CZASOPISMO TECHNICZNE M
WYDAWNICTWO POLITECHNIKI KRAKOWSKIEJ

z. 3-M/2008
ISSN 0011-4561
ISSN 1897-6328

GRZEGORZ FILO*

DEVELOPING OF SOFTWARE FOR CARRYING OUT RESEARCH IN THE AREA OF HYDRAULIC CONTROL SYSTEMS

BUDOWA OPROGRAMOWANIA WSPOMAGAJĄCEGO BADANIA UKŁADU STEROWANIA HYDRAULICZNEGO

A b s t r a c t

In the work an attempt was made to develop our own computer software for carrying out investigations of a hydraulic cylinder positioning system. The software was built according to *UML* 2.0 modeling techniques and object oriented programming, using *Delphi* 2006 software development system. The developed software allows carrying out research of hydraulic systems with different types of digital controllers. The following types of control algorithms were tested: standard PID algorithm, fuzzy logic algorithms and neural networks. Computer simulations as well as laboratory tests were made. Laboratory tests were carried out at an especially built test stand. It follows from the results of investigations that using control techniques such as fuzzy logic or neural networks can be an advantageous option for conventional solutions.

Keywords: *hydraulic system, digital control, simulation, object oriented programing, software development process, software modelling, UML, software modelling language*

S t r e s z c z e n i e

Głównym zadaniem zrealizowanym w niniejszym artykule było zaprojektowanie i wykonanie własnego oprogramowania do prowadzenia badań układu pozycjonowania z siłownikiem hydraulicznym. Oprogramowanie zostało zaprojektowane zgodnie z zasadami technik modelowania obiektowego zawartymi w języku *UML* 2.0. Do implementacji wykorzystano środowisko programowania obiektowego *Delphi* 2006. Zbudowane oprogramowanie pozwala na prowadzenie badań układów hydraulicznych z różnymi typami i rodzajami sterowników cyfrowych. Z jego zastosowaniem przeprowadzono badania układu ze sterownikiem PID, sterownikami opartymi na logice rozmytej oraz wykorzystującymi sieci neuronowe. Przeprowadzono zarówno badania symulacyjne, jak również laboratoryjne, na specjalnie przygotowanym stanowisku badawczym. Z rezultatów badań wynika, że zastosowanie nowych technik sterowania, jak logika rozmyta, może w pewnych przypadkach znacząco wpłynąć na poprawę jakości sterowania w porównaniu z rozwiązaniami konwencjonalnymi.

Słowa kluczowe: *układ hydrauliczny, sterowanie cyfrowe, badania symulacyjne, programowanie obiektowe, projektowanie oprogramowania, modelowanie oprogramowania, język UML*

* Grzegorz Filo, PhD, Institute of Applied Informatics, Cracow University of Technology.

1. Introduction

Nowadays, in heavy duty machines we often have a connection between machine, digital computer controller and the software. Concerning hydraulic systems in these machines we can see that until recently they were mainly controlled by mechanical or electrical analog regulators. Now, analog systems are replaced with faster, more reliable and more accurate digital computer regulators. Using digital regulators also allows application of more complex, advanced control techniques, as fuzzy logic or neural networks. However, application of these new control techniques requires carrying out a series of simulation and laboratory tests.

Before tests are begun, it is necessary to answer the question whether to use commercial software or develop one's own software. There are many commercial simulation systems on the market. *Matlab-Simulink*, *VisSim* or 20 – *sim* can be mentioned, among others. Using commercial software is easier and does not require from the researcher informatic skills the researcher. Developing one's own software is more difficult and requires computer programing skills, but brings valuable advantages. For example, it allows automating research process, significantlly decreases computation time and allows integration of various functionalities, such as aiding numerical simulation as well as laboratory tests, within one software package.

In the present work an attempt was made to develop our own computer software for carrying out investigations of a hydraulic cylinder positioning system. The software, named *SimExpert*, was used to carry out simulations and aid laboratory tests of a hydraulic system with various types of digital controllers, using advanced control techniques such as fuzzy logic or neural networks. In the developing process *UML* 2.0 (*Unified Modeling Language*) [8] and *Delphi* software development system were used [1, 3].

2. Project of the *SimExpert* software

2.1. Functional and non-functional requirements

The first step in developing process was specifying the requirements for the system. The main requirements which should be met by developed software are to:
– include a mathematical model of hydraulic system,
– include models of controllers: with *PID* algorithm, with fuzzy logic and with neural networks,
– allow training neural networks with the backpropagation algorithm,
– allow carrying out simulations of the system with various values of parameters,
– present results of research in numerical and graphic forms,
– implement interface to *Advantech PCL*-1711 laboratory card,
– generate and sends control signals to hydraulic control elements,
– acquire and saves data from transducers installed in the hydraulic system.

2.2. Model of hydraulic system

SimExpert software includes a model of hydraulic system presented in Fig. 1.

The hydraulic system shown in Fig. 1 was built using standard hydraulic elements [5, 6]. It consists of the following components: 1 – supply unit with variable pump delivery, 2, 10 – proportional relief valves, 3 – control valve, 4 – hydraulic cylinder, 5, 6 – pressure transducers, 7 – position transducer, 8 – computer control and data acquisition system, 9 – supply unit with constant pump delivery.

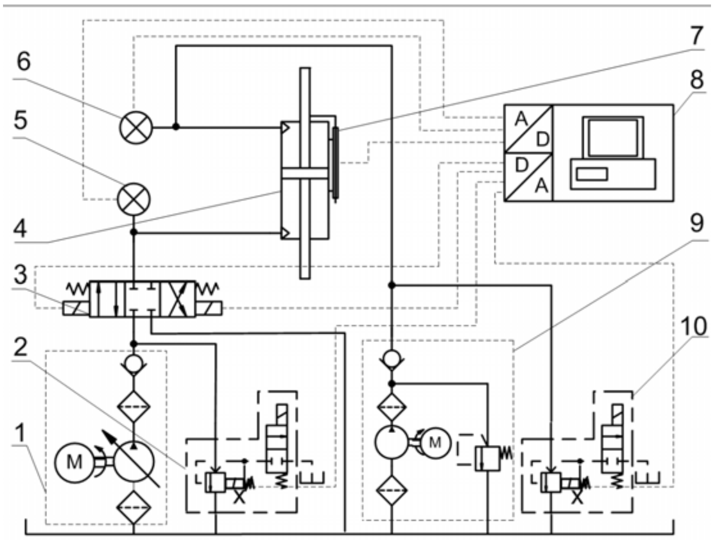

Fig. 1. The modelled hydraulic system

Rys. 1. Schemat modelowanego układu hydraulicznego

2.3. Models of digital regulators

In *SimExpert* software five models of digital regulators were included: two based on fuzzy logic, two with neural networks and one with *PID* algorithm. The fuzzy models consist of three parts: fuzzification, inference and defuzzification [2, 4]. The models of regulators differ from each other in the method of defuzzification. In the first one *Height Method* (*HM*) was applied. According to this method, the output signal is computed using formula [4]

$$u = \left(\sum_{i=1}^{N} \mu_{Zi} \cdot Y_i \right) \Big/ \left(\sum_{i=1}^{N} \mu_{Zi} \right) \tag{1}$$

where:

- N – number of fuzzy sets defined for output signal,
- μ_{Zi} – activation of fuzzy sets defined for output signal,
- Y_i – modal values of fuzzy sets.

In the second model the *Center of Sum* (*CoS*) defuzzification method was used. In this method the output signal is computed in the following way [4]

$$u = \left(\int x \cdot \mu(x)dx\right) \Big/ \left(\int \mu(x)dx\right) \tag{2}$$

where:

x – computational domain of output signal u,

μ – summary activation function.

A model of neural network was developed with the following assumptions: the modelled network was feed-forward type without signal feedbacks. The network was built of nonlinear neurons with hyperbolic tangent activation function [7]. The network consisted of four layers of neurons: input, two hidden and output. The network was trained using the backpropagation algorithm with *momentum* parameter.

2.4. The *UML* diagrams of the software

Software development process will be shown on the basis of several *UML* diagrams [8].

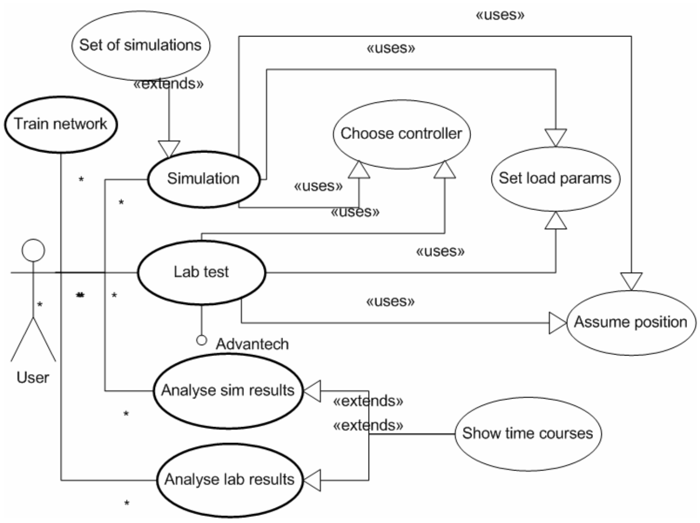

Fig. 2. The use-case diagram of modelled software

Rys. 2. Diagram przypadków użycia modelowanego oprogramowania

The use-case diagram shows the functionality of the developed software from the user's point of view [8]. The diagram defined for the software is presented in Fig. 2. As follows from the figure, software realizes five main functionalities for the user: training neural network, carrying out simulations and laboratory tests, and analysing the results of simulations and laboratory tests. Both simulations and laboratory tests include configuration settings: choosing of controler type, setting up load params and setting assumed position of hydraulic cylinder. In laboratory test interface to *Advantech* card must also be implemented.

The analysis of results can be extended by presenting the results in a graphic form as time courses of chosen parameters.

As follows from the object analysis of the software, it was convenient to develop separate applications for realizing individual functionalities. This partition is shown in *UML* deployment diagram (Fig. 3).

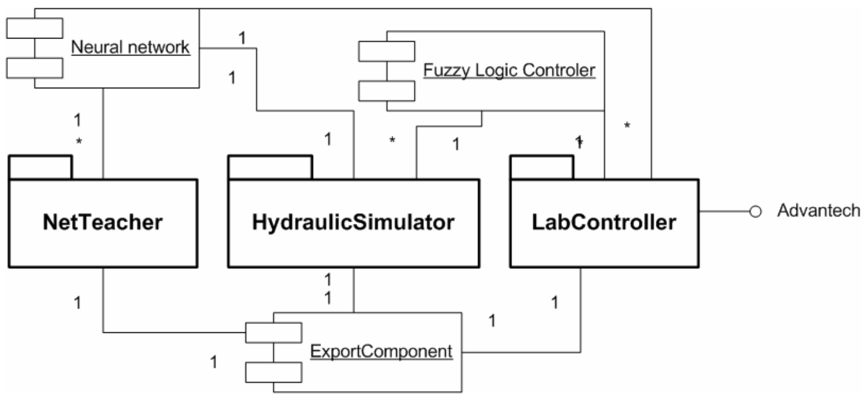

Fig. 3. Deployment diagram of modelled software

Rys. 3. Diagram implementacji modelowanego oprogramowania

As shown in Fig. 3, *SimExpert* software consists of three applications: *NetTeacher* for training neural networks, *HydraulicSimulator* for carrying out simulations and *LabController* for aiding laboratory tests. In each application components which include models of neural networks, models of fuzzy logic controllers and export of data are used.

Class diagram for *HydraulicSimulator* application is shown in Fig. 4.

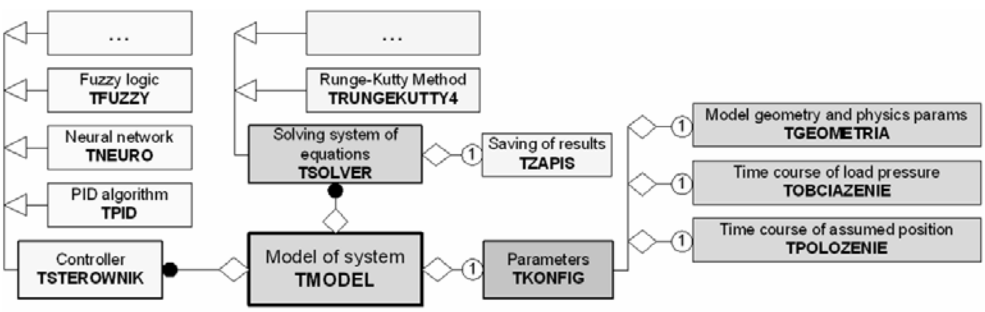

Fig. 4. Class diagram of modelled software

Rys. 4. Diagram klas modelowanego oprogramowania

3. Program of experiments

SimExpert software was used to carry out experiments of a hydraulic system using the adopted research plan. The plan includes Cartesian product of the following combinations of system components:

– five regulators: two based on fuzzy logic, two with neural networks and one *PID*,
– nine types and values of pressure in load line: constant values from 2 MPa to 10 MPa, linear increase, linear decrease, rectangular course and triangular course,
– two assumed positions of hydraulic cylinder: simple (one assumed position during the experiment) and advanced (two assumed positions during the experiment).

The Cartesian product of specified parameters gives us $5 \times 9 \times 2 = 90$ experiments. The same plan was used in simulations and laboratory tests. During each experiment the values of several parameters for estimating the control efficiency were obtained. These parameters were:

– time of obtaining the assumed position t given in seconds,
– mean deviation from the assumed position e in milimeters,
– sumarized cost of control K (without unit),
– parameter called *IAE* (*Integral Area Error*).

4. Results of experiments

4.1. Results of simulations

In Figure 5 results of example simulation are presented. The values of parameters assigned for this simulation were as follows: pressure in load line p_B = 9 MPa, applied regulator: fuzzy logic regulator with defuzzification method *HM*, assumed position of cylinder rod: 500 mm.

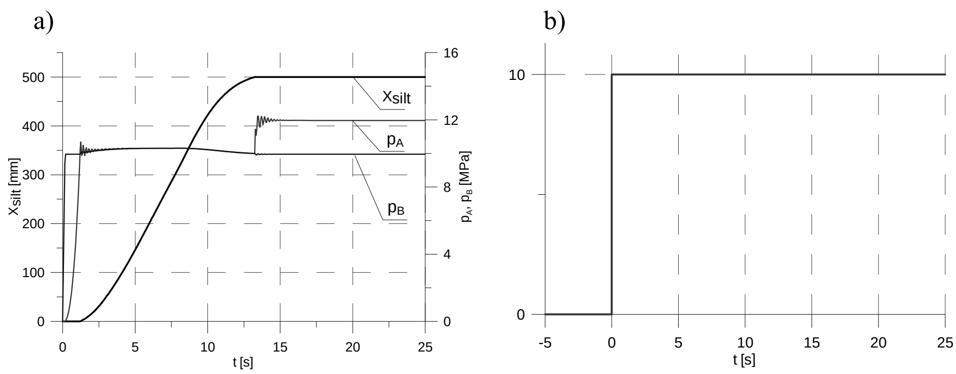

Fig. 5. Time courses obtained in example simulation: a) input function, b) system answer: x_{silt} – position of cylinder rod, p_A, p_B – pressures in supply line and load line

Rys. 5. Przebiegi czasowe uzyskane w przykładowej symulacji: a) wymuszenie, b) odpowiedź układu: x_{silt} – położenie tłoczyska, p_A, p_B – ciśnienia w liniach zasilającej i obciążającej

In this case the assumed position of the hydraulic cylinder rod was obtained in time t = 13,9 s, mean deviation from the assumed value was e = 0,197 mm, cost of control K = 509,7 and the parameter IAE = 3,434.

4.2. Results of laboratory experiments

In Figure 6 the results of example laboratory experiment are presented. In this experiment the following values of parameters were assigned: pressure in load line p_B = 10 MPa, applied regulator: fuzzy logic regulator with defuzzification method *CoS*, assumed position of cylinder rod: 500 mm.

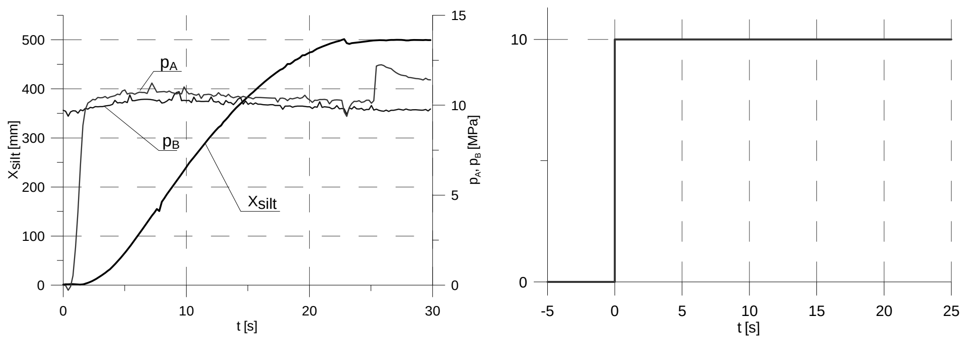

Fig. 6. Time courses obtained in example laboratory test: a) input function, b) system answer: x_{silt} – position of cylinder rod, p_A, p_B – pressures in supply line and load line

Rys. 6. Przebiegi czasowe uzyskane w przykładowym eksperymencie laboratoryjnym: a) wymuszenie, b) odpowiedź układu: x_{silt} – położenie tłoczyska, p_A, p_B – ciśnienia w linii zasilającej i obciążającej

In this case the assumed position of hydraulic cylinder rod was obtained in time t = 22,6 s, mean deviation from the assumed value was e = 0,449 mm, cost of control K = 570,9 and the parameter IAE = 5,446.

4.3. Summary of research results

It follows from the results of all simulations and laboratory tests that the best results in the whole discussed domain were obtained while using a regulator based on fuzzy logic with the *HM* defuzzification method. A bit worse results were obtained by the fuzzy regulator with *CoS* defuzzification method. *PID* controller in the whole domain obtained the worst results. However, in a few cases when the input function was close to regulator's linearization point, the results were even better than the ones obtained by fuzzy units. One neural network obtained results similar to *PID* regulator, the seccond one could not even obtain the assumed position.

5. Summary

The main aim of the present work was to develop computer software for carrying out simulations and laboratory tests of a hydraulic control system. In developing process the *UML* 2.0 software modelling language and *Delphi* 2006 software development system, which is part of *Borland Developer Studio* 2006 were used. The developed software *SimExpert* allowed carrying out all assumed tests of hydraulic system with automation of many tasks. The experiments confirmed that the application of control techniques such as fuzzy logic can improve the efficiency of hydraulic systems control process. The software was developed according to rules of object oriented programming, so it can be easily extended or modified.

References

[1] Boduch A., *Delphi. Kompendium programisty*, Wydawnictwo Helion, Gliwice 2002.

[2] Czogała E., Pedrycz W., *Elementy i metody teorii zbiorów rozmytych*, PWN, Warszawa 1985.

[3] Daniluk A., *Borland Developer Studio 2006. Kompendium programisty*, Wydawnictwo Helion, Gliwice 2006.

[4] Piegat A., *Modelowanie i sterowanie rozmyte*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 1999.

[5] Stryczek S., *Napęd hydrostatyczny*, tom I: *Elementy*, WNT, Warszawa 1997.

[6] Stryczek S., *Napęd hydrostatyczny*, tom II: *Układy*, WNT, Warszawa 1997.

[7] Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza RM, Warszawa 1993.

[8] Wrycza S., Marcinkowski B., Wyrzykowski K., *Język UML 2.0 w modelowaniu systemów informatycznych*, Wydawnictwo Helion, Gliwice 2006.