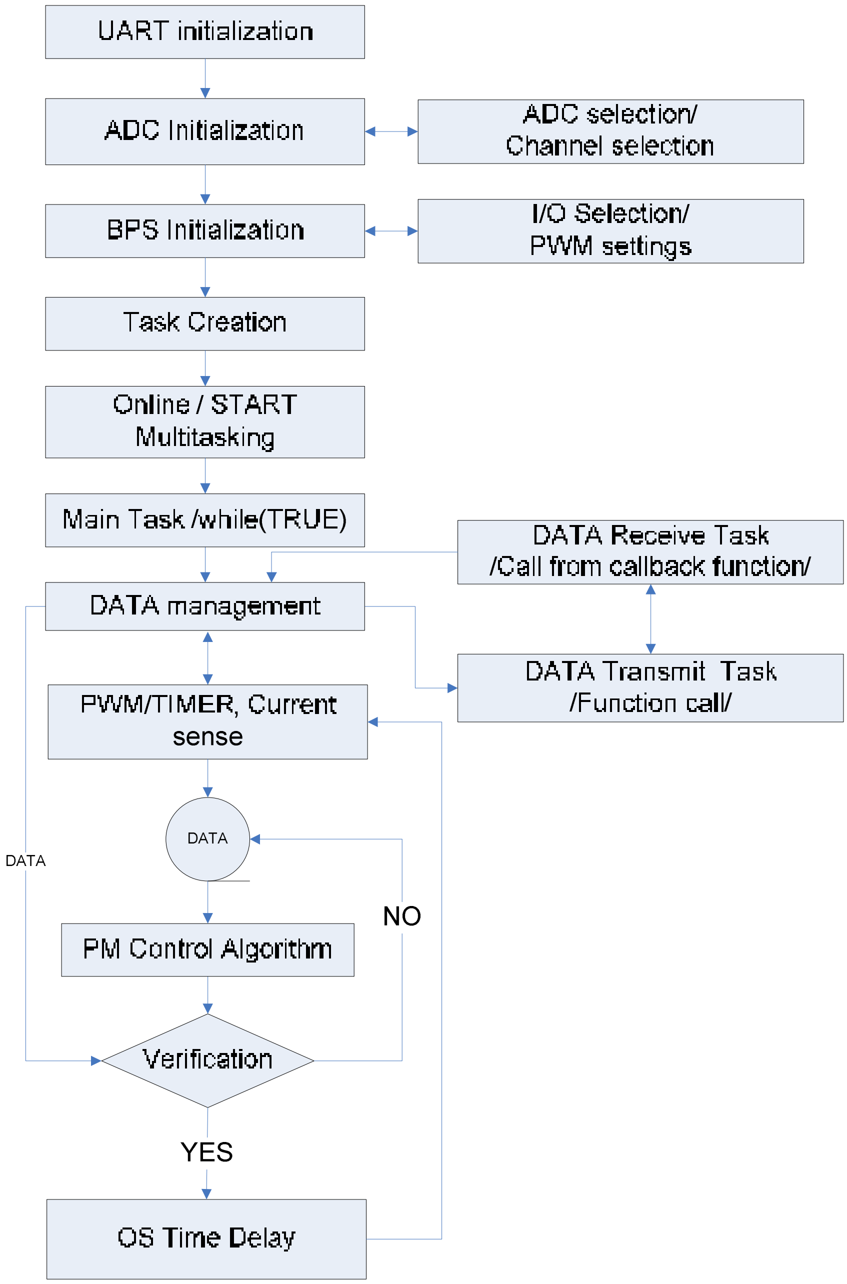

Fig. 6. General Structure of the Board-Control Algorithm “BOACON”

The Data Transmit Block does the actual transfer of the already treated data and commands (via the Function Call module), towards the Data Management Block. There, the currently generated answers (i.e., results) are processed and transferred towards the specific PM Control Algorithm (i.e., the algorithm that controls the operation of the Platform Motors (PM).

Actually, the Data Management Block is one of the most important modules of the BOACON-algorithm, since it does the coordination, the treatment and the generation of all commands and representative data-bases.

All actions, related to the processing of all specific motor-control parameters (i.e., current, rotation, time-cycles, etc.), which are generated by the PM Control Algorithm, as well as the determination of the new settings for the motor control (via the PWM/Timer module), are also effectuated in this block.

Once the new settings of the motor control parameters are generated – then the PM Control Algorithm starts the new motor control regimes.

During the realization of all these algorithmic cycles, the Kernel of the uCOS remains in its current configuration for a time period, defined by the OS time delay. The uCOS system can thus process some other types of tasks (related mainly to the modeling and simulation activities), or can remain in an idle status.

The general structure of the developed “Motors-Control Algorithm (MOTCON) is shown at Figure 7. The MOTCON algorithm is creates the basic levels of the GCA and BOACON algorithmic structures, where its general purpose spreads over the checking and verification of the motors control parameters, (current, number of tours, types of the cycles, etc.). The command actions (i.e., the commands are created via the PWM-control and are based on PID Speed regulation (Figure 7). The generated loop operates until the set time period expires, or until the new settings (generated via the Data Management block of the BOACON-algorithm) enter into the MOTCON structure. Once the setting are submitted – then the new phase (i.e., a new loop containing an alternative parametric values) of the Control algorithm become operational.

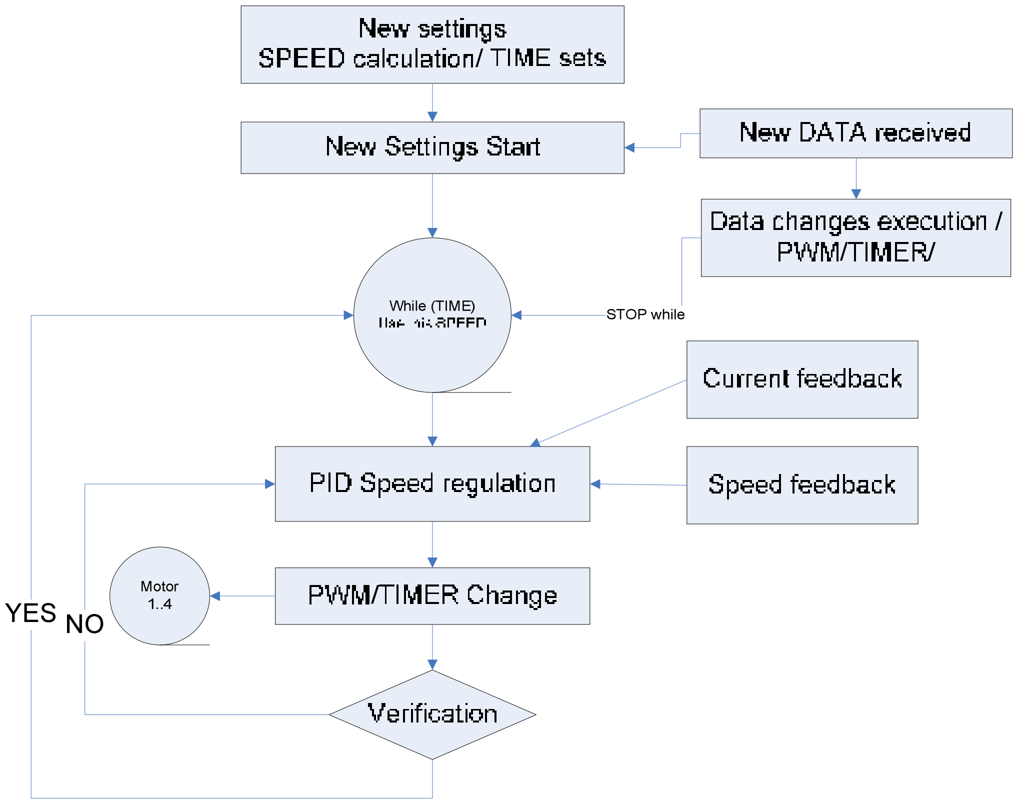

Fig. 7. General Structure of the Motors – Control Algorithm “MOTCON”

Finally, an enhanced (specific) ***Software System, named “Intelligent Diagnostic and Control System (DIACON),*** designated for evaluation and control over the generated process realizations, as well as for visualization of the algorithmic modules was also developed and applied in the experimental platforms mainframe. The DIACON system was created as a Windows-based Software System, designed under *Visual C* and *Visual Basic* environment.

Some sample realizations of the DIACON performances are shown in Figure 8.

The DIACON software system was also realized as a specific hierarchical structure (possessing several processing and decision-making levels), which provides an enhanced options for submitting, treatment and evaluation of the data-bases, as well as capacities for learning (supervised and/or un-supervised), visualization, storage, and generation of the required decisions (current commands and/or final results).

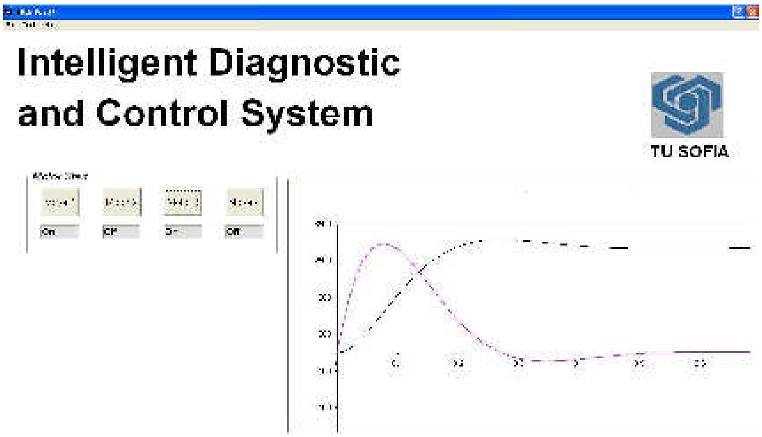

Fig. 8. A sample realization of the Intelligent Windows-based Software System, (INTEDIACON), specially developed to control the Experimental Platforms functional activities

4. Conclusions

4.1. The entire mainframe of an advanced Experimental Platform designated for process modeling, simulation and analysis of systems states and technical conditions during the development of Fault Diagnosis in real and/or simulated industrial processes was designed and realized. The newly developed Experimental Platform is entirely completed and ready to fulfill laboratory experiments in the area of fault diagnosis, operational reliability, simulation and modeling, of systems technical states, pattern recognition etc.

4.2. Some particular and general type algorithms, (of the GCA, BOACON and MOTCON type), designed for evaluation and treatment of all sensors data bases, as well as for flexible control of the platforms Central Processing Unit (CPU) STM32-SKP, and the platforms motors were developed and applied.

4.3. An enhanced specific Software System, DIACON, designed for evaluation and control over the generated process realizations, as well as for, learning and visualization of the algorithmic modules was also developed and applied in the experimental platforms operation.

References

1. Dimitrov, K.D., Danchev, D.: *Reliability of Machines and Systems*. Technica, Sofia, 1999 (in Bulgarian)
2. Dimitrov, K.D. et all: *Multivariable, Model-based Expert System for Condition Assessment and Fault Diagnosis of Industrial Systems*. Injenerno Proecktirane (Engineering Design), Vol. 1, p. 63-69, 2008, Sofia, ISSN 1313-7530 (in Bulgarian). Available at: http://ptst.mf.tu-sofia.bg/Journals/ BSIP/ articles/ article019.swf
3. Dimitrov, K.D. et all: *An automated, modular system for pattern recognition and evaluation of process characteristics in industrial systems*. HCTech – 2008, Injenerno Proecktirane (Engineering Design), Vol. 1, p. 71-80, Sofia, 2008, ISSN 1313-7530 (in Bulgarian). Available at: http://ptst.mf.tu-sofia.bg/Journals/BSIP/ articles/article012.swf
4. Frank, P.M.: *Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results*. Automatica, Vol. 26, no. 3, p. 459-474, 1990, ISSN 0005-1098
5. Isermann, R.: *Process Fault Detection based on modeling and estimation methods – A Survey*. Automatica, Vol. 20, no. 4, p. 387-404, 1984, ISSN 0005-1098
6. Jain, L., de Silva, C.: *Intelligent Adaptive Control*. CRC Press, 1999, ISBN 9780849398056
7. Tzafestas S.G.: *System fault diagnosis using knowledge-based methodology*. Prentice Hill, 1999

Received in February 2010