Krzysztof JANISZOWSKI, Paweł WNUK
Warsaw University of Technology, Warsaw

PExSim – A NOVEL APPROACH TO THE PROBLEM OF THE INVESTIGATION OF COMPLEX DYNAMIC SYSTEMS IN AN INDUSTRIAL ENVIRONMENT

Keywords

Simulation, modeling, identification, fuzzy models.

Abstract

An approach to dynamic systems investigation based on a simulation kernel with a different auxiliary function is described. A proposed methodology is focused on processes, working in a more or less complex technology environment and integrates different approaches for the simulation of components of the investigated systems. It is realised in the form of an elastic programmable software package. A basic assumption of this proposition was a possibility of connecting this software with a real environment using standard electronic interfaces for the software emulation of parts of the technologic system. The package structure is very flexible thanks to plugin technology. Plugins are objects can represent the following: parts of control at the direct and supervising level, actuators and controllers, data processing elements (transducers, measurement units or sensors), dynamic systems described by differential systems of known components or identified, statistic models of a dynamic part system or table represented relations. All these forms are integrated in one platform and will work in real time; hence, it will be possible to use them as complex models of technologic installation. Responses of this type of system can be used for the simulation of technologic system behaviour in different or even extreme states, the testing of alarm and safety software, the

verification of intended control modifications, system prediction of future states of the system, the training of control staff, etc.

Introduction

Software simulation of dynamic systems is commonly used technique for the investigation of system behaviour in different operation conditions. Almost allmodern SCADA systems have some tools for modelling dynamic blocks, but they are adopted to a platform used by this system and are not standard. The scope of modelling possibilities is different and can vary from simple blocks, individually constructed for each part, e.g. ABB Advant, to models in the form of an artificial neural net, e.g. DeltaV. These tools can be used when the models are defined and can be included to the control structure. The professional SCADA systems are expensive and generally do not provide, or only provide, a limited possibility to programming in widely used languages like C++ or Fortran 77. Of course, a client of specialised technology can usually buy a model of factory installed type, but this is an extra effort and with a higher cost.

Another widely used approach to simulation is based on special written tools packages, that can be used extend to modelling or simulation, when the models of process are known. These packages are more or less specialised. For example, a package for the modelling large scale mechanic systems with sophisticated analysis of system kinematics and dynamics, based on finite element method principle – Adams, [1] is quite precise and used for the investigation of the resistance aspects of the mechanical structure construction or the optimisation of the designed system but can not be used for control or cooperation with an automation system for supervision control, etc. Other systems, designed for the simulation of hydraulic, pneumatic and electro-mechanical structures – SimulationX [2] has many attractive features, e.g. a built in large library of components for the control of mechanic systems with elaborated integration techniques that can solve dynamic transients with demanded accuracy. The user can built a system from blocks, e.g. an electro-pneumatic flow valve, a long pipe, a cylinder filled with air of a variable air temperature, a nonlinear friction with hysteresis, a mechanical load, etc., ignoring their nature (ordinary differential equations linear or non-linear type or partial differential equations) and run experiment. The results of simulation reveal different phenomena like parasitic oscillations, dead time in response and many others, but a necessity of solving a problem with predefined accuracy is in conflict with the possibility of real time operation and do not include a possibility of elastic introduction programs or procedures written in other standards. Both of the mentioned systems are developed with a high level professionalism and have, in fact, a closed structure - i.e. the user has to combine predefined components with his own parameters and perform the simulation. The introduction and application of one’s own procedure is of secondary importance and, in effect, is limited to special functions with very modest possibilities.

Other approach is presented in the general-purpose system - Matlab package [3], commonly used in academic institutions for the simulation of simple dynamic systems and the investigation of modern control procedures. This software is based on the assumption, that the system dynamic is well known or can be replaced by a form (e.g. transfer function or state equations) resulting from the processing of data with toolboxes like Process Identification. The Matlab package even has the possibility to control the system in real time with the RTW-toolbox. A simulation kernel of this package – Simulink [3] includes a set of different linear and nonlinear static and dynamic blocks in form of transient functions of a different nature. The user has to the define parameters of these blocks and put them together to perform the simulation. This approach, creating all possible effects from basic components, demands the definition of primary blocks with a very simple and original nature. This yields an uncomfortable result – a construction of one more complex phenomenon or functional component, e.g., nonlinear friction force with asymmetric hysteresis or a PID controller with commonly used actions, which induces a very complex block diagram, containing more then 30 - 40 basic blocks. A user's effort and time, needed for creating, testing and the verification of commonly used components is high, and, in effect, Simulink is used for simulation rather simple systems. More complex problems, like those mentioned in case of ADAMS or SimulationX, are not possible to solve in Simulink with the same quality.

The aim of the authors was to develop a package that can be used for the emulation of complex dynamic systems, composed of predefined dynamic blocks or corresponding multivariable models (estimated with own procedures), can cooperate in real time with industrial environment, and easy and flexible to extension by the user writing his own plugin objects. The package called PExSim (Process Explorer and Simulator) is written in C++ and can be run on a Windows or Unix (limited version) platform. The package has a form of menu controlled software with different options for the choice of operators on processed signals, simple dynamic models of components and additional elements used for simulation of time signals or events with possibility of visualisation. Possibilities for investigation of dynamic structures are shortly presented in the next section.

1. PExSim operators and components

1.1. Deterministic components

The first group of components is used for the generation of signals, processing and visualisation. The sources of signals can be used to determine time functions, Fig.1, in form of:

– constant value (with possible manual change at simulation),
– periodical signals like sinus, saw like, or pulse transients,

– time events with controlled parameters, like a step or ramp function,
– a generator of random pulses for each sampling interval.

The above signals have to be defined by the user setting the necessary parameters.

Signals can also be generated from text files. This way of generation creates the possibility of repetition and the comparison of responses of the dynamic modelled system with different version of control, supervision or alarm algorithms.

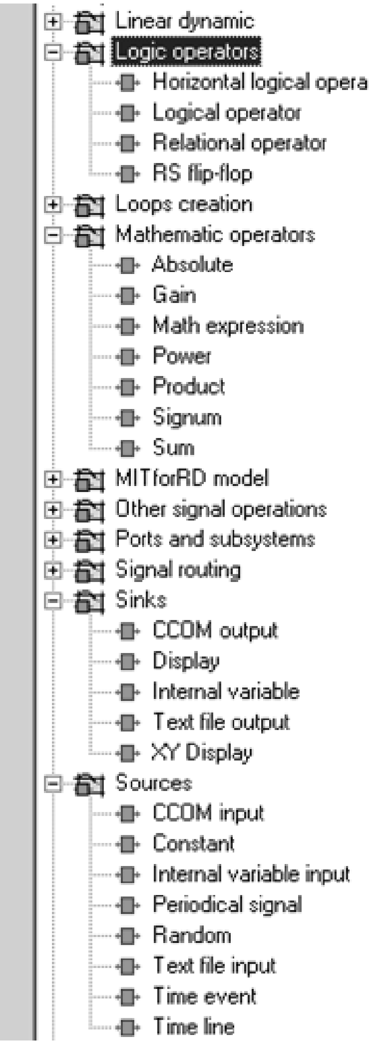

Fig. 1. List of arithmetic, logic operators and time events generated in PExSim

Another possibility of the introduction signals into structure of the simulated system is a direct input from an electronic interface or from an automation system (SCADA). In this case, a sampling interval is determined by a simulation option and can be not less then 10 ms.

The results of the simulation can be observed in the form of numbers exposed on top of each node of the simulated system, on multiple time displays with the elastic configuration of colours, the magnitude of responses and time marks on the tangent axis. PExSim has a possibility of arranging a XY display for specified pairs of signals, which is very suitable for the determination or verification a nonlinear correspondence between values. Other way to manage output signals is by recording them to a file or sending them to an electronic interface, e.g. cooperation with a supervisory SCADA system.

Signals within the package can be processed without delay by operators through the following:

- Arithmetical operations can be used with distinguished blocks in the form of commonly used sum, difference, product, amplification, absolute value, square, power or signum value, Fig.1. For more a complicated formula the user can adopt an arithmetic operator, that can be determined in commonly used a formula composed of function names, brackets, mathematic operators +,-, * and / and one, two or more arguments, corresponding to input signals or the results of functions determined in brackets.
- A supplement to the above operations can be used that provides different nonlinear functions of arguments (on values of signals) that are in the form of a table which defines the correspondence between values, called an "in system look up table".
- Logic operators can be used containing standard OR, AND, NOR, NAND, XOR (in multi-input form) and operators on logic signals corresponding to RS flip. The package will also provide the possibility to process signals according to fuzzy logic principles. Except for the mentioned operations, the package provides the possibility of the realisation of the functions of controlled switch that change the structure of the simulated system in dependence to the value of the signal. Another possibility is to change the system structure is to use min and max selectors, which provides a choice of the proper signal. Both mentioned functions are very useful and often used control structures.
- Another group of operations can be used that correspond to events that can not be defined before the simulation and depend on the observed or recorded transients of signals. It is possible to set a function of alarm corresponding to the approach of a value that represents a signal (to high or to low) or a state (fail of detector, sensor etc.) dangerous for the control system or user.

The second group of basic components can be shortly described time operators or dynamic blocks introducing a time delay equal to one sampling interval or more. To this group belong the following:

– linear dynamic blocks that represent: first and second order inertia, an oscillating process, an ideal differential term and differential term with inertia, an integrator and structure in form of a transfer function with one input and one output or a state space representation for multi-input, multi-output process description. This last form can be used for modelling of single input and single output system too (Fig. 2),
– time operations performed on single signals like smoothing, the calculation of a time derivative simple or the smoothed and integration of signal.

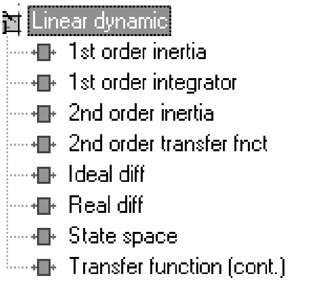

Fig. 2. Linear dynamic predefined blocks

Time dependent calculation of values corresponding to plugins demands sequent realisation (calculation) of all defined blocks. In the case of the direct propagation of signals (from sources to final outputs), the simulation is easy. In the case of closed loop structures, it is necessary to define a queue of calculations in a way possible to perform. Then it is sometimes necessary to shift in time the results of the operation to avoid the undefined states of calculation. For this purpose, a one step delay operator can be used that shifts in time (equal to single sampling interval) a result of the block output.

The next, large groups of predefined dynamic blocks are different models of dynamic components gathered in libraries. Such model can describe a component of installation, a physical or chemical process, an actuator or an algorithm. At the moment, there are the following libraries for these models:

Electric components for drive and positioning systems containing models for AC and DC motors, models of electronic controllers for these machines with the possibility of velocity, position and torque control,
– hydraulic and pneumatic components with models of proportional valve, nonlinear friction force, pneumatic cylinder and hydraulic cylinder,
– water-steam relations for the determination of a steady state of balance in a boiler with predefined steam parameters.

These libraries contain ready to use models of components that are intended to be use in the investigation of future applications: models of fast electro-static control in the modern drive of an extruder machine, the design and control of boiler drum in a local electrical turbine at power station in petrochemical applications, etc. The user has to define the corresponding parameters (e.g. motor power, the demanded or torque velocity profile, the displacement inertia of rotating pump, etc.) to start simulation.

The last type of library is a set of control algorithms. These are not models of processes but models of controller algorithms that can be used in simulation. In this case, the user has to define commonly used parameters, like gain, time constants for differential and integral action, the limitations of each action, initial states, etc. The parameters can be at first approximated with the application of the commonly used Nichols-Ziegler experiment and then used for tuning final controller settings.

1.2. Statistic models

The described possibilities of simulation are enlarged by including a class of dynamic models estimated by the statistic optimisation of multivariable, linear, neural or fuzzy models of processes. These models can be calculated by the MITforRD program [4] included in the PExSim package.

A structure of fuzzy model θ can be considered as a linear combination of a set of local dynamic models θ_i [5, 6]:

$$\theta = \sum_{l=1}^{L} \varphi_l(\mu)\theta_l \tag{1}$$

valid in areas defined by membership functions $\varphi_i(\mu)$. The above structure is called a TSK-model (Takagi-Sugeno-Kanga) and can be very efficient at the determination of dynamic quite nonlinear processes too [5]. The membership functions satisfy the following relations

$$\varphi_l(\mu) \geq 0, \qquad |\varphi_l(\mu)| \leq 1, \qquad l = 1,..,L \tag{2}$$

and usually are supposed to satisfy a normalisation condition::

$$\sum_{l=1}^{L} \varphi_l(\mu) = 1, \qquad \mu \in < A, B > \tag{3}$$

The argument of these functions is variable μ, called a fuzzy parameter, and defined over interval <A,B>. The shape of the membership function can be different, but it is always continuous and covers interval <A, B> with local domination of one function $\varphi_i(\mu)$. Typically used functions are trapezoids, Gaussian or Bayes distribution.

The form of each local model θi corresponds to a linear difference equation, equivalent to a transfer function G^i_{yu} (or set of transfer functions) between variable (or variables) defined as activation u and a variable representing, corresponding, result y

$$y = G^i{}_{yu} u \tag{4}$$

The MITforD program can automatically optimise both the number and shape of the membership functions φ_i and order and coefficients of transfer functions G^i_{yu} as well [7, 8]. This powerful tool for the determination of process dynamics introduces one limitation into simulation structure – the sampling interval at simulation must be equal to that used at recording data and used next at identification of TSK model (1).

An efficiency of this way of representation of the dynamics of the investigated process can be shown by the example of modeling a dynamics of one, future applications, real processes – a velocity of toggle form in a medium size extruder machine with a clumping force of 6000 N, Fig. 3. On Fig. 4 are drawn 3 different transients which represent the demanded velocity of the form, measured velocity of toggle and the simulated velocity of toggle based on measured input to the toggle drive system.

The quality of modelling is very good and suggests a simple reflection of the optimal control policy in the drive unit of the extruder; therefore, experiments on the drive system of the toggle can be replaced by experiments

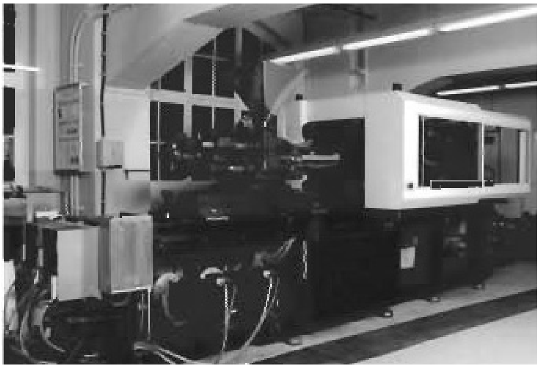

Fig. 3. Investigated electro-static controlled extruder HSIM-machine

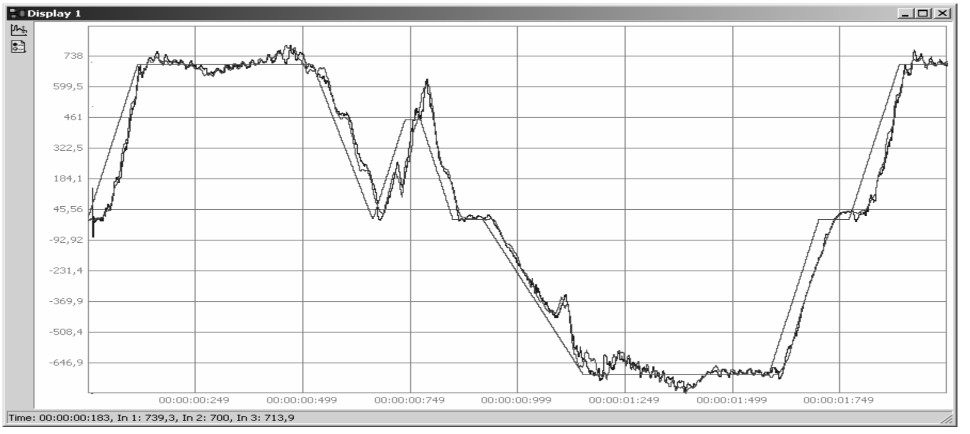

Fig. 4. Demanded toggle velocity measured toggle velocity and modelled toggle velocity

with the application of a model of this system using the fast prototyping approach, and the selected final algorithms can be tested and verified on final machine construction. The final construction can be not exactly known in moment of research and testing but is defined its structure and its dynamics will be very alike to those presented on Fig. 4.

The use of determined with MITforD model within PExSim and a way of simulation, shown on Fig. 4, is shortly presented on Fig.5, where MITforD model, as one of used in simulation structure components, is fed by signals measured within experiment and used for identification. These signals, recorded at experiments on real machine were gathered in Text file input 1, were next used for simulation of machine answer. Integral blocks, visible on Fig. 5, were used for determination of position of plunger cylinders driving a form in investigated extruder.

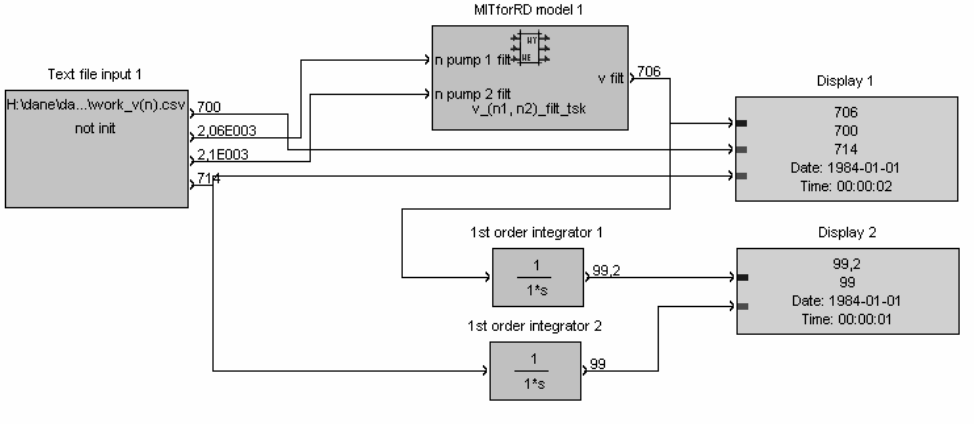

Fig. 5. The structure of the PExSim program for the testing and the verification of the MITforD model of dynamics investigated clumping unit in extruder

1.3. Time coordination and simulation options in PExSim

Simulation in this package is performed in discrete time with predefined sampling interval Δ. This value is fixed and determined before simulation. At simulation, this value is controlled and synchronised with a computer clock (in case of real-time simulation). The least possible sampling interval Δ, determined by the Windows platform, is equal 10 ms, but it can be dependent on the modelled system complexity. In case of a large value of sampling interval Δ (slow transients), the user can change the time scale and perform calculations faster. In very fast systems, the simulation can be slowed down and executed in latent time. The package has different options that control the process of simulation. Simulation can be stopped, interrupted and performed step by step with manual control for the observation of fast or non-continuous phenomena.

The above short presentation of components used in PExSim has to be completed by a discussion on time coordination. The use of the identified model of process dynamics determines a sampling interval for simulation. Then all components of the simulated structure have to be processed with the same sampling interval. During the simulation of linear dynamic blocks or logic operations, it does not induce problems. However, in the case of predefined models of components like valves, motors, pneumatic or hydraulic actuators, the fixed sampling interval can cause a converging solution when it is to large for the specified component. If the sampling interval is too large for fast reacting components, it yields the elimination of the components dynamics in calculated transients and in effect their dynamic behaviour is superfluous. These components have to be replaced by a corresponding static reaction in predefined activation. Then each model in mentioned libraries of ready components has to be defined for a sampling interval short enough to expose the component dynamics and the static relation for large sampling intervals. The problem of sampling interval fitting is less relevant in cases when the simulation loop will not contain MITforRD models of processes.

All of the above mentioned predefined blocks, operators and libraries are prepared by package authors. These components have met certain standards to be translated by special software. However, the user has the option of including fragments of his software through simple "Scripts." Scripts have a structure defined in the Script Pattern, written and translated using freeware GNU C++. This structure is quite sufficient to create even complicated dynamic nonlinear blocks.

2. PExSim structures

The PExSim package has a hierarchical structure that corresponds to the growing complexity of the modelled component. Each defined or parameterised block has at least one input or output node. The signal corresponding to this node (or nodes) can be observed, recorded or displayed.

The presentation of components demands a strict sequence of operation within defined blocks. The dynamic blocks (with exception of controlled time delay) are defined with a delay corresponding, at most, to one sampling interval. Some blocks of a pure static behaviour will not induce any delay. The primary blocks can have many input and output nodes in the case of multivariable dynamic systems of a high order, but its simulation is always performed with one step delay. Fig. 6 presents a closed loop system for velocity control with a DC motor, PID controller and DC amplifier, representing the limitation existing in the real drive together with the external torque and friction modelled as a linear function of the rotations and the display unit. The complete structure looks quite complex and leaves no space for the modelling of other components.

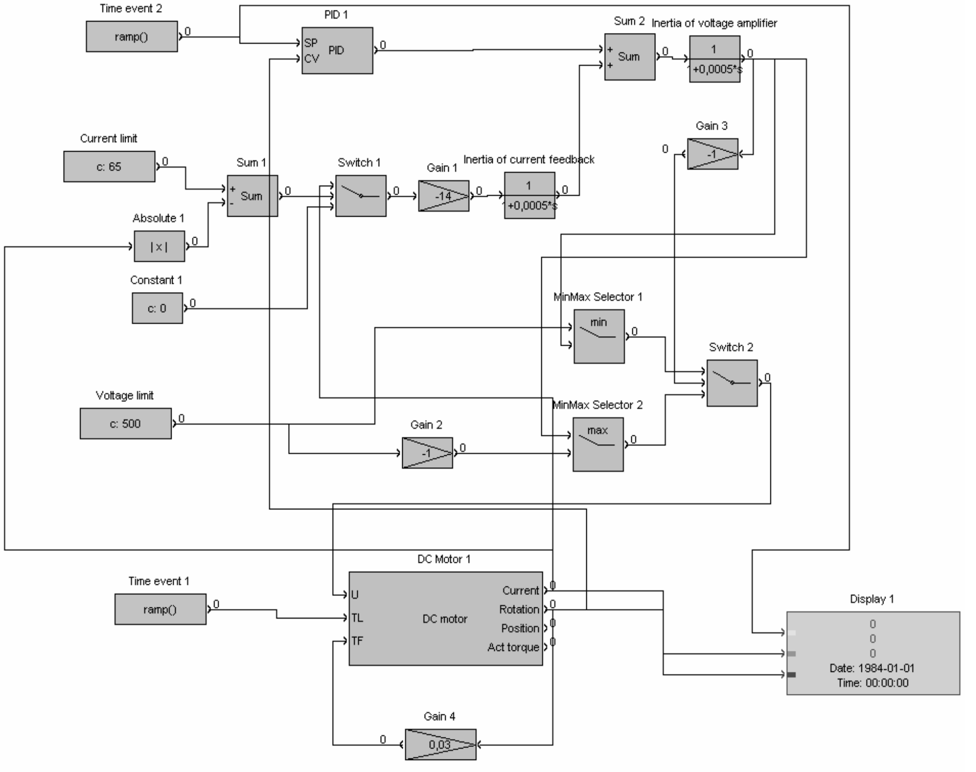

Fig. 6. Block system for the velocity control with DC motor, controller and current amplifier

A complete connection of some elements, here a set of blocks representing a DC amplifier functions, can be gathered in form of a “path” (Fig. 7).

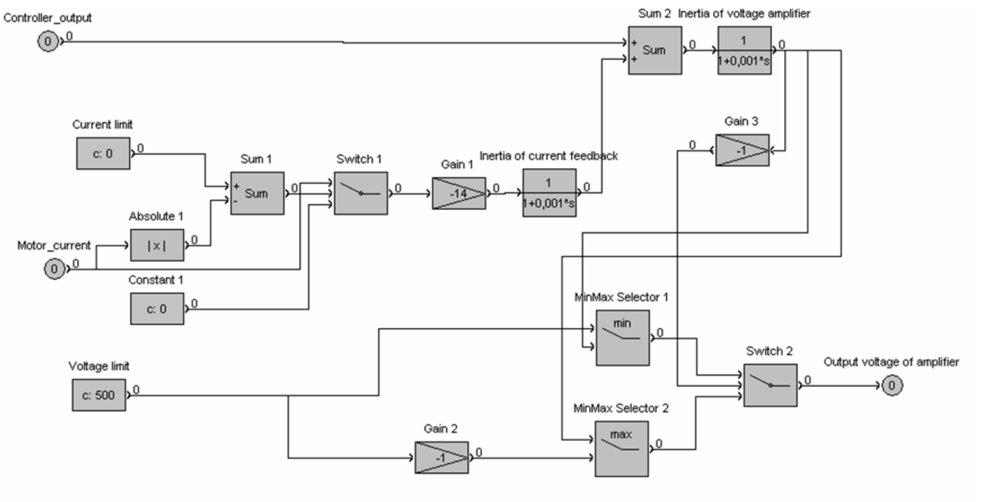

Fig. 7. Functional blocks of current amplifier path

The user has to define the input and output nodes of this type of structure (Fig. 6) and optionally give them corresponding names. Signals in these points are distinguished and processed as in the primary block. A simulation within one path is performed with a time delay equal to one sampling interval too.

The introduction of this path significantly reduces the total presentation of the drive unit and makes its much more transparent (Fig. 8).

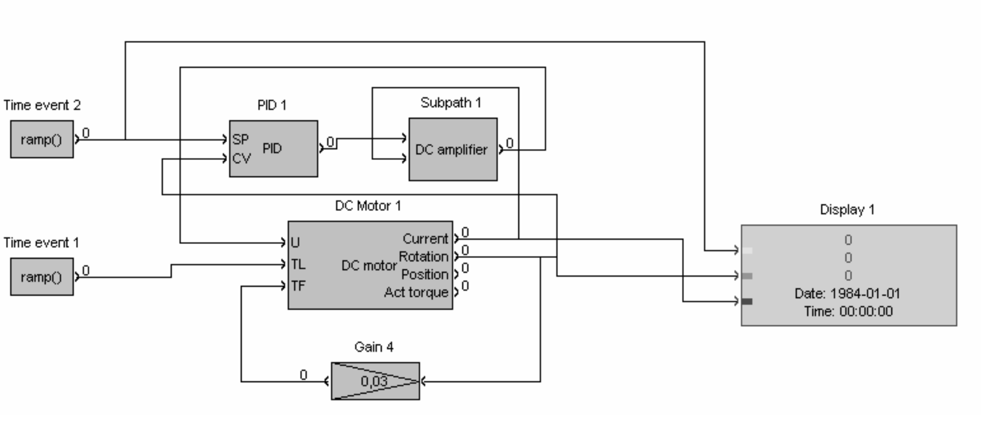

Fig. 8. Block diagram with a compact representation of the DC amplifier in form of path called DC amplifier

3. Application of PExSim

Europe is the world's leading region for the production of plastics processing machines. There are more than 15 big manufacturers of plastics processing machines, and there are many processors (around 15.000 in injection moulding, blow moulding and thermo-forming) that use these machines. The demand for plastics processing machinery can be readily traced to the demand for plastic parts, for which the application fields are as broad as construction, packaging, automotive, telecommunication, household goods, etc.

Plastic injection moulding machines (IMM) are the most important factory equipment for production of plastic components for industrial and consumer use, e.g. automotive, packaging, medical, audio or house hold supplies. A major industrial trend in plastic processing is towards components with higher complexities and reduced wall thickness. Many portable industrial components had, in the past, an average wall thickness of about 2.5 mm. Today, this is being reduced to about 1 mm. Furthermore, new plastic material replaces metal, e.g. for suction pipes or fuel supplies in automobiles. A mandatory requirement for these achievements is that the melted plastic material has to be injected into the mould at a higher speed than was common before (high-speed injection moulding (HSIM)). Injection moulding machines are highly productive machines with a very short cycle time and long weekly operation hours (e.g. 24h/day, 6 days/week). The leading edge of technology in plastic component production is HSIM, and it is required to produce parts with reduced wall thickness. The production process requires machines with very fast and precise control of hydraulic pressure and flow, due to the technology of the injection process and the need for minimum cycle time. The key challenge to achieve improvements in HSIM is the requirement to prepare flexible, intelligent and precise control mould and toggle control [9].

PExSim was applied to build a model of an injection-moulding machine, to be used at a later time during the development of the new control strategies. The original machine was build as a prototype in Technische Universität Dresden, where it was possible to measure positions, pressures and motor rotational speed, Fig. 3. At the moment, the model of the injection machine is limited to the clamping unit only. The clamping unit consists of many mechanical parts, Fig. 9, and its dynamic is quite complex. Two electric motors with fixed displacement pumps are used for suppling hydraulic pressure to main piston. Plunger cylinder pistons (two pairs, cross mounted) move plate with form through nonlinear toggle. At the stop position of the form (and minimal velocity), the toggle mechanism is activated to build the force necessary to hold the form fixed at the injection phase of the production cycle, Fig. 9. The position and velocity of the form is very important, because the form is the most expensive part of the whole machine. The proper form control has to provide very fast closing of the clamping unit but with zero velocity at the approach of the clumping position. An overview of one of the investigated control structures is presented on Fig. 10, where two motors are driving each pair of plunger cylinders.

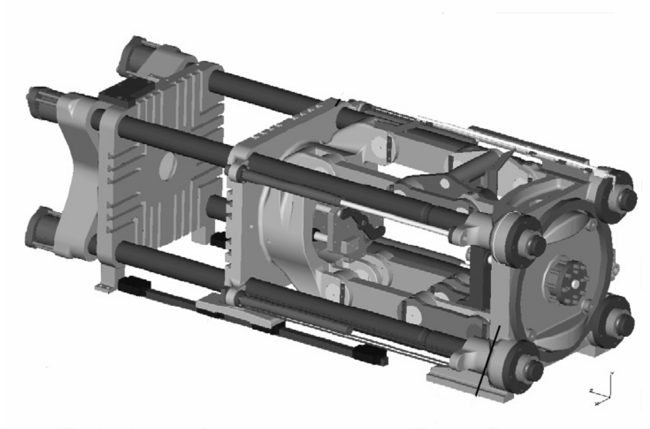

Fig. 9. Construction of mould and toggle mechanism in HSIM machine

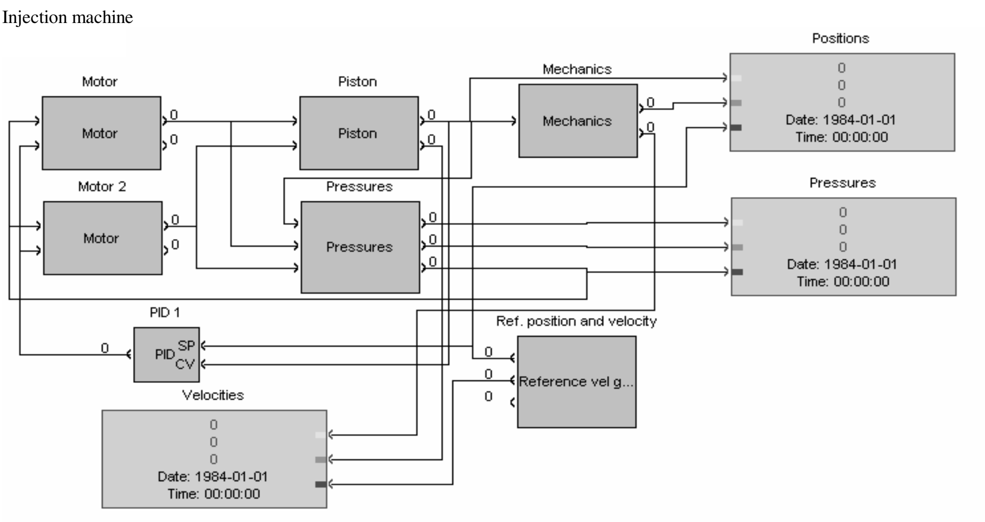

Fig. 10. A block diagram of closing mechanism of the HSIM machine

First part of the model is the reference velocity signal generation. Usually, the reference velocity signal has the form presented in Fig. 11. A PExSim model in the form of a path was used for its generation (Fig. 12).

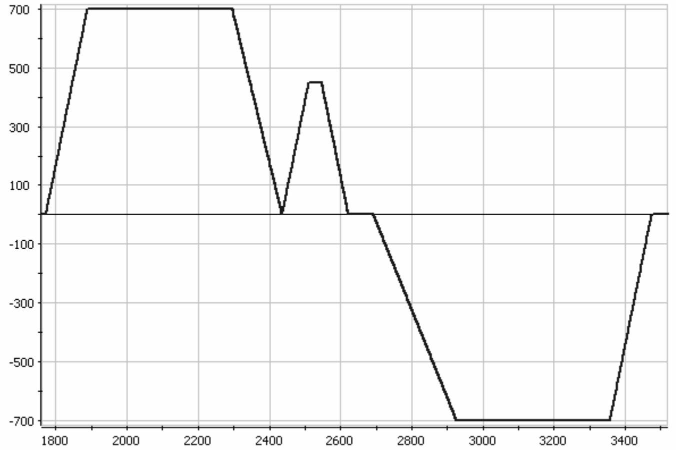

Fig. 11. A demanded trajectory of moulding form

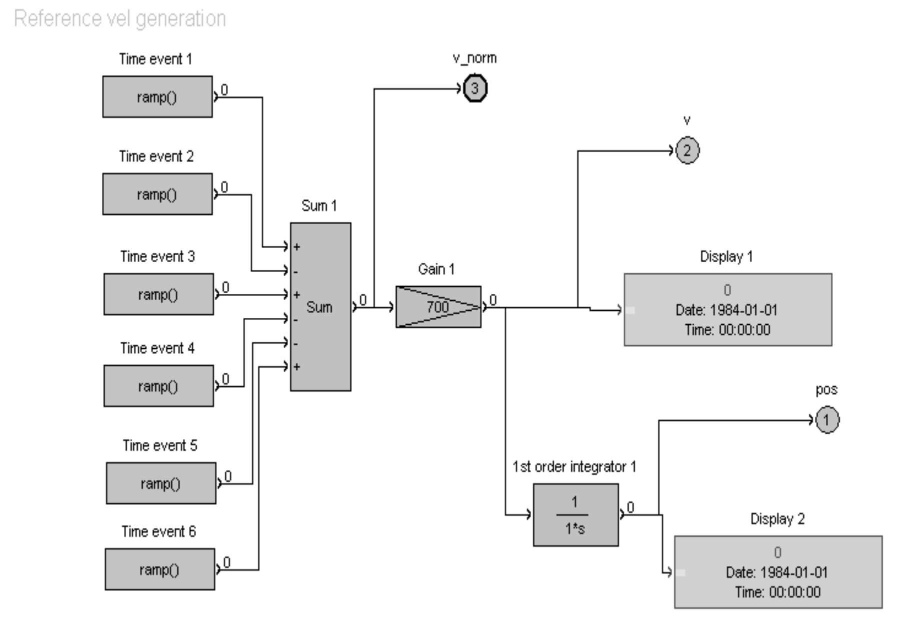

Fig. 12. A sub-path for generation of molding form trajectory

At the beginning of movement clamping unit is moving as fast as possible in order to reach a point, when the form touch injection part of machine. Because of possible damage of the form – the velocity of touch should be reduced to 0. Next, the clamping force is built up. After injection, the clamping unit should be opened, which is the second part of the figure. To realise the velocity and the related position, a reference signal was used as presented in Fig. 12.

Electric motors were modelled using analytical relations, describing the physical phenomena in machine. This model is implemented as a plugin in PExSim software. The model of internal friction in the motor is simplified to the form of a linear function of rotation speed. Each motor has its own dedicated DC amplifier modelled as sub-path, Fig. 13.

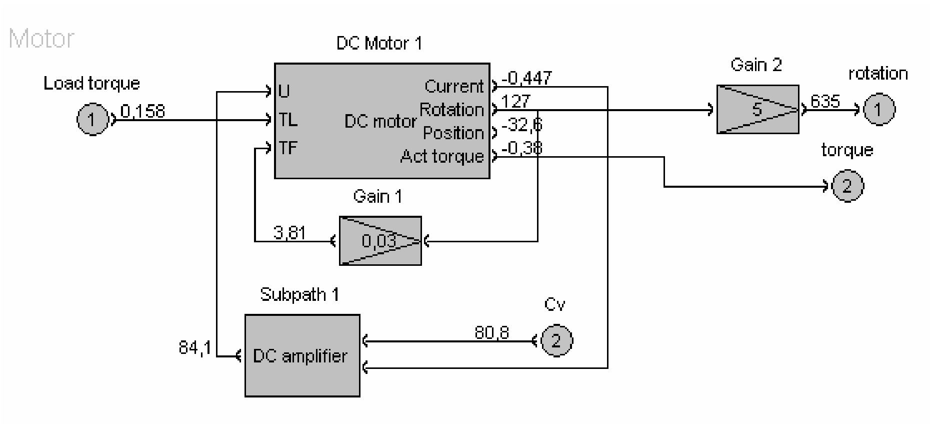

Fig. 13. A sub-path used for modelling of electric drive action

As the next part was a model of the mechanical components of the machine, which represent the efficiency and dynamic of constant stroke hydraulic pumps, piston reaction and the mechanic properties of toggle mechanism. The first part was a model of pumps driven by motors. Its dynamics was dependent on the motors' rotations, the actual position of form and the hydraulic connection between pumps (at closing of the form both motors are acting together), Fig. 14.

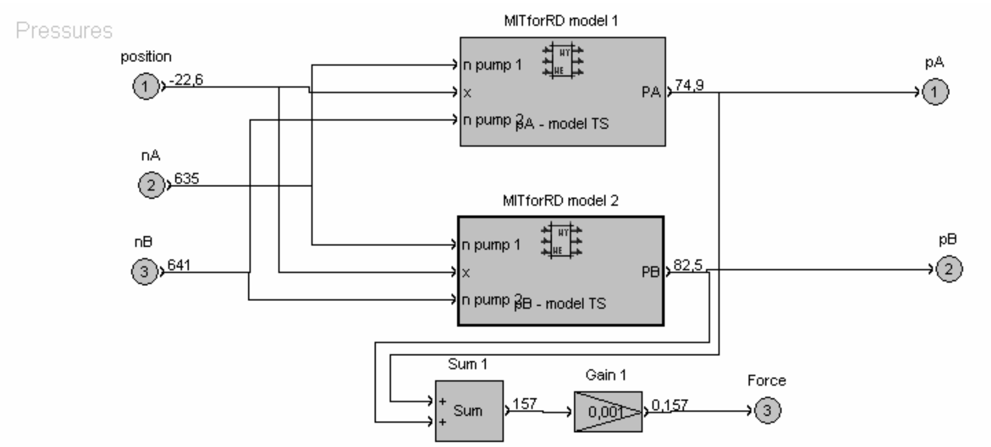

Fig. 14. Model of the dynamics of pump set driven by motors

Knowledge of pressure values pA and pB in the plunger cylinders and all of the dimensions of the cylinders can be sufficient for the calculation of the movement of the piston; however, some phenomena, like friction forces or compression in hydraulic pipes (sometimes visible in form of fast oscillations), are not easy to represent. Therefore, the reaction of the plunger cylinder set was modelled by the MITforD package too. After the determination of the piston velocity model, (see Fig. 4) it was used as an integrator for the calculation of actual position of the form, Fig. 15.

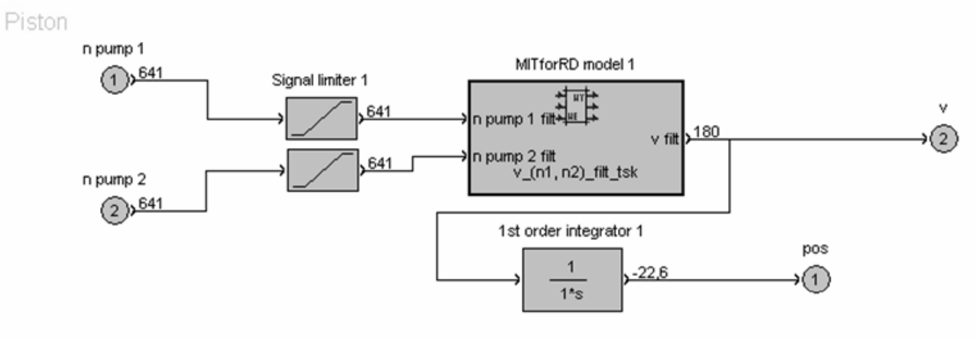

Fig. 15. Structure used for modeling reaction of piston set

The structure of the toggle mechanics, Fig. 9, is quite complex, but it can be after some effort modelled in an analytic way. The toggle can be treated as a nonlinear gearbox in respect to position. In this phase of the HSIM cycle, the form is steady, but the cylinder action has to built a clumping force in the toggle mechanism, through the proper movement of cylinder pistons x. It was easy to use MITforD for the modelling of its response, Fig. 16. This was not a trivial problem, because the construction of the toggle mechanism is not available at the moment, but it can be modelled by the application of specialised software (SimulationX) by the producer, after the specification of the final dimensions and construction parameters. The recorded simulation runs were used as data for the MITforD and finally a model Fig. 16 was determined.

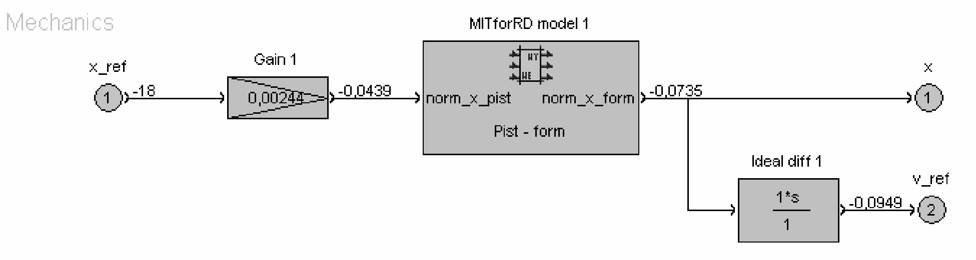

Fig. 16. Structure used for the modelling of the response of the toggle mechanism

A hierarchic structure of the modelling program, e.g. a sub-path, called "motor" represented on Fig. 10, uses a sub-path called "DC amplifier," Fig. 13, and this sub-path can be quite deep. PExSim can build up to 10 levels of subsequent sub-paths. Hence, the final structure can be quite simple and transparent and quite convenient to use.

Conclusions

The final structure of the system modelling a movement of a form was presented on Fig. 10. The simplicity of this structure did not reflect the real amount of process knowledge, inserted in sub-paths from Fig. 13 to 16 and others. They can be (and are) of quite a different nature and can easy cooperate together in a very convenient form. For example, the model of electric drives, Fig. 14, is of quite a deterministic nature and reflects different approaches to the control of form movement. The most respectable potential profits (in sense of fast and quite form control) are within the algorithm of drive control. The efficiency of pumps used in final construction can be measured and easy modelled in the form of a sub-path from Fig. 14. The unknown (but defined by the delivery of the toggle mechanism) structure of this mechanism can be modelled in the form of a nonlinear relation, Fig. 16, based on data delivered to designer of the drive algorithm from the simulation of mechanism's behaviour. The drive algorithm can be optimised and tested on the virtual model of non-existing complex mechanical system. It a short time, many experiments have been made and compared, using different modifications to the virtual system. Malfunctions of not proper control approaches do not disturb or destroy the designed construction or its parts and can give some clues for changes to be made in of the construction of the prototype. This approach enables the designer of the control and measurement equipment to continue his work while the producer is finishing the prototype. He can even suggest some modifications, e.g. inserting a cheap and robust magnetic sensor instead of classic, sometimes quite expensive instrumentation, or fine optic measurements of form position, etc. This is the future of the construction and design of advanced control systems.

Acknowledgements

Scientific work financed by the Ministry of Science and Higher Education and carried out within the Multi-Year Programme "Development of innovativeness systems of manufacturing and maintenance 2004–2008".

References

1. System documentation, http://www.mscsoftware.com/products/adams.cfm.
2. System documentation, http://www.simulationx.com/Default.aspx.
3. System documentation, http://www.mathworks.com/.
4. Wnuk P.: Identyfikacja obiektów w systemie AMandD. Pomiary Automatyka Kontrola, 2005, 9, 163–165 (in Polish).
5. Wnuk P.: Algorithms for fuzzy model structure identification, Ph.D dissertation. Warsaw University of Technology, 2004 (in Polish).
6. Janiszowski K.: Parametric models identification. Exit 2002, Warsaw, MMAR 2005, Międzyzdroje (in Polish).
7. Wnuk P.: The use of evolutionary optimization in fuzzy TSK model identification. Safeprocess, Bejing (China), 2006.
8. Janiszowski K.: Performance Index - a way for getting more of identification procedure.
9. Helduser S., Helbig A.: Electric hydrostatic injection moulding machine (EHIM). Final Report GRD1-2001-40454, Dresden, 2001.

Reviewer:
Tadeusz UHL

PExSim – nowe podejście do badania dynamiki złożonych procesów i urządzeń przemysłowych

Słowa kluczowe

Symulacja, modelowanie, identyfikacja, modele rozmyte.

Streszczenie

Niniejszy artykuł opisuje metodę badania dynamiki systemów w oparciu o łączoną symulację modeli tworzonych różnymi technikami. Zaproponowana metodologia jest przeznaczona głównie dla procesów przemysłowych. Artykuł opisuje elastyczne środowisko symulacji stworzone przez autorów w trakcie prac. Jednym z podstawowych założeń podczas tworzenia oprogramowania było umożliwienie prowadzenia symulacji w czasie rzeczywistym przy połączeniu z urządzeniem (układem sterowania za pomocą kart wejść) wyjść. Struktura pakietu umożliwia jego łatwe rozszerzanie dzięki mechanizmowi wtyczek. Wtyczki implementują bloki odpowiadające elementom układu sterowania, dokonujące przetwarzania sygnałów, opisujące elementy wykonawcze czy też części instalacji technologicznej na bazie równań fizykochemicznych lub identyfikowanego, nieliniowego modelu rozmytego. Oprogramowanie można stosować do symulacji instalacji technologicznych w celach testów działania systemów alarmowych i bezpieczeństwa, weryfikacji układów sterowania, przewidywania przyszłych stanów systemu czy też szkolenia załogi.