COMPLEX ALGORITHMIC AND SOFTWARE SYSTEM FOR SIMULATION OF TECHNICAL STATES ON AN EXPERIMENTAL LABORATORY PLATFORM

Konstantin D. DIMITROV, Dimitar NURKOV
Technical University - Sofia, Bulgaria

Abstract. The present paper describes the development and the application of a complex and sophisticated algorithmic and software system, which is designed to simulate technical conditions and systems states, which could be generated in a real industrial equipment, (and under real operational conditions), but on a specific experimental laboratory platform, (specially created for the purpose).

Keywords: algorithmic system, experimental simulation, technical conditions, systems states

1. Introduction

During the recent years, the great deal of efforts of many engineers and sciences were focused on a creation and a development *of automated and/or intelligent systems for fault diagnosis* [3, 4].

In general, such systems find their application mainly in the area of evaluation and real-time control of the operational reliability in industrial complexes [2, 3, 4].

It is expected, that, such types of advanced diagnostic and control systems should be capable of providing *safety, superior reliability* and *fault tolerance* during the operation stages of industrial and logistics system(s) [3, 4].

The needs to provide so enhanced capabilities in the contemporary industrial equipment, gave rise to the development of some really sophisticated algorithmic and software systems for ***monitoring and supervision of the systems technical state(s), under real operational conditions*** [2, 4, 5].

In order to be able to develop their full power, the diagnostic algorithmic systems should therefore be submitted to various *preliminary experimental and modeling tests*.

In general, the majority of these experimental tests should be developed on specific *laboratory platforms,* that are specially created and applied for the purpose [1, 3].

The present paper describes the development and the application of a complex and sophisticated algorithmic and software system, which is designed to simulate technical conditions and systems states, which could eventually be generated in a real industrial equipment (and under real operational conditions), but on a specific experimental laboratory platform (specially created for the purpose) [1, 2]. The system is then applied for simulation, modeling and learning procedures of real industrial processes and systems states.

2. Operational (functional) strategies and general exigencies of the developed algorithmic and software system

During the developed laboratory experiments, the diagnostic algorithmic and programming systems should be applied for functional and diagnostic tests over various simulated industrial processes and technical states, i.e., under a simulated operational environment. Actually, all this means, that, the developed experimental laboratory platform(s), should be capable to create, (i.e., *to simulate*) such types of operational environment, i.e., simulation of various *technical (systems) states*, generation of a *failure and/or fault in the process variables*, simulation(s) of various *process realizations*, etc., etc. A great deal of supervised and un-supervised learning procedures must also be developed and applied over the created algorithmic modules.

A specific, *experimental laboratory Platform for Process Modeling, Simulation and Analysis of Technical Conditions was specially created and applied during the developed algorithmic procedures.* The detailed structure, the command algorithms and the operational characteristics of the created experimental platform are developed in details in [1]. The general scheme, presenting all major modules of the experimental platform is shown at Figure 1.

The experimental platform and respectively the created *algorithmic and software system* is designed to perform three main operational strategies, of the following type:

- ***Operational strategy A***. Generation of representative data bases, (simulating the process variables), based on the stochastic rotation of the experimental platforms motors (see Figure 1), i.e., creation of data sets with entirely random content(s).

  This strategy is suitable for some initial analysis, but also for testing the modeling options over the created sets. The strategy is also very useful for testing the *fault tolerance of the systems*, i.e., the reaction of the command/control system to some *sudden* and eventual faults and failures (that are randomly generated during the creation of the process realizations)
- ***Operational strategy B***. Generation of representative data sets, via modeling function(s). The strategy is based on the application of specific command blocks of the platforms control algorithm BOACON, which is developed in [1]. During the operation stage of the BOACON-algorithm, some specific commands (which are based on modeling functional characteristics), are submitted (in real-time) to the motors control system and thus - the modeling data sets are created during the operation periods of the five platforms motors. The strategy is very flexible and suitable for current analysis and evaluation of process realizations and systems states, in the cases, when there is some lack of preliminary knowledge, i.e., for *unsupervised learning*.
- ***Operational strategy C.*** Generation of representative data sets via preliminary determined functional models. The strategy is based on a selection and application of *already developed models* of systems states and/or process realizations, which are stored in a specific modeling library. The strategy is appropriate for evaluation of *supervised learning*, i.e., in the cases, when the process realizations and systems states cold are modeled adequately, based upon some *existing preliminary knowledge.*

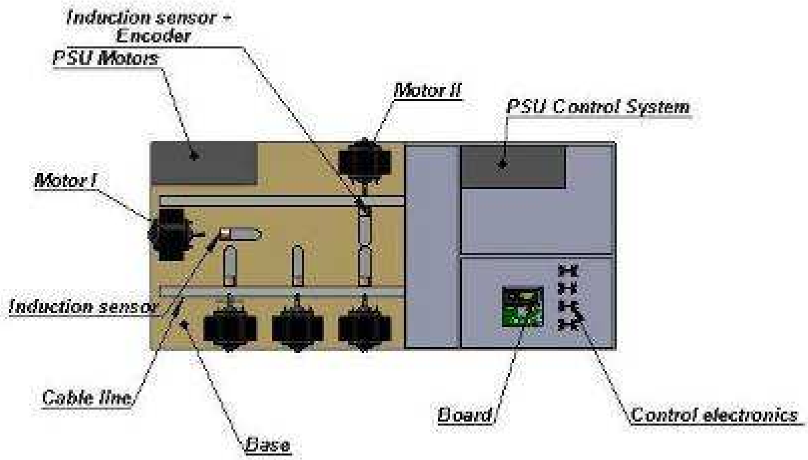

Figure 1. General Scheme of the developed experimental platform

3. General structure and operation of the complex algorithmic structure TECHNOSIM and the software system DIACON - simulation of technical states and process realizations

The developed hierarchical structure of the complex algorithm, designed for simulation and modeling of systems states and process realizations on an experimental laboratory platform (named TECHNOSIM) is presented at Figure 2.

The so-created TECHNOSIM-algorithm is further realized as an enhanced *Software System, named **"Intelligent Diagnostic and Control System" (i.e., DIACON).***

The software system is designated for evaluation and control over the generated process realizations, systems states, generated (simulated) faults and failures etc.

The DIACON-system is realized as a Windows-based Software System and designed in *Visual C* and *Visual Basic* environment.

The initialization process of the DIACON-system includes the Com Port selection, the transfer speed synchronization, the determination of the bits (number, stop-functions, etc.). During this first stage the system opens an initialization window, which presents the following parameters:

- Com Port: 1

- Baud Rate: 115 200
- Parity space: None
- Data bits: 8
- Stop nit: 1.

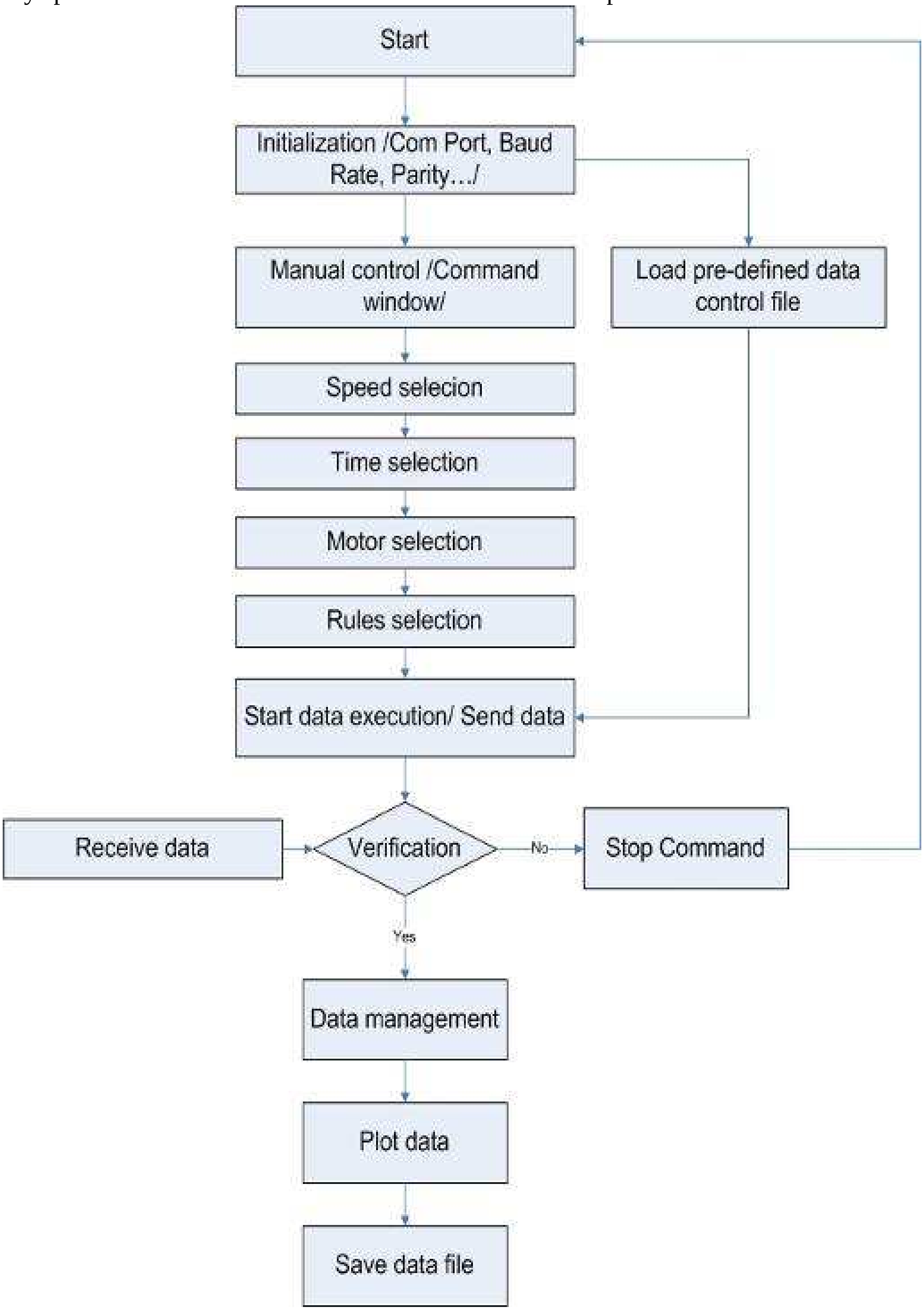

Figure 2. Hierarchical structure of the complex algorithm for simulation and modeling of technical states on an experimental laboratory platform - TECHNOSIM

When the initialization stage is completed, the DIACON-systems enters into a STANDBY-regime. Then, one of the three operational strategies – respectively "A", "B" and "C" can be selected.

During the development of *Strategy "A"*, all necessary control parameters, as well as their values (i.e., number and type of the platforms motors, their speed of rotation, their time periods, etc.), are selected form the specific menus, developed in separate windows of the DIACON-system.

During the development of *Strategy "B"*, the required values of the modelling functions can be either submitted by the BOACON modules, either selected from a library of experimental values (preliminary determined).

During the development of *Strategy "C"*, all values of the representative data sets for the simulated process realizations and systems states (i.e., the preliminary developed models), are extracted from the DIACON-library, as a specific file(s), which is submitted for realization.

Once the initial stage and the selection of the appropriated strategy is effectuated, the DIACON systems begins the realization of the connection processes between the modules of the BOACON-algorithm (please see [1]), and the TECHNOSIM-algorithm (please see Figure 2).

During the connection stages, the program modules of DIACON effectuate the data transfer of the selected/generated/extracted data towards the board (i.e., the programmable microcontroller **STM32 – SK –** please see [1]). At the same time, all data, generated by the platforms motors and treated by the microcontroller are submitted to the TECHNOSIM-modules – please see Figure 2. The communication between the algorithmic modules is effectuated via specialized serial ports and a specially created protocol in DIACON-system. The submitted data are treated in the "Verification", "Data management" and "Plot data" modules of the TECHNOSIM-algorithm (please see Figure 2.).

The visualisations of the currently obtained and/or final results of the DIACON-system operation are realized via two types of buffers – the first buffer type is designed for graphical and numerical visualization, and the second one – for saving the files, realized during the experiments.

The operation stages of all strategies can be interrupted, paused and even combined, i.e., a transfer and an exchange of experimental and modelling data can be effectuated during the simulation and modelling experiments.

Some examples of the realizations, generated by the DIACON-system during the development of *Operational Strategies* "A" and "B" are respectively shown at Figure 3 and Figure 4.

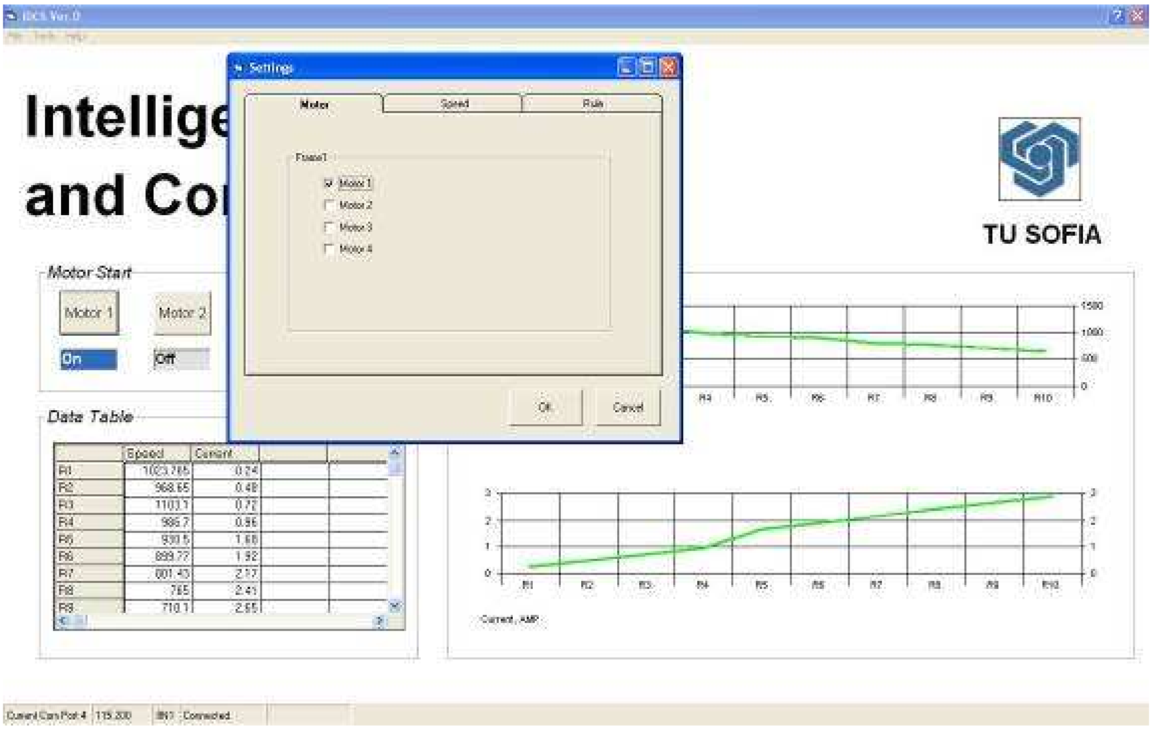

Figure 3. Some examples of algorithmic decisions, generated by the DIACON-system, during the realization of an Operational Strategy from "A"-type

4. Conclusions

4.1. Three general types of operational (functional) strategies for the developed algorithmic and software system are developed and applied for modeling and simulation of process realizations.

4.2. The general structure of the complex algorithmic structure ***TECHNOSIM***, as well as the windows-based software system ***DIACON***, designed under *Visual C* and *Visual Basic* environment and designated for evaluation and control over the generated process realizations, as well as for visualization of the algorithmic decisions and results is developed in this paper.

4.3. The created algorithmic and software systems are applied for direct simulation and modeling of systems states and process realizations on an experimental laboratory platform (specially created for the purpose).

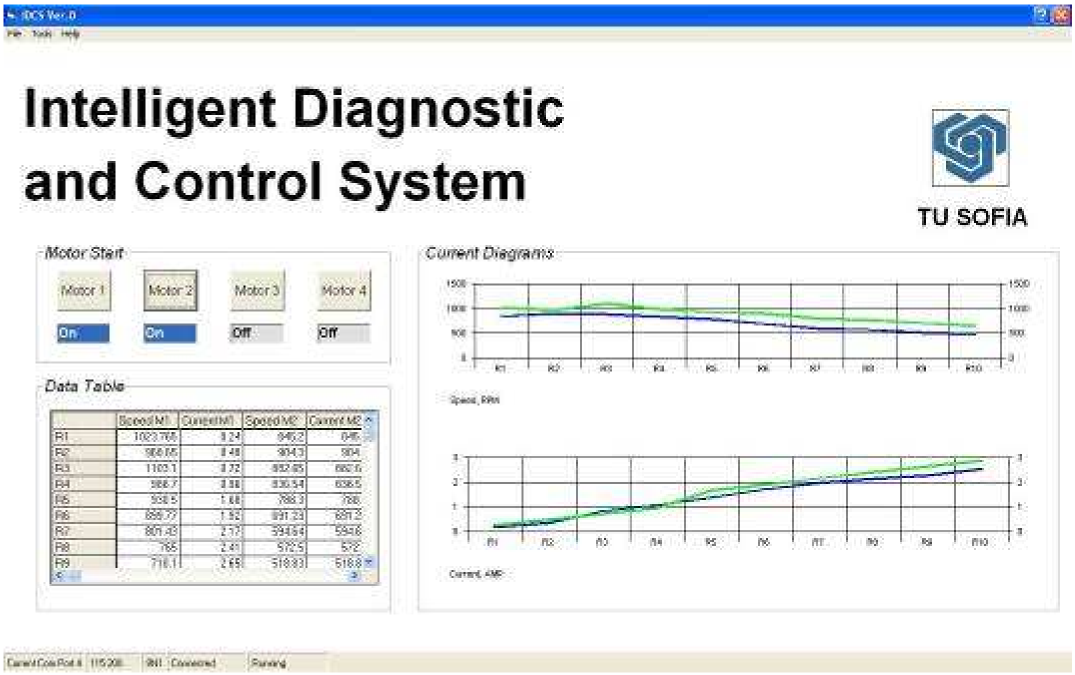

Figure 4. Some results, generated by the DIACON-system during the realization of an Operational Strategy “B”

References

1. Dimitrov, K.D., Nurkov, D.I.: *Experimental Platform for Process Modeling, Simulation and Analysis of Technical Conditions and Fault Diagnosis*. **RECENT**, Vol. 11 (2010), N^o 1(28), March 2010, p. 23-28, ISSN 1582-0246, Brasov, Romania
2. Dimitrov, K.D.: *Model-based fault diagnosis in a waste-processing industrial system via casual graphs*. **RECENT**, Vol. 10 (2010), N^o 2(26), July 2009, p. 101-105, ISSN 1582-0246, Brasov, Romania
3. Isermann, R.: *Process Fault Detection based on modeling and estimation methods – A Survey*. Automatica, 20, 1984
4. Kretsovalis, A., Mah, R.S.H.: *Effect of redundancy on estimation accuracy in process data reconciliation*. Chem. E. S., Vol. 42, p.2115-2121, 1997
5. Pinch, E.R.: *Optimal Control and the Calculus of Variations*. Oxford University Press, Oxford, 1993

Received in January 2010