

DESIGN OF A HYDRAULIC-MECHANICAL LOAD-SENSING SYSTEM USING OBJECT-ORIENTED MODELLING AND SIMULATION

Gunnar Grossschmidt

Institute of Machinery

Tallinn University of Technology

Ehitajate tee 5, 19086

Tallinn, Estonia

E-mail: gunnar.grossschmidt@ttu.ee

Mait Harf

Institute of Cybernetics

Tallinn University of Technology

Akadeemia tee 21, 12618

Tallinn, Estonia

E-mail: mait@cs.ioc.ee

KEYWORDS

Load-sensing system, multi-pole modelling, object-oriented programming environment, simulation.

ABSTRACT

This paper deals with the design of a hydraulic-mechanical load-sensing system using computer modelling and simulation in an object-oriented programming environment. As initial, the scheme of a hydraulic-mechanical load-sensing system of Bosch GmbH has been taken. A modified scheme of the system is proposed. The object-oriented mathematical model of the system is composed. The approach is based on using multi-pole models and signal-flow graphs of functional elements, enabling methodical, graphical representation of mathematical models of large and complicated chain systems. In this way we can be convinced of the correct composition of models. The high-level programming environment NUT is used as a tool for modelling and simulation. A procedure of adjusting the multi-pole model is proposed. Modelling and simulation of separate objects, subsystems and the whole system are discussed.

1 HYDRAULIC-MECHANICAL LOAD-SENSING SYSTEM

Fluid power systems, in which working pressure (pressure in pump output) is kept proportional to load, are called hydraulic load-sensing systems. Such systems are mainly used in mechanisms containing numerous drives to run with the purpose to save energy. The scheme of the hydraulic load-sensing system of Bosch GmbH is shown in Fig. 1, the scheme of hydraulic-mechanical controller in Fig. 2 and the section of the valve block in Fig. 3.

In Fig. 1, the variable displacement axial piston pump is driven by an electric motor M. Hydraulic-mechanical control of the pump volumetric flow is performed by control valve and hydraulic cylinder. The feeding chain of the hydraulic motor R_{Verbr} contains tube R_{L-zu}, pressure compensator R_{IDW}, measuring valve R_{VW}, check valve, meter-in throttle edge R_{SK-zu} and connection elements.

The output chain of the hydraulic motor R_{Verb} contains a meter-out throttle edge R_{SK-r}, and tube R_{L-ab}. The device contains load-sensing pressure feedback with resistance R_{LS}.

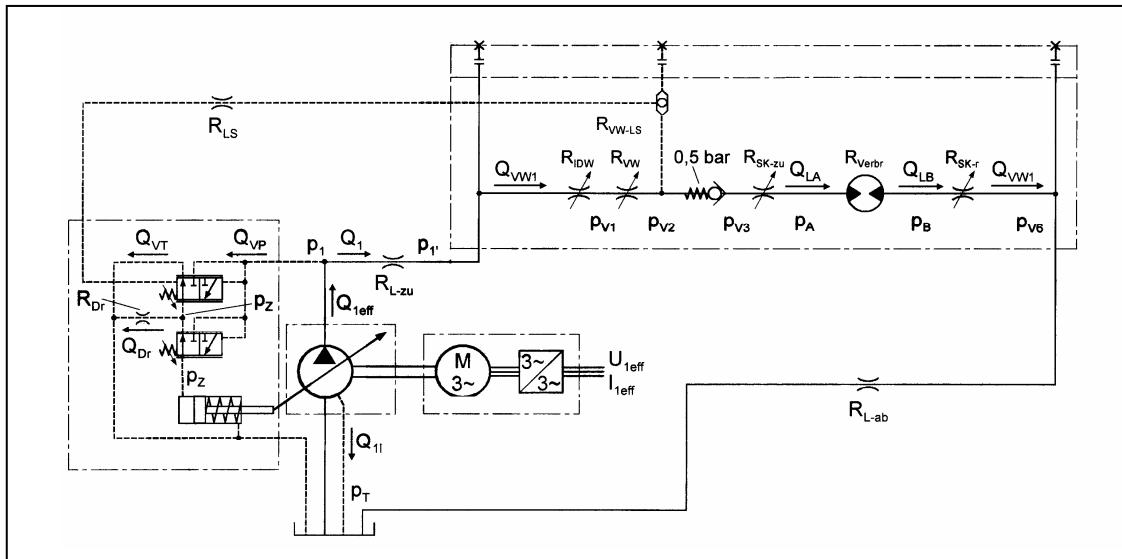


Fig. 1 - Scheme of the hydraulic-mechanical load-sensing system of Bosch GmbH.

The scheme of a hydraulic-mechanical controller (**Fig. 2**) contains a spool valve (effective area A_V) with inflow and outflow slots, constant resistor (volumetric flow Q_{Dr}), positioning cylinder (effective area A_z), and swash plate with spring.

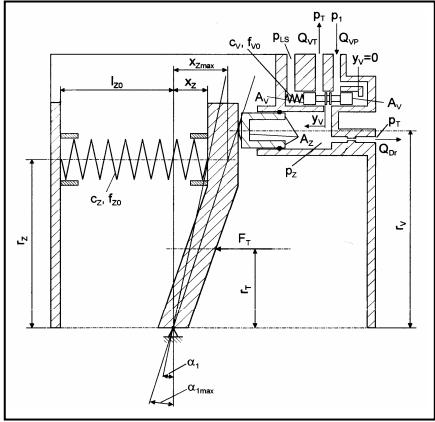


Fig. 2 - Scheme of a hydraulic-mechanical controller.

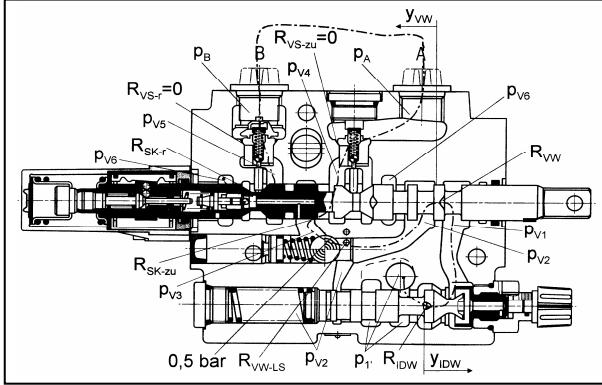


Fig. 3 - Section of the valve block.

The main part of the valve block (**Fig. 3**) is a directional valve with measuring throttle edge R_{vw} , meter-in throttle edge R_{vs-zu} and meter-out throttle edge R_{sk-r} . The valve block also contains pressure compensator with throttle edge R_{idw} and check valve with pressure drop 0.5 bars.

Examples of modelling and simulation of this load-sensing system were discussed in (Grossschmidt et al. 2006). The load-sensing system is quite complicated and contains several feedbacks. Iterations that were used in the simulation process turned out to be unstable. The resulting graphs appeared to be not smooth.

To get better results we need to improve the model. The main reason of instability is that the feeding pressure of the controller was taken from the output of a variable displacement hydraulic pump. But the pressure in the pump output depends on the resistance of the whole system. Pressure drops in connecting tubes of feedback chains affect the system behaviour as well.

2 MODIFIED LOAD-SENSING SYSTEM

A modified scheme of load-sensing system is proposed, in which the controller has an independent constant pressure feeding. Feedback pressures have been taken directly from the measuring valve with pressure compensator R_{idvw} .

The scheme of modified load-sensing system is shown in **Fig. 4**.

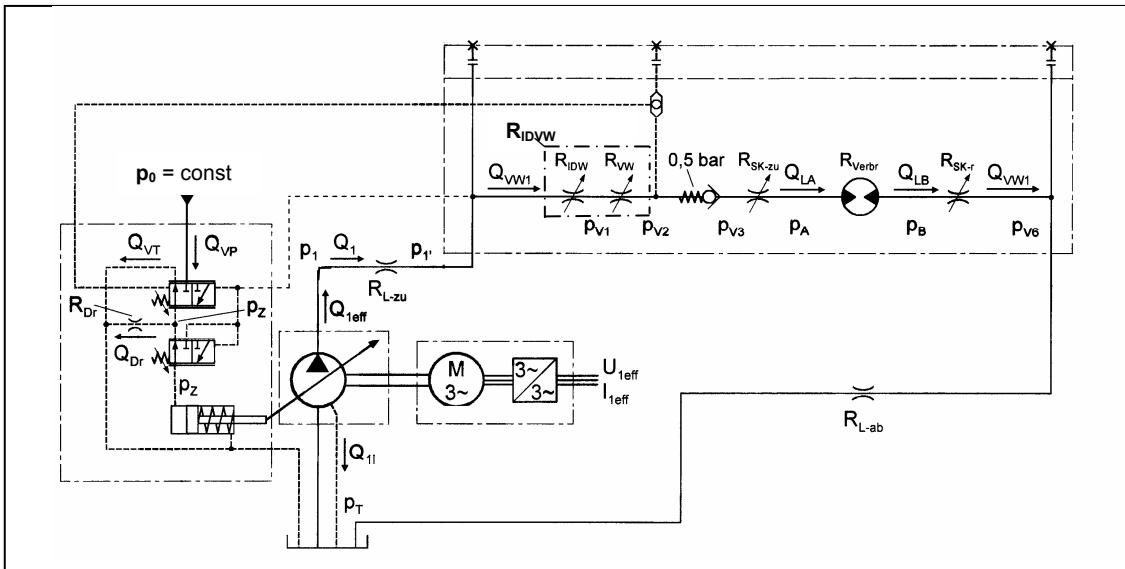


Fig. 4 - Scheme of the modified hydraulic-mechanical load-sensing system.

3 NUT PROGRAMMING ENVIRONMENT

The NUT system is a programming tool, which supports declarative programming in a high-level language, automatic program synthesis and visual programming (Tyugu et al. 1997). The NUT programming language rests on two paradigms: procedural object-oriented programming and the automatic synthesis of programs from declarative specifications. The NUT language is object-oriented. Concepts in it are specified as classes, and then used either in computations or for specifying new concepts. The description of NUT classes may contain specifications of their components, methods, initial values and other properties. Due to an equation solver built into the language processor, the system is able to interpret arithmetic equations as multi-way procedures for computing the unknown components of the equation. Each class can have a visual representation as well, so that specifications can be described visually.

Automatic synthesis of programs is a technique for the automatic construction of programs from the knowledge available in specifications of classes. Having a specification of a class, we are, in general, interested in solving the following problem: find an algorithm for computing the values of components y_1, \dots, y_n from the values of components x_1, \dots, x_m . The automatic synthesis of programs, as practised in NUT, is based on proof search in intuitionistic propositional logic.

The NUT graphics facilities include Graphics Editor, a set of graphics functions in the language, and the Scheme Editor. The Scheme Editor is a tool for visual programming that allows the user to define and use classes by means of graphical schemes. In order to draw schemes of problem descriptions, we have, for each class, an icon in the palette and an image, which will represent an object in a scheme. There are number of built-in features of the scheme editor, which support visual programming.

4 MODELLING AND SIMULATION OF FLUID POWER SYSTEMS

A number of packages for different fluid power systems modelling and simulation have been implemented in the NUT system (Grossschmidt et al. 2000, 2003). Multi-pole models of functional elements have been described as NUT classes together with their icons and images. Using visual specifications of described multi-pole models one can graphically compose models of various fluid power systems. When solving specific simulation problem, the model has to be adjusted by evaluating different parameters of the elements and adding sources to elements of the model that describe disturbances of the necessary shape and values.

During the simulation, some elements of the model need parameters, the values of which cannot be computed at the moment they are required. For computing values of such parameters, a special iteration method has been used. When starting the process, approximate values of critical parameters have been given as initials. At each step of the simulation process we try to refine initial approximate parameters. We use the NUT system to synthesize programs for re-computing some parameters and try to re-compute them iteratively until precise values of the parameters have been attained.

It is possible to solve a great number of various computing tasks on each fluid power system evaluating some components as inputs and computing some other components as outputs.

The whole computing process is organised by supporting class "process". State variables are introduced for every functional element to characterize the features of the element at the current simulation time step. The simulation process starts from the initial state and includes calculation of following state (*nextstate*) from previous states (from *oldstate* and *state*). Final state (*finalstate*) is computed as a result of simulation.

5 COMPOSING MODEL OF THE MODIFIED LOAD-SENSING SYSTEM

The multi-pole model (Fig. 5) represents the scheme of the modified load-sensing system (Fig. 4).

To build up the multi-pole model it is necessary to decompose the scheme of the load-sensing system into logical components and subsystems.

Models of the following components of the load-sensing system have been developed: variable displacement axial piston pump, electric motor, hydraulic-mechanical controller, valve block, hydraulic motor, tubes and multiple tube connection elements.

Model of the hydraulic-mechanical controller includes models of control valve, meter-in throttle edge, meter-out throttle edge, resistor and positioning cylinder with swash plate.

Model of the valve block includes a measuring valve with pressure compensator, check valve with meter-in throttle edge and meter-out throttle edge of the hydraulic motor.

The multi-pole model of the whole load-sensing system has been built up using the components models. First, necessary components have been connected through poles. Second, variables of connection poles have been defined as inputs or outputs for every component depending on required causalities (Grossschmidt et al. 1998).

All the models of the load-sensing system components have been described as NUT classes together with their images and icons.

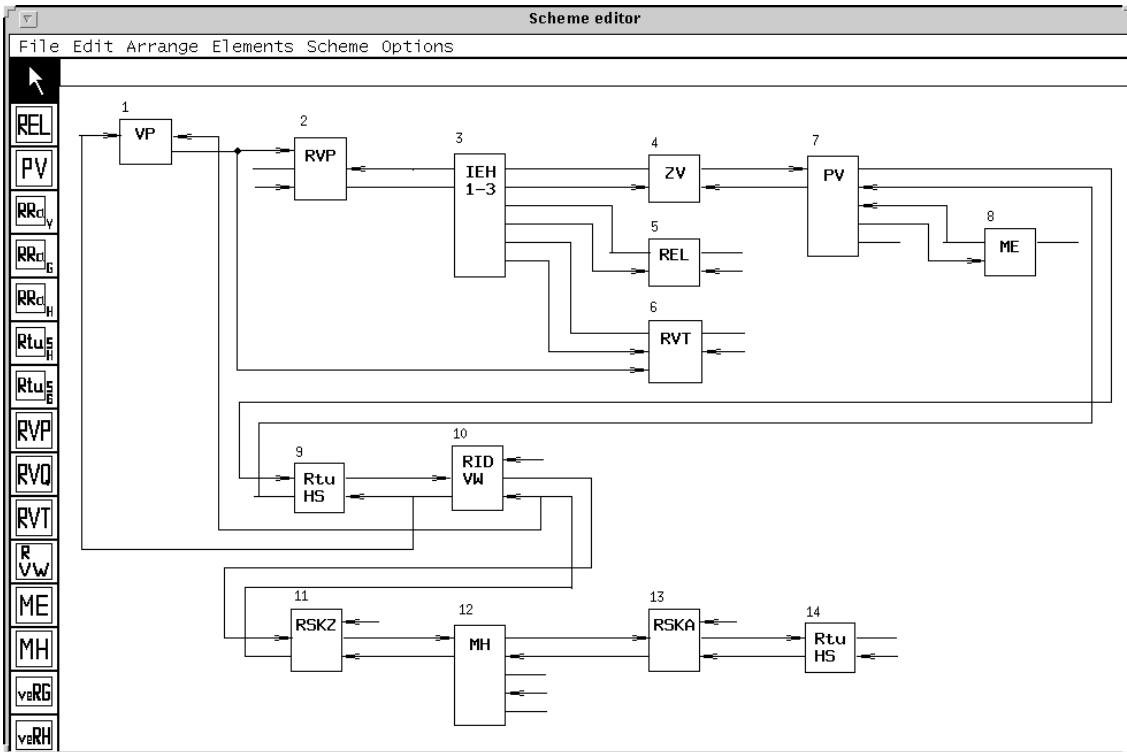


Fig. 5 - Multi-pole model of the modified hydraulic-mechanical load-sensing system.

Notations: 1 - Displacement of the control valve VP; 2 - Meter-in throttle edge RVP of the control valve; 3 - Hydraulic interface element IEH1-3; 4 - Positioning cylinder ZV; 5 – Constant resistor REL; 6 - Meter-out throttle edge RVT of the control valve; 7 - Variable displacement pump PV; 8 - Electric motor ME; 9 - Tube RtuhS; 10 - Measuring valve with pressure compensator RIDVW; 11 - Meter-in throttle edge for hydraulic motor RSKZ; 12 - Hydraulic motor MH; 13 - Meter-out throttle edge for hydraulic motor RSKA; 14 - Tube RtuhS.

6 SPECIFYING SYSTEM PARAMETERS

The most important stage in design of hydraulic load-sensing system is parameters specification. In order to help designer to find better solutions, computer simulation technology is proposed that includes following steps.

First, we need to choose hydraulic motor and hydraulic pump parameters.

Axial piston hydraulic motor with working volume $V = 40 \cdot 10^{-6} \text{ m}^3/\text{rev}$; axial-piston variable displacement pump maximum working volume $V_{\max} = 63 \cdot 10^{-6} \text{ m}^3/\text{rev}$, nominal rotation frequency $n_{\text{nom}} = 1475 \text{ min}^{-1}$ and maximum displacement angle of the pump swash plate $\alpha_{\max} = 0.3264 \text{ rad}$ have been taken.

Second, the fluid and its properties must be chosen.

The oil HLP46 has been taken (kinematic viscosity $\nu = 45 \cdot 10^{-6} \text{ m}^2/\text{s}$ at temperature $T = 40^\circ\text{C}$ and density $\rho = 890 \text{ kg/m}^3$ at temperature $T = 15^\circ\text{C}$). The working temperature $T = 40^\circ\text{C}$ has been chosen. Kinematic viscosity and density depending on the pressure are calculated at each step.

Third, we need to set up initial approximate values of pressures and pressure drops for pump control.

Maximum working pressure $p_{\max} = 250 \text{ bar}$, pressure drop in measuring valve $\Delta p_{IDW} = \sim 5 \dots 7 \text{ bar}$, pressure drop in measuring valve with pressure compensator $\Delta p_{IDVW} = \sim 14 \dots 19 \text{ bar}$, pump control system feeding pressure $p_0 = 120 \text{ bar}$ and pump control pressure $p_{CP} = \sim 7 \dots 25 \text{ bar}$ have been taken.

Fourth, we need to set up maximum displacements of the valves.

Maximum displacement of the pump controller valve $y_v = 0.003 \text{ m}$, maximum displacement of directional valve $y_{vw} = 0.007 \text{ m}$, maximum displacement of the pressure compensator $y_{IDW} = 0.007 \text{ m}$ have been taken.

Fifth, all the models of components must be tested separately. For every component the simulation problem must be composed and input signals must be chosen. Behavior of the component must be simulated. Initial approximate components parameters values (e.g. stiffness of springs, geometry of valves working slots, etc.) must be refined as a result of simulations.

Sixth, the separately tested components models must be connected into more complicated subsystems, tested in behavior and adjusted if necessary.

Seventh, the whole load-sensing system must be built up and simulation tasks must be solved.

7 SIMULATION OF SUBSYSTEMS

7.1 Measuring valve with pressure compensator

Simulation problem description of measuring valve with pressure compensator is shown in **Fig. 6**.

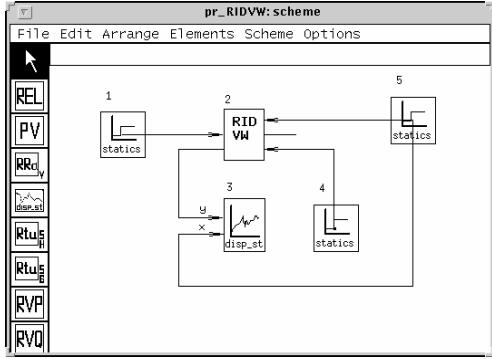


Fig. 6 - Simulation problem description of measuring valve with pressure compensator.

Notations: 1 - Input volumetric flows; 2 - Measuring valve with pressure compensator RIDVW; 3 - Drawing graphic; 4 - Constant input pressure at right port; 5 - Input displacements of the directional valve.

The measuring valve has two identical slots. Dependency of pressure drop in measuring valve on direction valve displacement is mainly influenced by these two slots profiles. Slot profiles have been designed to contain 6 linear fragments of different shape in order to achieve almost linear dependency.

The pressure compensator has four identical slots. Dependency of pressure drop in pressure compensator on direction valve displacement is influenced by these four slots profiles. Slot profiles have been designed to contain 15 linear fragments of different shape in order to achieve almost linear dependency. It was quite difficult to get linear dependency in the cases of minor displacements of the directional valve (**Fig. 7**).

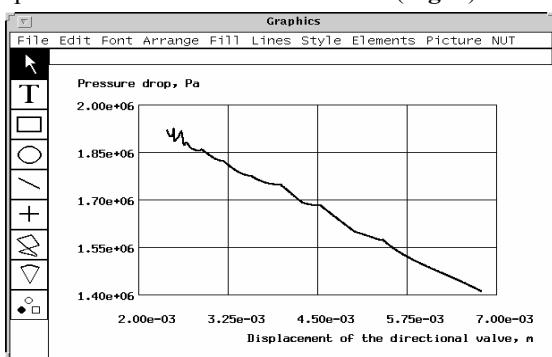


Fig. 7 - Simulated pressure drop in measuring valve with pressure compensator depending on the displacement of the directional valve.

7.2 Hydraulic-mechanical controller

The hydraulic-mechanical controller includes constant pressure feeding that enables to make feedback independent of pressure in hydraulic pump output.

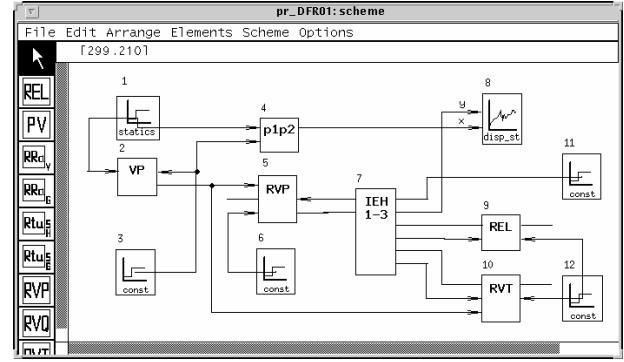


Fig. 8 - Simulation problem description of hydraulic-mechanical controller.

Notations: 1 - Pressures at the left port of VP; 2 - Displacement of the control valve VP; 3 - Constant pressure at the right port of VP; 4 - Calculation of pressure differences; 5 - Meter-in throttle edge RVP of the control valve; 6 - Constant feeding pressure of the pump controller; 7 - Hydraulic interface element IEH1-3; 8 - Drawing graphic; 9 - Constant resistor REL; 10 - Meter-out throttle edge RVT of the control valve; 11 - Zero volumetric flow to the positioning cylinder ZV; 12 - Constant pressure in outlet.

Pressure interval, required for pump volumetric flow regulation from *max* to *min* is $p_{CP} \approx 7 \dots 24$ bar. Calculations start from the right end of the graph (**Fig. 9**). The short vertical part in the curve is caused by approximate initial values of subsystem parameters.

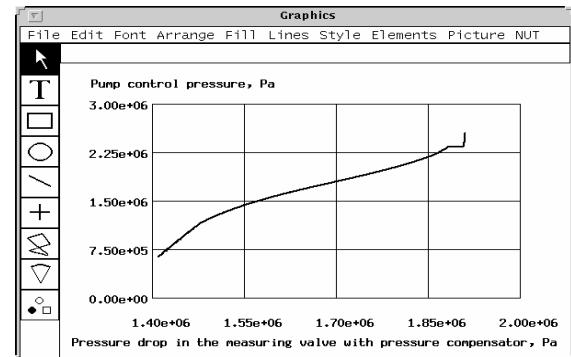


Fig. 9 - Simulated pump control pressure depending on the pressure drop in the measuring valve with pressure compensator.

7.3 Hydraulic motor subsystem

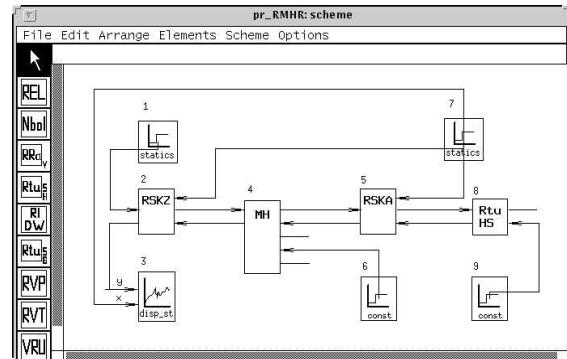


Fig. 10 - Simulation problem description of hydraulic motor subsystem.

Notations: 1 – Input volumetric flows; 2 - Meter-in throttle edge for hydraulic motor RSKZ; 3 - Drawing graphic; 4 - Hydraulic motor MH; 5 - Meter-out throttle edge for hydraulic motor RSKA; 6 - Constant load moment; 7 - Input displacements of the directional valve; 8 - Tube RtuHS; 9 - Constant pressure in outlet.

Meter-in and meter-out throttle edges ensure minor changes of subsystem inlet pressure in dependence of the displacement of the direction valve (**Fig. 11**).

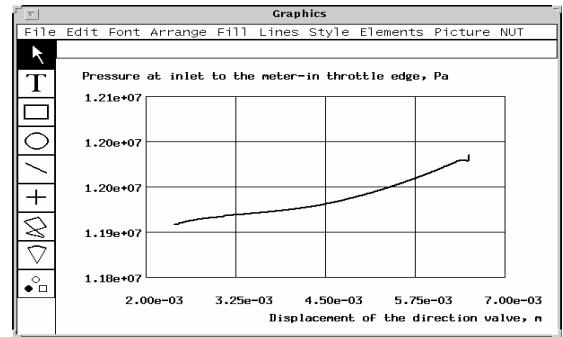


Fig. 11 - Pressure at inlet to the meter-in throttle edge of hydraulic motor depending on the displacement of the direction valve (load moment $M = 68.7 \text{ Nm}$).

8 SIMULATION OF THE WHOLE LOAD-SENSING SYSTEM

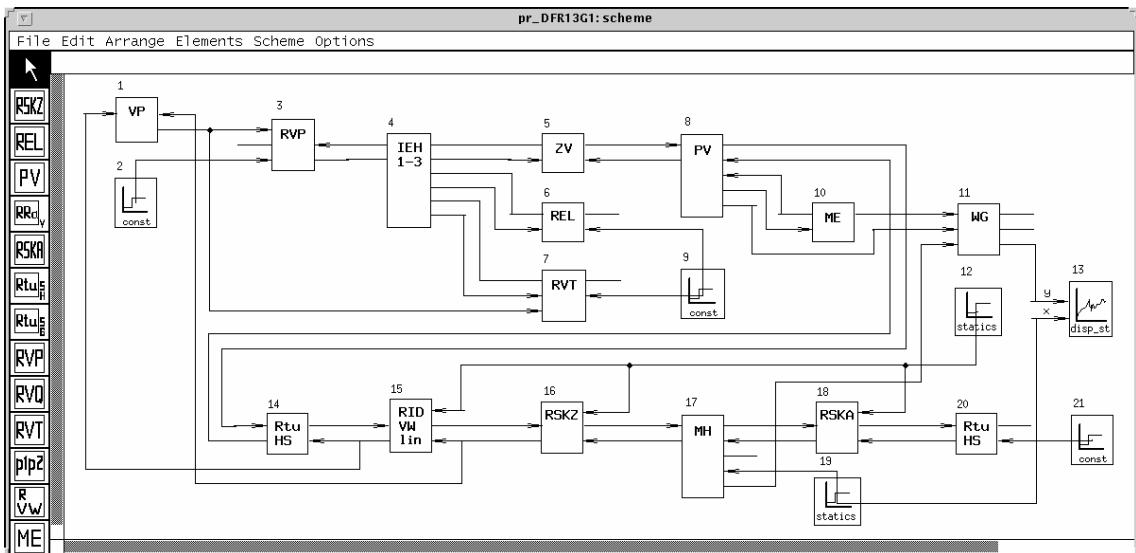


Fig. 12 - Simulation problem description of the whole load-sensing system (calculating efficiency coefficient depending on the static values of the directional valve displacement and hydraulic motor load moment).

Notations: 1 - Displacement of the spool valve VP; 2 - Constant feeding pressure for the pump controller; 3 - Meter-in throttle edge of the control valve RVP; 4 - Hydraulic interface element IEH1-3; 5 - Positioning cylinder ZV; 6 - Constant resistor REL; 7 - Meter-out throttle edge RVT of the control valve; 8 - Variable displacement pump PV; 9 - Constant pressure in outlet; 10 - Electric motor ME; 11 - Efficiency coefficients calculator WG; 12 - Input static values of the directional valve displacement; 13 - Drawing graphic; 14 - Tube RtuHS; 15 - Measuring valve with pressure compensator RIDVWlin; 16 - Meter-in throttle edge for hydraulic motor RSKZ; 17 - Hydraulic motor MH; 18 - Meter-out throttle edge for hydraulic motor RSKA; 19 - Input static values of the hydraulic motor load moment; 20 - Tube RtuHS; 21 - Pressure in outlet.

Angle velocity of the hydraulic motor depends on the pump volumetric flow (**Fig. 13**)

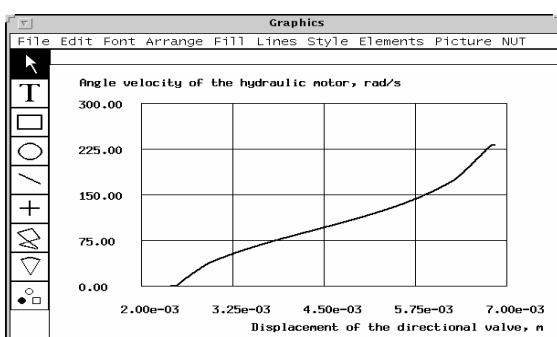


Fig. 13 - Simulated angle velocity of the hydraulic motor depending on the displacement of the directional valve (load moment of the hydraulic motor 68.7 Nm).

Increasing the hydraulic motor load moment the angle velocity decreases (**Fig. 14**). It is caused by volumetric losses in pump and motor.

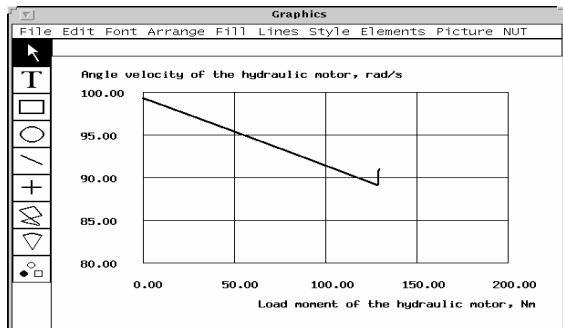


Fig. 14 - Simulated angle velocity of the hydraulic motor depending on the load moment of the hydraulic motor (displacement of the direction valve 0.0045 m)

Output power of the electric motor in the case of zero load moment of hydraulic motor is 1486 W (**Fig. 15**). Increasing the load moment causes almost linear increase of the output power of the electric motor.

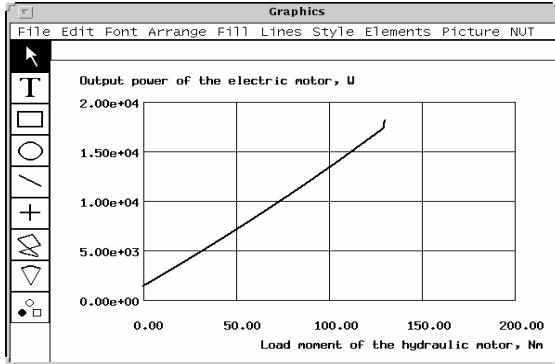


Fig. 15 - Simulated output power of the electric motor depending on the load moment of the hydraulic motor (displacement of the directional valve 0.0045 m).

Efficiency coefficient of the whole system is maximal if the load moment of the hydraulic motor is higher than $M_L \approx 50$ Nm (**Fig. 16**). In the cases of less load moment the efficiency coefficient falls.

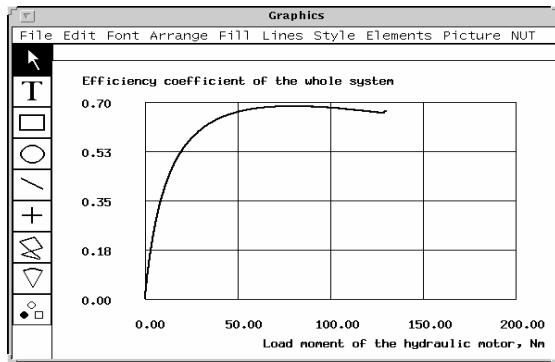


Fig. 16 - Simulated efficiency coefficient of the whole system depending on the load moment of the hydraulic motor (displacement of the directional 0.0045 m).

The output power of the electric motor is proportional to pump volumetric flow in the case of constant load moment of the hydraulic motor (**Fig. 17**).

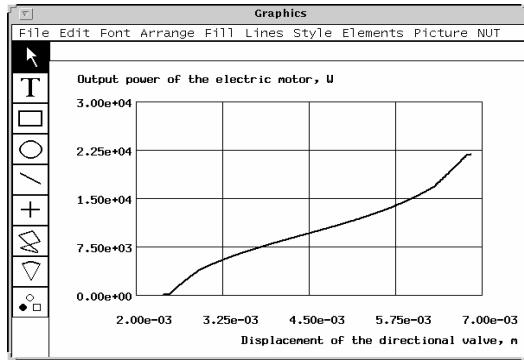


Fig. 17 - Simulated output power of the electric motor depending on the displacement of the directional valve (load moment of the hydraulic motor 68.7 Nm).

The efficiency coefficient of the whole system increases from ~ 0.65 to 0.73 when the pump volumetric flow is higher than $\sim 2.5 \times 10^{-4}$ m³/s (**Fig. 18**). In the cases of less pump volumetric flows the efficiency coefficient falls rapidly.

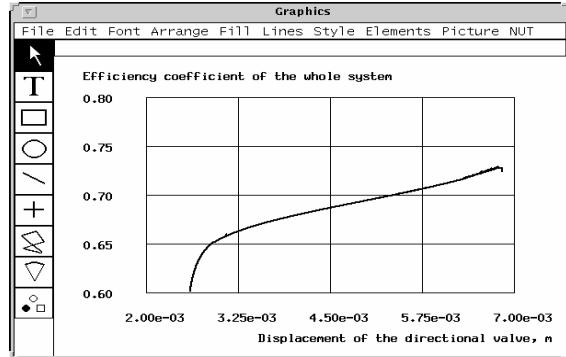


Fig. 18 - Simulated efficiency coefficient of the whole system depending on the displacement of the directional valve (load moment of the hydraulic motor 68.7 Nm).

9 SIZE AND COMPLEXITY

The modeling and simulation task of the whole load-sensing system considered in the paper contains:

- 21 classes, including 15 component classes;
- 923 variables;
- 9 variables that have to be iterated during the computations;
- 54 bindings between system components.

The automatically constructed problem-solving algorithm contains:

- 160 operators (equations, methods, assignments)
- 26 separate algorithms for solving subtasks.

All the simulations were performed on the Sun workstation in the UNIX environment.

CONCLUSIONS

Hydraulic load-sensing systems are complicated automatically regulating systems with number of components and several feedbacks. Feedbacks make the system very sensitive and unstable for simulation. A very precise parameter setting, especially for resistances of hydraulic valve spools, is required to make the system function.

A modified scheme of load-sensing system is proposed, in which the controller has an independent constant pressure feeding. Feedback pressures have been taken directly from the measuring valve with pressure compensator.

Using an object-oriented approach enables one in flexible way to compose and experiment with various large and complicated models. Automatic program synthesis allows describing and solving a great number

of simulation tasks in order to find out better solutions to design problems.

As a result of the current research, a simulation system is proposed that enables one to perform computer experiments at the first stage of design. Results of simulations can be used as initial data while building trial versions of real load-sensing fluid power systems.

The main features of the approach proposed in the paper are as follows.

- Mathematical models of the functional elements are composed as multi-pole models taking into account signal propagation in both directions.
- Used multi-pole models can have various causalities.
- The mathematical model of the fluid-power system contains models of functional elements and carries the full information about connections of input/output variables, which express the considered mathematical causalities and guarantees the completeness of the model.
- Modelling and simulation are built up in an object-oriented way using the NUT programming environment. This enables one to visualize and automate the simulation process.
- Simulation is performed step by step, starting from simulation of components and moving to more complicated subsystems.
- Calculations are performed separately for each multi-pole model. Iteration methods are used in cases of loop dependencies that may appear between component models when they are connected together into more complicated ones.

ACKNOWLEDGEMENT

This research was in part supported by the Estonian Science Foundation (Grant No. 7091).

REFERENCES

- Grossschmidt, G. and Vanaveski, J. 1998. "Causality of Mathematical Models of Technical Systems". Proc. of the 12th European Simulation Multiconference ESM'98, as part of the 50th anniversary celebrations of the University of Manchester, Simulation - Past, Present and Future, June 16-19, Manchester, United Kingdom, pp. 191-195.
- Grossschmidt, G.; Vanaveski, J. and Harf, M. 2000. "Simulation of Hydraulic Chains using Multi-pole Models in the NUT Programming Environment". Proc. of the 14th European Simulation Multiconference ESM'2000, Simulation and Modelling: Enables for a better quality of life, May 23-26, Ghent, Belgium, pp. 709-713.

Grossschmidt, G. and Harf, M. 2003. "Multi-pole Modelling and Simulation of an Electro-hydraulic Servo-system with a Non-linear Regulator in NUT Environment". Fünftes Deutsch-Polnisches Seminar "Innovation und Fortschritt in der Fluidtechnik", 18. -19. September, Warszawa, pp. 146-161.

Grossschmidt, G.; Harf, M. and Sallaste, T. 2006. "Modelling and Simulation of Fluid Power Systems in Object-Oriented Programming Environment". 8th Biennial ASME Conference on Engineering Systems Design and Analysis ESDA2006, July 4-7, Turin, Italy, [Proceedings], ASME, , paper ESDA2006-95387, 1-10.

Tyugu, E., Valt, R. 1997. "Visual programming in NUT". Journal of visual languages and programming, v. 8, pp. 523-544.

AUTHOR BIOGRAPHIES

GUNNAR GROSSSCHMIDT received his Dipl. Eng. degree from Tallinn University of Technology (1953) and Candidate of Technical Science degree (Ph.D) from the Kiev Polytechnical Institute (1959). His scientific interests are concentrated around modelling and simulation of fluid power systems for several decades. He has developed an original system for modelling of fluid power system.

His list of scientific publications contains over than 70 items. He has been lecturer in Tallinn University of Technology 50 years, as assistant, lecturer, associated professor and head of the chair of Machine Design.

MAIT HARF received his Dipl. Eng. degree from Tallinn University of Technology (1974), and Candidate of Technical Science degree (Ph.D.) from the Institute of Cybernetics, Tallinn (1984).

His research interests are concentrated around intelligent software design. He worked on methods for automatic (structural) synthesis of programs and their applications to knowledge based programming systems such as PRIZ, C-Priz, ExpertPriz and NUT.

He worked on Soviet new generation computer project Start (1985-1988). His research on developing intelligent software based on using hybrid knowledge bases and structural synthesis of programs in the Institute of Cybernetics at Tallinn University of Technology is continuing till now.