8th International DAAAM Baltic Conference
"INDUSTRIAL ENGINEERING"
19-21 April 2012, Tallinn, Estonia

MULTI-POLE MODELING AND INTELLIGENT SIMULATION OF TECHNICAL CHAIN SYSTEMS (PART 1)

Gunnar Grossschmidt and Mait Harf

Abstract: *Composing of multi-pole models and simulation of dynamic responses of a technical chain system is considered in the paper.*
Part 1 of the paper discusses difficulties arising in using existing simulation tools. A methodology is proposed that seems to be free of most of these disadvantages. Modeling of electro-hydraulic servovalve is considered as an example of chain system. An intelligent simulation environment CoCoViLa supporting declarative programming in a high-level language and automatic program synthesis is used as a tool for modeling and simulation.
In Part 2 multi-pole mathematical models of functional elements are described. Computing transient responses of the servovalve are considered.
Key words: *multi-pole model, electro-hydraulic servovalve, intelligent programming environment, simulation.*

1. INTRODUCTION

Most of technical systems are chain systems. Chain systems are e.g. various machines with drives (electromechanical, hydraulic, pneumatic) and automatic control systems, vibroisolation and amortization systems etc.
The most wide spread general purpose simulation tool Matlab/Simulink [1] possesses variety of built-in simulation engines. The simulation process is flow-based i.e., all the connecting arcs are directed and all the ports are either inputs or outputs.
Bond graphs are used in simulation of chain systems as well. The key of bond graph modeling is the representation (by a bond) of power as the product of efforts and flows with elements acting between these variables and junction structures to put the system together [2]. Bond graphs are oriented to bond graph elements. It makes the models complex and not easy understandable. Bond graph elements are expressed as two-pole elements, feedbacks cannot not be described and taken into account correctly.
Modeling and simulation tools in existence such as SimHydraulics™, ITI SimulationX, DSHplus, Dymola, HOPSAN, VisSim, AmeSim, 20-Sim, DYNAST, MS1™, HYVOS 7.0 etc. [3, 4] are object-oriented (systems are described as functional or component schemes) using equations with fixed causality or equations in non-causal form for each object.
Using only two-pole models for mechanical and hydraulic systems is not correct, as components of such systems exert feedback actions. The obtained large equation systems usually need checking and correcting to guarantee solvability. It is very complicated to debug and solve large differential equation systems with hundred of variables. The special integration procedures must be used in case of stiff differential equations. Mostly the observed systems are subjects to simplification. Usually models simplified to 3th…5th order are used in simulations. Often the models are linear. When using such models dynamics of all components can't be taken into account adequately.
In the current paper an approach is proposed, which is based on using multi-pole models with different oriented

causalities and oriented graphs of functional elements [3].

A special technique is used that allows avoid solving large equation systems during simulations. Therefore, multi-pole models of large systems do not need considerable simplification.

An intelligent simulation environment CoCoViLa [5] supporting visual programming and automatic program synthesis is used as a tool for modeling and simulation. Designer do not need to deal with programming, he can use the models with prepared calculating codes. It is convenient to describe simulation tasks visually, using prepared images of multi-pole models with their input and output poles.

2. MULTI-POLE MODELS

In general a multi-pole model [3] represents mathematical relation between several input and output values (poles).

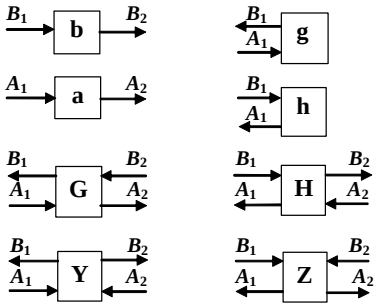

Fig. 1. Two- and four-pole models of technical system functional elements

The two-pole models (Fig.1) express the relations between flow variables $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ (form **b**), potential variables $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$ (form **a**), potential variable $\boldsymbol{A_1}$ and flow variable $\boldsymbol{B_1}$ (forms **g** and **h**). Dependences between variables in two-pole models of elementary functional elements (inertia, damping, resistance, elasticity) are expressed by one equation.

The four-pole models show the relations between pairs of potential and flow variables ($\boldsymbol{A_1}$, $\boldsymbol{B_1}$ and $\boldsymbol{A_2}$, $\boldsymbol{B_2}$). One of the variables in pair must be as input. Models of this form express the physical content of processes with feedback. Four forms of such four-pole models, or otherwise, four forms of mathematical causalities exist. They are denoted by letters **G**, **H**, **Y** and **Z**. Dependences between variables in four-pole models of elementary functional elements are expressed by two equations.

Further in the paper only multi-pole models are considered that express relation between at least two input and two output poles. Using such models enables to express both direct actions and feedbacks as it occurs in hydraulic and mechanical systems.

Each component of the system is represented as a multi-pole model having its own structure including inner variables, outer variables (poles) and relations between variables.

The oriented mathematical dependences between inner variables of components are convenient to express as oriented graphs.

Using multi-pole models allows describe models of required complexity for each component. For example, a component model can enclose nonlinear dependences, inner iterations, logic functions and own integration procedures. Multi-pole models of system components can be connected together using only poles. Using multi-pole models enables methodical, graphical representation of mathematical models of large and complicated systems. In this way we can be convinced of the correct composition of models and we don't need to check the solvability. It is possible directly simulate the statics or steady state conditions without using differential equation systems.

Implementing the multi-pole models for each component gives us possibility to use distributed calculations. The integration is performed in each model separately. Solving smaller equation systems is required instead of solving large equation systems. The multi-pole model of the whole system doesn't need substantial simplification. So we can perform simulations, taking the performance of all components into account adequately. In case of loop dependences between poles of component models the iteration method is used.

2. ELECTRO-HYDRAULIC SERVOVALVE

Electro-hydraulic servovalve has control function in electro-hydraulic servo-systems. The history of significant references in the area of electro-hydraulic servo-systems is given by Maskrey and Thayer [6] and Gordić et al. [7]. In servo-analysis and system synthesis it is often convenient to represent an electro-hydraulic servovalve by a simplified, equivalent transfer function [6-11]. Difficulty in assigning simplified, linear transfer functions to represent servovalve response is that these valves are highly complex devices that exhibit high-order, nonlinear responses. These approximations to servovalve response have resulted in such expressions as "the equivalent time constant of the servovalve is – seconds" or "the apparent natural frequency of the servovalve is – radians /second" [6]. The simplified block diagram is a third order system consisting of the armature/flapper mass, damping and stiffness, together with the flow-integration effect of the spool [6].

Servovalve dynamic response is described in terms of the logarithmical amplitude ratio and phase angle lag of the output in response to a sinusoidal input of varying frequency.

Functional scheme of the electro-hydraulic servovalve is shown in Fig. 2.

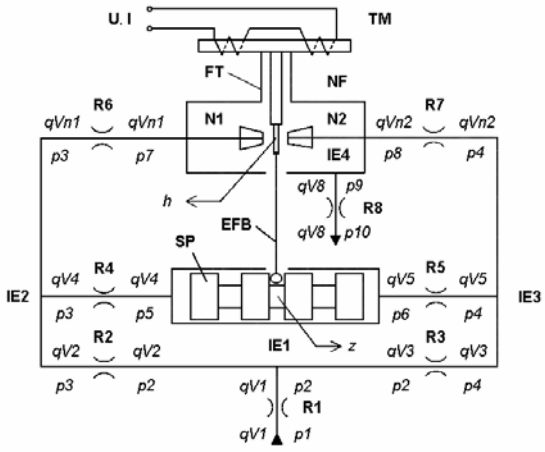

Fig. 2. Functional scheme of an electro-hydraulic servovalve

Electro-hydraulic servovalve consists of the following functional elements and subsystems: torque motor **TM** with flapper, flexure tube **FT**, nozzle-and-flapper valve **NF** with nozzles **N1** and **N2**, hydraulic resistors **R1**...**R8**, interface elements (tee couplings) **IE1**...**IE4**, sliding spool **SP** and elastic feedback **EFB** (as elastic conic rod) from spool to flapper. Structurally working slots of sliding spool belong to the servovalve. Functionally it is appropriate to consider working slots as separate subsystem when modeling and simulating a servo-system.

Input variables: input voltage ***U*** for the torque motor and the pressures ***p1*** and ***p10***.
Output variables: current to the **TM** ***I***, position of the flapper ***h***, position of the sliding spool ***z*** and volumetric flow rates ***qV1*** and ***qV8***.
Inner variables: pressures ***p2...p9,*** volumetric flow rates through nozzles ***qVn1*** and ***qVn2***, volumetric flow rates through resistors ***qV2...qV7***.
Scheme of mechanical parts of an electro-hydraulic servovalve is shown in Fig. 3.

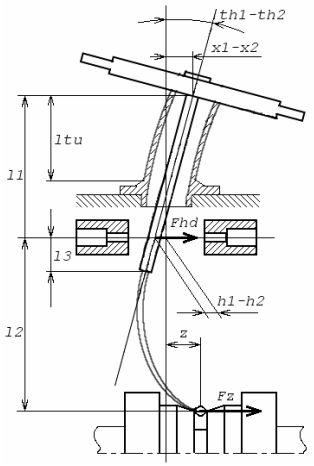

Fig. 3. Scheme of mechanical parts of an electro-hydraulic servovalve

The anchor of the torque motor is fixed on the flexure tube. The anchor turn angle ***th1*** transmits to the stiff rod, which gives flapper the moving ***h1*** between the nozzles. Difference of pressures at the ends of sliding spool causes the spool shift ***z***. Elastic feedback rod bends, the flapper moves in opposite direction on size ***h2*** and

the anchor turns in opposite direction on the angle ***th2***.

Flapper takes position ***h = h1 – h2*** and the anchor takes angle ***th = th1 – th2***. Anchor gets horizontal move ***x = x1 – x2***. The force acting to the sliding spool is ***Fz*** and the hydrodynamic force of fluid jets of nozzles is ***Fhd***. The geometrical distances ***l1, l2, l3, ltu*** are also shown.

3. MULTI-POLE MODEL OF AN ELECTRO-HYDRAULIC SERVOVALVE

The multi-pole model is decomposed into three components – torque motor with flapper **TM**, nozzle-and-flapper valve **NF** and spool in sleeve with elastic feedback to flapper **SP.** The model is presented in Fig.4

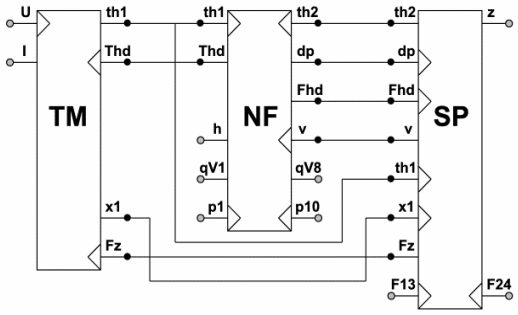

Fig. 4. Multi-pole model of an electro-hydraulic servovalve

The inputs of the multi-pole model of an electro-hydraulic servovalve are voltage ***U,*** pressures ***p1***, ***p10*** and hydrodynamic forces of fluid jets ***F13***, ***F24*** against sliding spool. The outputs are current ***I***, displacement of the flapper ***h***, displacement of the sliding spool ***z*** and volumetric flow rates ***qV1***, ***qV8***. Torque evoking through hydrodynamic force of the fluid jets ***Thd***, difference of pressures on the ends of sliding spool ***dp*** and velocity ***v*** of the sliding spool are used as well.

Representation of the model bases on the following assumptions: an ideal current source (infinite impedance) is used; deformations of the rod from anchor to flapper are negligible.

Multi-pole models of components **TM**, **NF,** and **SP** are described more detail in Part 2 of the paper.

4. SIMULATION ENVIRONMENT

CoCoViLa is a flexible Java-based simulation environment that includes both continuous-time and discrete event simulation engines and is intended for applications in a variety of domains [5]. The environment supports visual and model-based software development and uses structural synthesis of programs [12] for translating declarative specifications of simulation problems into executable code. The environment is developed as an open-source software, its extensions can be written in Java and included into simulation packages. CoCoViLa is implemented in the Institute of Cybernetics at the Tallinn University of Technology. The CoCoViLa environment is free and platform-independent.

CoCoViLa (Fig. 5) supports a language designer in the definition of visual languages, including the specification of graphical objects, syntax and semantics of the language. CoCoViLa provides the user with a visual programming environment, which is automatically generated from the visual language definition.

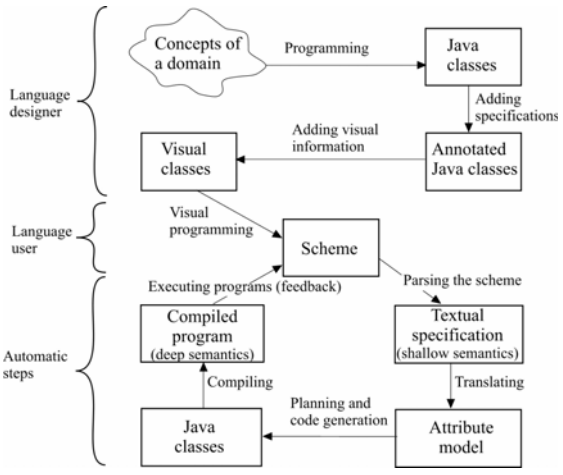

Fig. 5. Technology of visual programming in CoCoViLa

When a visual scheme is composed by the user, the following steps – parsing, planning and code generation – are fully automatic. The compiled program then provides a solution for the problem specified in the scheme, and the results it provides can be

feedback into the scheme, thus providing interactive properties.

Structural synthesis of programs is a technique for the automatic construction of programs from the knowledge available in specifications [12]. The method is based on proof search in intuitionistic propositional logic.

The synthesizer (planner) determines computational paths from initial variables to required goal variables (i.e., tries to solve a given computational problem "find values of V from given values of U", where U and V are sets of input and output variables). The planner's task is not only to construct a linear dataflow, but also to solve subtasks (higher-order dataflow) [13].

From a user's point of view the CoCoViLa framework consists of two components: Class Editor and Scheme Editor. The Class Editor is used for defining models of components of schemes as well as their visual and interactive aspects. The Scheme Editor is a tool for the language user. It is intended for developing schemes and for compiling (synthesizing) programs from the schemes according to the specified semantics of a particular domain. It provides an interface for visual programming, which enables one to compose a scheme from shapes of classes. The environment generated for a particular visual language allows the user to draw, edit and compile visual sentences (schemes) through language-specific menus and toolbars. The Scheme Editor is fully syntax directed in the sense that the correctness of the scheme is forced during editing. Drawing syntactically incorrect schemes is impossible.

When the visual classes have been built by software developers who must understand the problem domain as well, the language user need not be a software expert, but can work on the level of visual programming, arranging and connecting objects to create a scheme. Manipulating the scheme – a visual representation of a problem, is the central part of the user's activities.

5. COMPUTING PROCESS ORGANIZATION

Using visual specifications of described multi-pole models of technical chain system components one can graphically compose models of various chain systems for simulating statics, steady state conditions and dynamic responses.

When simulating statics or steady state conditions chain system behavior is simulated depending on different values of input variables. Initial and final values of input variables as well as number of calculation points are to be specified.

When simulating dynamic behavior, transient responses of the chain system caused by applied disturbances are calculated. Disturbances are considered as changes of input variables of the system (displacements, velocities, pressures, volumetric flows, load forces, load moments, control signals, etc.). Time step length and number of steps are to be specified. For integrations in dynamic calculations the fourth-order classical Runge-Kutta method is used in component models.

Computing processes are organized by corresponding process classes. To follow the system behavior in time, the concept of state is invoked. State variables are introduced for each component to characterize the element behavior at the current simulation step.

The simulation process starts from the initial state and includes calculation of following state (*nextstate*) from previous states (usually from *oldstate* and *state*). Final state (*finalstate*) is computed as a result of simulation.

A special method is used for calculating variables in loop dependences that cannot be calculated in straightforward way.

Such variables are split, initial approximate values are assigned and the variables are iteratively recomputed. Recomputing algorithms are constructed by the CoCoViLa program synthesizer as subtask solving algorithms. Feasibility of splitting

the variable, approximate initial value and request to find recomputing algorithm must be described in the multi-pole model of the technical chain system component.

SUMMARY

Difficulties arising in using existing modeling and simulation tools have been discussed.
Principles of multi-pole modeling have been described for technical chain systems. Modeling of electro-hydraulic servovalve is considered as an example of chain system. An intelligent simulation environment CoCoViLa supporting declarative programming in a high-level language and automatic program synthesis is used as a tool for modeling and simulation.
A special technique has been proposed that allows avoid solving large equation systems during simulations. Therefore, multi-pole models of large systems do not need considerable simplification.
In Part 2 multi-pole mathematical models of functional elements are described. Computing transient responses of the servovalve are considered.

REFERENCES

[1] Dabney, J. B. and Harman, T. L., 2001, Maste-ring SIMULINK, Prentice Hall.
[2] Fishwick, P., 2007, Handbook of Dynamic System Modelling, Chapter 26, Peter Breedveld, Port-Based Modeling of Engineering Systems in Terms of Bond Graphs. Taylor & Francis Group, LLC.
[3] Grossschmidt, G. and Harf, M., 2009, "COCO-SIM - Object-oriented Multi-pole Modeling and Simulation Environment for Fluid Power Systems, Part 1: Fundamentals", *International Journal of Fluid Power,* 10(2), pp. 91 - 100.
[4] HYVOS 7.0., 2010, Simulation software for valve-controlled cylinder drives, Bosch Rexroth, DCA_EN_0000019/en GB/15.01.2010.
[5] Kotkas, V., Ojamaa, A., Grigorenko, P., Maigre, R., Harf, M. and Tyugu, E., 2011, "CoCoViLa as a multifunctional simulation platform", In: *SIMUTOOLS 2011 - 4th International ICST Conference on Simulation Tools and Techniques,* March 21–25, Barcelona, Spain: Brussels, ICST, [1-8].
[6] Maskrey, R.H. and Thayer, W.J., 1978, A Brief History of Electrohydraulic Servomechanisms, Moog Technical Bulletin 141.
[7] Gordić, D., Babić, M. and Jovićić, N., 2004, "Modelling of spool position feedback servovalves", *International Journal of Fluid Power,* Vol. 5, No. 1 pp. 37-50.
[8] Thayer, W.J., 1965, Transfer Functions for Moog Servosystems, Moog Technical Bulletins: 103. Moog Inc.
[9] Noah, D.M., 2005, Hydraulic Control Systems, John Wiley and Sons, New York, US.
[10] Johnson, J.L, 2008, Designer`s Handbook for Electrohydraulic Servo and Proportional Systems, 4th edition, IDAS Engineering Inc.
[11] Moog, Servo and Propotional Systems Catalog.
[12] Matskin, M. and Tyugu, E., 2001, Strategies of structural synthesis of programs and its extensions, *Computing and Informatics*, Vol. 20, pp. 1-25.
[13] Tyugu, E., 1991, Higher-Order Dataflow Schemas, *Theoretical Computer Science*, v.90, pp. 185-198.

ADDITIONAL DATA ABOUT AUTHORS

Gunnar Grossschmidt
Assoc. Prof. emer.
Tallinn University of Technology,
Institute of Machinery
Ehitajate tee 5, 19086 Tallinn, Estonia
gunnar.grossschmidt@ttu.ee

Mait Harf
Senior researcher
Tallinn University of Technology,
Institute of Cybernetics,
Akadeemia tee 21, 12618 Tallinn, Estonia
mait@cs.ioc.ee

Corresponding Author:
Gunnar Grossschmidt