4th International DAAAM Conference
“INDUSTRIAL ENGINEERING – INNOVATION AS COMPETITIVE EDGE FOR SME”
29 – 30th April 2004, Tallinn, Estonia

SIMULATION OF STATICS AND STEADY STATE CONDITIONS OF AN ELECTRO-HYDRAULIC SERVO-SYSTEM

Grossschmidt, G.; Harf, M.

Abstract: *The paper presents the construction of multi-pole models of an electro-hydraulic servo-system and the simulation of its static and steady state characteristics. Using the NUT programming environment as a tool enables one to describe graphically multi-pole model classes, to automatically compose the algorithm, and to perform the simulations. The outputs of each multi-pole model are computed separately. In case of loop dependencies between multi-pole models, the iteration procedure for equalising the values of variables is used. The modifying of characteristics is observed.*
Key words: *electro-hydraulic servo-system, statics and steady state conditions, multi-pole models, simulation, NUT programming environment.*

1. INTRODUCTION

The electro-hydraulic servo-systems are in various applications, for example in steering mechanisms of aeroplanes, cars and ships, in numerical control drives of machine tools, robots and manipulators, in flight simulation systems, vibration machines for testing materials, in hydraulic presses, in automatic machines for processing plastics, in linear and rotation amplifiers, in synchronised drives, as so on.

The electro-hydraulic servo-system is controlled by an electro-hydraulic servo-valve. For position feedback an actuator displacement sensor is used. For creating and modifying the control signal, a regulator is used.

For static and steady state conditions of the electro-hydraulic servo-system, the following requirements can be pointed out:

- linear characteristics in given diapasons,
- low hysteresis and a high sensitivity,
- getting the needed force gain,
- getting the needed velocity gain.

For modifying of the characteristics is necessary to adjust its regulator. The adjusting values must be changed, depending of working parameters (load size and direction, velocity size and direction). Elaborating the algorithm for such a regulator for each servo-system experimentally is time consuming and expensive.

Simulation enables one to save time and reduce the expense of experimental work. The result of simulation depends on the adequatness of the mathematical models and upon the simulation method.

The most noun program packages for simulation of hydraulic and electro-hydraulic systems are:

ITI®-SIM	ITI GmbH, Dresden, Germany,
DSHplus	Fluidon GmbH, Aachen, Germany,
BATHfp	Bath University, UK,
HOPSAN	Linköping University, Sweden,
HYSYS	Mannesmann Rexroth AG, Germany,
MOSIHS	Mannesmann Rexroth AG, Germany,
Flowmaster	Fluid Dynamics International Inc., USA,
SIMU	Tampere University of Technology, Finland.

All this simulation systems enables to simulate the dynamic responses. The static characteristics were computed as going to end of the dynamic responses. Direct computing of the static and steady state conditions of complicated hydraulic systems with these packages is impossible. In this paper is proposed the direct method of modelling and simulation static and stationary characteristics of the complicated system, what is the electro-hydraulic servo-system.

For creating of the mathematical models of an electro-hydraulic servo-system multi-pole mathematical models of their elements and subsystems are used (Grossschmidt, 2002). Using such models of electro-hydraulic servo-system elements enables us to construct models and simulate in object-oriented way.

Constructing the model of the whole system includes:

- choosing multi pole models of the elements with the necessary causality;
- connecting the necessary output poles to input poles of the different multi-pole models to get the desired functionality of the whole model.

Using the NUT environment as a tool enables one to describe graphically the simulation problem and automatically compose the algorithm and perform computations (Uustalu, 1994).

2. NUT PROGRAMMING ENVIRONMENT

The NUT system is a programming tool, which supports declarative object-oriented programming in a high-level language, visual programming and automatic program synthesis (Uustalu, 1994). The NUT language is object-oriented. Concepts in it are specified as classes. The text of a class includes of different sections specifying superclasses, components, relations and initialisations of class components. 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. This enables one to use classes by means of visual programming.

Automatic synthesis of programs is a technique for the automatic construction of programs from the knowledge available in specifications of classes. 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 a Scheme Editor. The NUT Scheme Editor gives a number of built-in features, which support visual programming:

- connection lines between poles which represent binding the poles;
- an interactive zoom-in window, which can be used for showing or editing any object of a scheme;
- request for computing variables of a scheme.

A package for electro-hydraulic servo-system modelling and simulation is implemented in the NUT system. All the used multi-pole models of functional elements are specified as NUT classes together with their icons and images.

Besides multi-pole model classes, several supporting classes as **“const”** for constant input, **“statics”** for static input values, **“display_static”** for graphical displaying results are specified. The whole computing process is organised by supporting class **“static_process”**. Class **“balance”** organises the iteration procedure of variables in connection points of multi-pole elements. For **“static_process”** it is necessary to give the initial and the final value of the argument and a number of the calculating points.

Using visual specifications of multi-pole models one can graphically construct models of various systems. One can specify and easily solve a great number of various computing problems on each system, evaluating some variables as inputs and trying to compute some other variables or trying to compute all the variables which can be computed in this system.

3. USED MULTI-POLE MODELS

For modelling of statics and stationary motion the following multi-pole models of functional elements are used (Fig. 1 - 6).

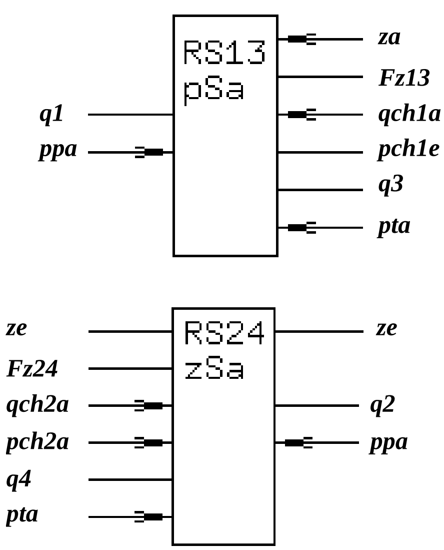

Fig. 1. Multi-pole models **RS13pSa** and **RS24zSa** of servo-valve slot pairs.

The inputs are denoted with arrows and their notations are signed with “a” The outputs are denoted without arrows. The variable notation with “e” is an iteration variable.

The multi-pole models **RS13pSa** and **RS24zSa** of servo-valve slot pairs have the following input and output variables:

za, ze displacement of servo-valve from initial position,
Fz13, Fz24 hydrodynamic force of jets trough slots 1, 3 and through slots 2, 4,
q1, q2, q3, q4 volumetric flows through slots 1, 2, 3 and 4,
qch1a, qch2a volumetric flow at connection into/out of hydraulic cylinder left and right chamber,
pch1e, pch2a pressure at the connecting port to left and right chamber of hydraulic cylinder,
ppa, pta pressure at the supply port and pressure at the return port.

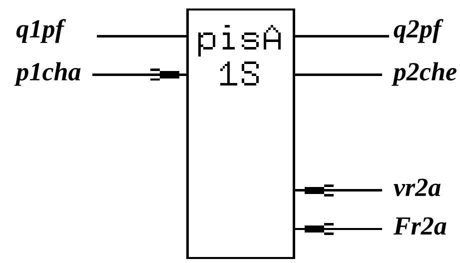

Fig.2. Multi-pole model **pisA1S** of piston, where

q1pf, q2pf volumetric flow at the left end and at the right end of a piston,
p1cha, p2che pressure in the left and in the right chamber of a cylinder,
vr2a piston rod right end velocity,
Fr2a force at the piston rod right end.

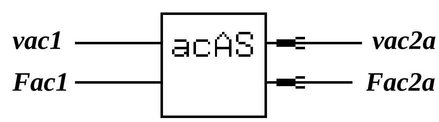

Fig.3. Multi-pole model **acAS** of actuator, where

vac1, vac2a actuator left and right end velocity,
Fac1, Fac2a force at the actuator left and right end.

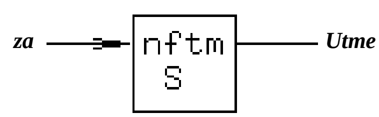

Fig.4. Multi-pole model **nftmS** for nozzle-and-flapper valve with torque motor, where

za displacement of the servo-valve from initial position,
Utme voltage to the torque motor.

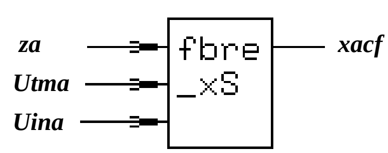

Fig.5. Multi-pole model **fbre_xS** for feedback and regulator, where

za displacement of the servo-valve from initial position,
Utma voltage to the torque motor,
Uina input control voltage,
xacf actuator position without influence of servo-system elasticities.

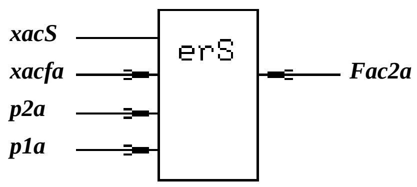

Fig.6. Multi-pole model **erS** for servo-system elasticities, where

xacS actuator position with influence of servo-system elasticities,
xacf actuator position without influence of servo-system elasticities,
p1a, p2a pressure in the left and in the right chamber of a cylinder,
Fac2a force at the actuator right end.

4. SIMULATION PROBLEM DESCRIPTIONS

For statics and the steady-state motion the following problem descriptions are composed:

- computing ***xac = F(Uin)***, by ***Fac = const, vac = const;***
- computing ***xac = F(Fac)***, by ***Uin = const, vac = const;***
- computing ***xac = F(vac)***, by ***Uin = const, Fac = const;***
- computing ***z = F(Fac)***, by ***Uin = const, vac = const;***
- computing ***z = F(vac)***, by ***Uin = const, Fac = const.***

It is also possible to compose the problem descriptions for computing of the characteristics of other variables, which depend on ***Uin, Fac*** or ***vac***. The problem description for computing of characteristics ***xac = F(Fac)***, by ***Uin = const, vac = const*** is shown in Fig.7.

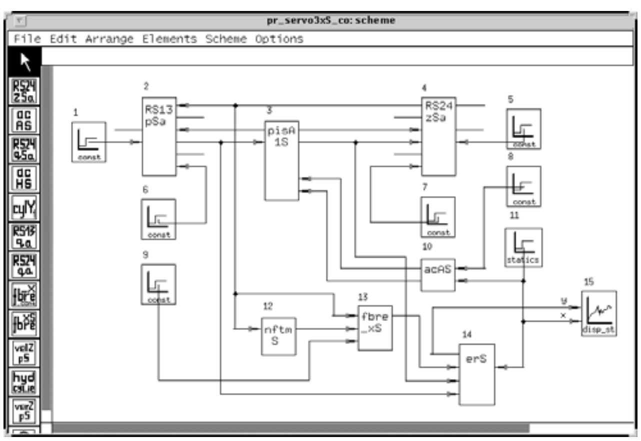

Fig.7. Simulation problem description for computing of characteristics ***xac = F(Fac)***.

In the Fig.7, in addition to the described multi-pole model classes, we used the constant inputs **("const")** for pressure at the supply port ***ppa*** (1 and 5), pressure at the return port ***pta*** (6 and 7), control voltage ***Uin*** (9) and velocity of an actuator ***vac*** (8). Class **"statics"** (11) gives the values of the force at the actuator ***Fac*** as argument. Class **"display_static"** (15) organises the graphical output for displacement of actuator ***xac.***

The given inputs ***Fac*** (11) and ***vac*** (8) trough actuator **acAS** (10) determine the force and the velocity for piston **pisA1S** (3). The piston has the volumetric flow ***q2pf*** and the pressure ***p2che*** as outputs, these are used as inputs for class of servo-valve slot pairs **RS24zSa** (4). The class of servo-valve slot pairs **RS24zSa** (4) determines the servo-valve displacement ***ze***, which is used as input for the class of servo-valve slot pairs **RS13pSa** (2), for nozzle-and-flapper valve with torque motor class **nftmS** (12) and for feedback and regulator class **fbre_xS** (13). The class **nftmS** (12) gives out the voltage ***Utme*** to the torque motor and the class **fbre_xS** (13) gives out the actuator position ***xacf*** without influence of servo-system elasticities. Class **erS** (14) calculates the elastic deformations of the servo-system and gives the **display_static** (15) the real static displacement ***xacS*** of the actuator.

The problem description for computing of characteristics ***xac = F(vac)***, by ***Uin = const, Fac = const*** is shown in Fig.8.

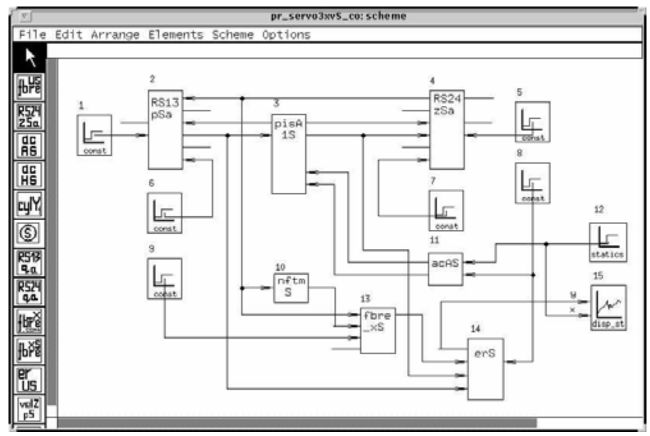

Fig.8. Simulation problem description for computing of characteristics ***xac = F(vac)***.

The constant value for force *Fac* is given by class **"const"** (9). Class **"statics"** (12) gives the values of the velocity of the actuator ***vac*** as argument.

5. EXAMPLES OF COMPUTING

The main parameters and input values for computing of the static characteristics have been chosen as follows (all the dimensions are in the SI system).

For the fluid: kinematic viscosity at temperature 40^0C *nu0* = 3.1e–04, density at temperature 15^0C *rho15* = 883 and temperature *theta* = 40^0C. The parameters of the fluid will be determined as depending on the arithmetical mean value of the pressures at the ends of each element.

For servo-valve: diameter of valve *d* = 8e-03, value of radial slot *s* = 3.5e-06, coefficient of slot length *k*=2/3, discharge coefficient of opened valve slots *mu* = 0.7, radius of valve edges *r* = 5e-06, initial values of slot displacement in initial position *z1...z4* = - 3e-05 (overlapped slots).

For piston: diameter of a piston *dpi* = 0.10, diameter of a left piston rod *dr1* = 0.04, diameter of a right piston rod *dr2* = 0.04, diameter of a orifice in the piston *d5* = 3e-04, constant value of a piston sealing friction force *Ffpi0* = 100, constant value of a right piston rod sealing friction force *Ffr2* = 50, discharge coefficient for orifice in the piston *mu* = 0.7.

For actuator: border values of an actuator friction force for imitation of mixed friction - see Fig.10.

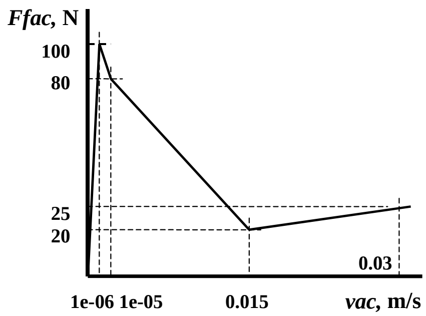

Fig.10. Friction force of an actuator ***Ffac*** as function of the actuator velocity ***vac***.

For nozzle-and-flapper valve and ***for torque motor:*** coefficient of proportionality of the servo-valve displacement to torque motor voltage *k* = 4e-05.

For feedback and regulator: coefficient of proportionality of voltage of feedback to actuator displacement *kpx* = 50. Basic amplifying coefficient of the regulator *ka0* = 20.

For elasticities: elasticity of a cylinder bush *ebu* = 1e-10, of a cylinder fixing *efi* = 1e-10, of a cylinder flange *efl* = 1e-11, of a cylinder lid *eli* = 1e-11, of a right piston rod *er2* = 1e-10.

Constant input values: pressure at the supply port ***ppa*** = 2.1e+07, pressure at the return port ***pta*** = 1.5e+06, input control voltage ***Uina*** = 2.5.

Computing parameters: allowed absolute iteration error for pressures *epsapi* = 6e+03, allowed relative iteration error *epsri* = 7e-03, iteration adjusting coefficient *adjust* = 0.3.

The computed characteristic ***xac = F(Fac)***, by ***vac*** = 0, using the basic constant amplifying coefficient of the regulator ***ka0*** = 20, is shown in Fig.11.

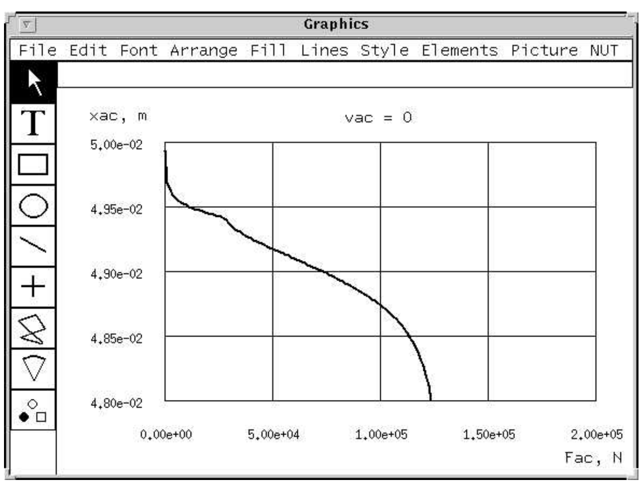

Fig.11. Computed characteristic ***xac = F(Fac)***.

The characteristic in the area of small loads is strongly non-linear due to overlapped servo-valve slots. Here, modification of this characteristic is proposed.

For this reason, various values of amplifying coefficient ***ka*** of the regulator as a function of servo-valve position ***z*** are chosen, as shown in Fig.12.

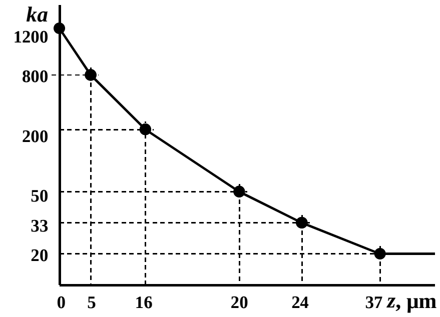

Fig.12. Amplification coefficient ***ka*** of regulator as function of the servo-valve position ***z***.

The computed characteristic ***xac = F(Fac)***, by ***vac = 0*** is illustrated in Fig.13.

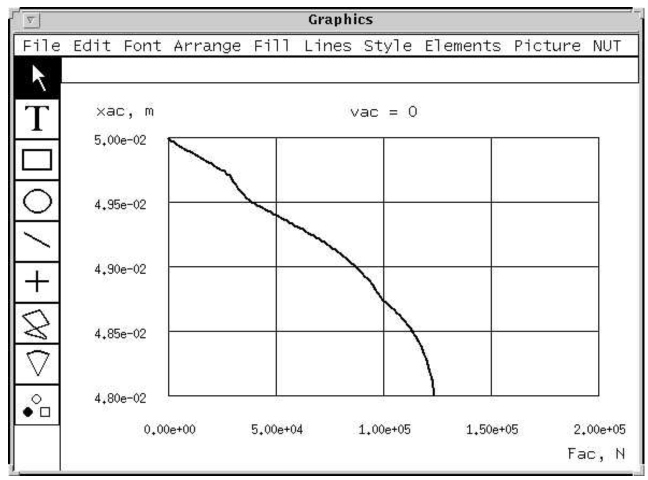

Fig.13. Computed modified characteristic ***xac = F(Fac)***.

CONCLUSIONS

The multi-pole modelling of an electro-hydraulic servo-system enables to construct very detailed and adequate object-oriented mathematical models for simulating of statics and steady state conditions.

Using of the programming and computing system NUT enables one to compose graphical simulation problem descriptions, automatically generate computing algorithms, and perform simulations. In this paper, static characteristic modification is proposed.

ACKNOWLEDGEMENT

The research was supported by Estonian Scientific Foundation (Grant no 5867).

REFERENCES

Grossschmidt, G. & Harf, M. (2002). Multi-pole Modelling and Simulation of a Four-way Valve Controlled Fluid Power System in NUT Environment. 16th European Simulation Multiconference June 3-5, 2002, Fachhochschule Darmstadt, Germany.

Uustalu, T.; Kopra, U.; Kotkas, V.; Matskin, M. & Tyugu, E. (1994). The NUT Language Report. Department of Teleinformatics, Royal Institute of Technology (Stockholm).