7th International DAAAM Baltic Conference
"INDUSTRIAL ENGINEERING"
22-24 April 2010, Tallinn, Estonia

SIMULATION OF HYDRAULIC CIRCUITS IN AN INTELLIGENT PROGRAMMING ENVIRONMENT (PART 1)

Grossschmidt,G. & Harf, M.

Abstract: *A hydraulic circuit of a fluid power system can be considered consisting of hydraulic tubes, local hydraulic resistors, fluid volume elasticity's and flow branching elements. Multi-pole mathematical models with lumped parameters having various causalities are composed for the components. Part 1 of the paper discusses the principles of composing multi-pole mathematical models for hydraulic circuits. In Part 2 computing steady state conditions and transient responses of the hydraulic circuits in the visual programming environment CoCoViLa are considered. The CoCoViLa environment used for modelling and simulation is a visual programming tool, which supports declarative programming in a high-level language and automatic program synthesis.*

Key words: hydraulic circuit, multi-pole model, CoCoViLa programming environment.

1. INTRODUCTION

A hydraulic circuit is an environment for energy transfer and transmitting signals in fluid power systems. Fluid elasticity, inertia and pressure drop cause energy losses in the hydraulic circuit. Energy transmission and control accuracy grow less. Dynamic phenomena of hydraulic circuits have great influence on the dynamic behaviour of fluid power systems.

Researchers have dealt with problems of hydraulic transmission lines for a long time (Stecki et al. [1], Kajaste [2]). Several research papers are concerned with the investigation of distributed parameters of fluid transmission lines, numerical and analytical approximations of the full solutions and various lumped parameter models and their solutions.

Computer simulation of frequency characteristics of hydraulic tubes with distributed parameters for various four-pole models has been discussed by Grossschmidt [3]. Approximations of the full solutions of tubes with lumped parameters for simulation of frequency characteristics have been proposed by Grossschmidt [4].

Fundamentals of model construction principles and methods have been described in papers by Grossschmidt and Harf [5, 6]. Multi-pole models with lumped parameters and different causalities are used for hydraulic circuit elements. Linear and square flow resistances, content of indissoluble air in fluid, dependence of fluid properties on pressure and temperature have been taken into account.

2. MATHEMATICAL MODELS OF HYDRAULIC CIRCUIT ELEMENTS

2.1 Mathematical models of hydraulic fluid properties

For hydraulic fluids used, density ρ_{15} at θ = 15 °C, cinematic viscosity ν at different temperatures θ in the interval from – 60 °C to +60 °C and the compressibility factor β_{F20} at θ = 20 °C must be specified. Computing physical properties of the fluid (density ρ, viscosity ν and compressibility factors β_A and β_F) depending on the temperature θ and the arithmetic mean pressure $p = (p1 + p2)/2$ at the ports of the hydraulic circuit element are described.

2.2 Multi-pole mathematical models of hydraulic tubes

Mathematical models with lumped parameters of the tube dynamics are represented as four-pole models of forms **G**, **H**, **Y** or **Z** having various causalities [3, 6].

They contain the lumped resistances ***R*** (linear resistance ***Rl*** and square resistance ***Rt***), inertias ***L*** and volume elasticities ***C***. Simultaneously, the hydraulic resistance model ***R*** takes into account the linear and square dependence of the pressure drop from volumetric flow. This enables one to use universal equations for laminar and turbulent flow of the fluid with no need to determine the Reynolds number. In addition, local hydraulic resistances that directly depend on the tube (bends, connections, elbows, inputs to a vessel, output from a vessel, etc.) are taken into account.

As result of analysis of various distributions of elements ***R***, ***L***, ***C*** in tube models the simplest structures of the four-pole models for tube dynamics, taking into account only the first natural frequency, are chosen as follows:

for form **G** in sequence ***L – R – C***;
for form **H** in sequence ***C – R – L***;
for form **Y** in sequence
L/2 – R/2 – C – R/2 – L/2;
for form **Z** in sequence
C/2 – R/2 – L – R/2 - C/2.

Hydraulic linear resistance at laminar flow ***Rl*** is expressed

$$Rl = \frac{2 * Al * l * \nu * \rho}{\pi * d^4}. \quad (1)$$

Hydraulic square resistance at turbulent flow ***Rt*** is expressed

$$Rt = \frac{8 * \rho}{\pi^2 * d^4} * \left(\varsigma + \lambda * \frac{l}{d} \right). \quad (2)$$

Inertial resistance of the flow ***L*** is expressed

$$L = \frac{4 * \rho * l}{\pi * d^2}. \quad (3)$$

Volume elasticity of the tube ***C*** is expressed

$$C = \frac{l * \pi * d^2}{4} * \left(\beta_F + \beta_A + \beta_T \right). \quad (4)$$

Compressibility factors β_F and β_A are expres-sed

$$\beta_F = \frac{1}{A_F * p_{FL} + B_F},$$

$$\beta_A = \frac{vol}{ka * \left(p + 10^5 \right)} * \left[\frac{10^5}{\left(p + 10^5 \right)} \right]^{ka}. \quad (5)$$

Expansibility factor of the tube wall β_T is ex-pressed

$$\beta_T = \frac{2}{K} \left[\frac{(d/2 + s)^2 + (d/2)^2}{(d/2 + s)^2 - (d/2)^2} + 0.3 \right]. \quad (6)$$

If it is necessary to use the tube model with lumped parameters of more detailed structure, we can represent it as consisting of several models of form **G** or **H** in sequence. In order to achieve correspondence of natural frequency of the tube with lumped parameters to the first natural frequency of the tube with distributed parameters, the values of ***L*** and ***C*** have been corrected by coefficients $kL = kC = (\pi/2) * (2/\pi)^{(1/n)}$. Here **n** is the number of joint similar models of form **G** or **H** in sequence. If the elements ***L/2*** or ***C/2*** are used (four-pole models of forms **Y** and **Z**) in the tube model, $kL = kC = (\pi/2) * (2/\pi)^{(2/n)}$. Coefficient ***kr*** is used for adjusting linear and square resistance values (which are calculated for steady state conditions) for dynamics.

The structure of the four-pole model of form **G** for the tube dynamics is represented in Fig. 1 and the corresponding signal flow graph in Fig. 2.

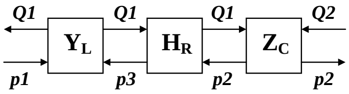

Fig. 1. The structure of the four-pole model of form **G** for the tube dynamics

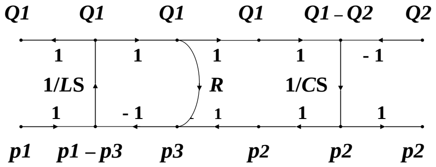

Fig. 2. Signal flow graph of the four-pole model of form **G** for the tube dynamics

For calculation the values $Q1$ and $p2$ the fourth-order classical Runge-Kutta method is used:

$$Q1next = Q1curr + (kq1 + 2*kq2 + 2*kq3 + kq4)/6,$$
$$p2next = p2curr + (kp1 + 2*kp2 + 2*kp3 + kp4/6. \quad (7)$$

To calculate Runge-Kutta coefficients kq and kp the differences of variables $dQ1$ and $dp2$ must be found:

$$dQ1 = \delta*(p1curr - p2curr -$$
$$kr*(Rl + Rt*abs(Q1curr)*Q1curr)/(kL*L),$$
$$dp2 = \delta*(Q1curr - Q2curr)/(kC*C). \quad (8)$$

The structure of the four-pole model of form **H** for the tube dynamics is represented in Fig. 3.

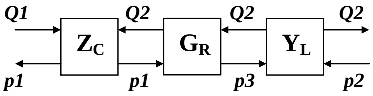

Fig. 3. The structure of the four-pole model of form **H** for the tube dynamics

The structure of the tube four-pole model of form **Y** for dynamics is represented in Fig. 4.

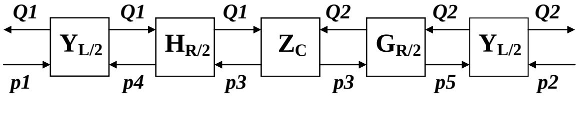

Fig. 4. The structure of the four-pole model form **Y** for the tube dynamics

The structure of the tube four-pole model of form **Z** for dynamic is represented in Fig. 5.

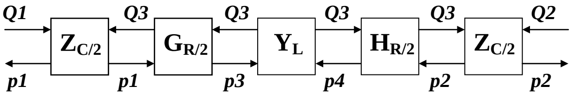

Fig. 5. The structure of the four-pole model of form **Z** for the tube dynamics

2.3 Multi-pole mathematical models of hydraulic local resistors

Mathematical models of hydraulic local resistors are represented as four-pole models of forms **G**, **H** or **Y**.

Hydraulic resistances can be described consisting of three components: linear resistance ***Rl***, square resistance ***Rt*** and inertial resistance ***L***. In present models inertial resistances ***L*** are not taken into account. Depending on the kind of hydraulic resistance, different equations are used for calculating the values of ***Rl*** and ***Rt***.

The following equations are used in models of different forms.

For model of form **G**:

$$p2 = p1 - (Rl + Rt*abs(Q2))*Q2,$$
$$Q1 = Q2; \quad (9)$$

for model of form **H**:

$$p1 = p2 + (Rl + Rt*abs(Q1))*Q1,$$
$$Q2 = Q1; \quad (10)$$

for model of form **Y** (the volumetric flow $Q1$ must be found from the square equation):

$$p1 - p2 - (Rl + Rt*abs(Q1))*Q1 = 0,$$
$$Q2 = Q1. \quad (11)$$

2.4 Multi-pole mathematical models of fluid volume elasticity

The mathematical model of fluid volume elasticity dynamics can be only represented as four-pole model of form **Z,** where the pressure $p1next$ is calculated using the Runge-Kutta method:

$$p1next = p1curr + (kp1 + 2*kp2 + 2*kp3 + kp4)/6, \quad (12)$$
$$p2next = p1next.$$

To calculate Runge-Kutta coefficients *kp* the difference of variable *dp*1 must be found:

$$dp1 = \delta * (Q1curr - Q2curr)/C. \quad (13)$$

Volume elasticity *C* is expressed:

$$C = V * (\beta_F + \beta_A). \quad (14)$$

2.5 Multi-pole model for hydraulic branching

Various hydraulic branchings (hydraulic interface elements) are used. The mathematical model of the hydraulic branching expresses the equation of volumetric flow continuity and the equal pressure condition.

The model determines the output volumetric flow, if the input volumetric flows are known. The pressures of all the ports are determined, if the pressure of one port is known.

3. PROGRAMMING ENVIRONMENT

CoCoViLa (Grigorenko et al. [7]) is a programming environment, which supports declarative programming in a high-level language, automatic program synthesis and visual programming. The CoCoViLa environment is Java based, platform-independent and free. CoCoViLa is developed in the Institute of Cybernetics at the Tallinn University of Technology.

Modelling and simulation of fluid power systems has been appropriate area to be considered in CoCoViLa environment. Ideas and experiences obtained as a result of investigations and computer experiments have been currently used in development of the CoCoViLa system.

The compiler-compiler of visual languages CoCoViLa (Fig. 6) 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.

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.

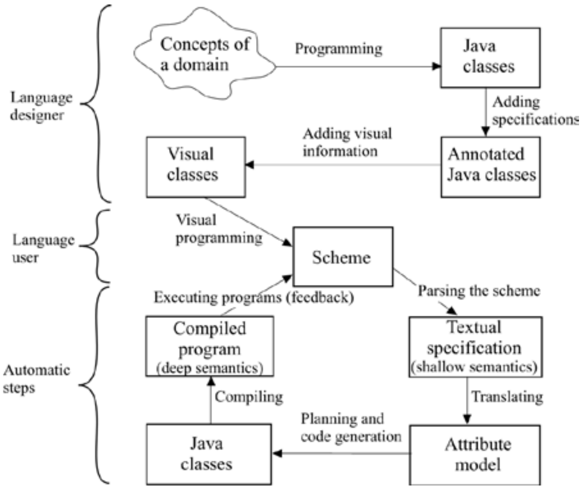

Fig. 6. Technology of visual programming in CoCoViLa

Automatic synthesis of programs is a technique for the automatic construction of programs from the knowledge available in specifications. Having a specification of a class, we are interested in solving following problems:

- Find an algorithm for computing the values of components y1,..., yn from the given values of components x1,..., xm.
- Find an algorithm for computing the values of all components that can be computed.

The automatic synthesis of programs is based on proof search in intuitionistic propositional logic.

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. The Scheme Editor 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.

Having developed the visual language we are able to load it in the Scheme Editor and build schemes by putting visual objects on the drawing canvas and connecting them through poles.

The Scheme Editor is fully syntax directed in the sense that the correctness of the scheme is forced during editing: drawing syntactically incorrect diagrams 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.

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.

4. CONCLUSIONS

Multi-pole models of hydraulic elements (tubes, various hydraulic resistors, volume elasticities, hydraulic interface elements) of hydraulic circuits, having various oriented causalities, have been proposed. The relatively simple non-linear four-pole models of the tube dynamics with lumped parameters have been proposed, which enable one to adjust the natural frequency and the damping, in very close agreement with the model with distributed parameters.

A brief overview of the CoCoViLa programming environment is presented. CoCoViLa environment is used as a tool for developing modelling and simulation system of hydraulic circuits, described in more detail in the paper “Simulation of Hydraulic Circuits in an Intelligent Programming Environment (Part 2)”.

5. ACKNOWLEDGEMENT

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

6. NOMENCLATURE

A	passage area
A_F, B_F	coefficients, depending on fluid and fluid temperature
Al	hydraulic friction coefficient of laminar flow
C	fluid volume elasticity
d	inner diameter of the tube or fluid jet
f	frequency
K	bulk modulus of the tube material
ka	polytrope exponent
kq, kp	Runge-Kutta coefficients
l	length of the tube or fluid jet
L	inertia resistance of the flow
p	pressure
$p1$, $p2$	pressures at the left and right port
$p1curr$, $p2curr$	pressures at the current time step
$p1next$, $p2next$	pressures at the next time step
Rl, Rt	hydraulic linear and square flow resistances
$Q1$, $Q2$	volumetric flows at the left and right port
$Q1curr$, $Q2curr$	volumetric flows at current time step
$Q1next$, $Q2next$	volumetric flows at the next time step
S	Laplace operator-variable

s	wall thickness of the tube
t	time
V	fluid volume,
vol	relative air volume in the fluid
β_A	compressibility factor of the air
β_F	compressibility factor of the fluid
β_T	expansibility factor of the tube wall
δ	time step
ζ	local resistance coefficient
θ	temperature
λ	hydraulic friction coefficient of turbulent flow
μ	flow coefficient
ν	fluid cinematic viscosity
ρ	fluid density
τ	inverse value of the time step

7. REFERENCES

1. Stecki, J. S and Garbacik A., 1995, “Viscosity Criteria for Selection of Long Transmission Model”, *Proc. of the 4th Scandinavian International Conference on Fluid Power, Sept. 26-29,* 1995, Tampere, Finland, Vol. 1 (2), pp. 384-398.

2. Kajaste, J., 1998, “Experiences in Simulating Pipeline Dynamics in Large-Scale Mechatronic Fluid Power Systems”, *Proc. of the 2nd Tampere International Conference on Machine Automation ICMA’98, Sept. 15-18*, 1998, Tampere University of Technology, Finland, pp. 789-803.

3. Grossschmidt, G., 1971, “Calculating Frequency Responses of Tubes with Distributed Parameters of Hydraulic Drives of Machine Tools”, *Proc. of Tallinn Polytechnical Institute Nr. 317,* 1971, pp. 147-156.

4. Grossschmidt, G. and Vanaveski J., 1971, “Calculating Frequency Responses of Tubes with Lumped Parameters of Hydraulic Drives of Machine Tools”, *Proc. of Tallinn Polytechnical Institute Nr. 317,* 1971, pp. 157-165.

5. Grossschmidt, G., 1997, “Principles of Modelling and Simulation of Technical Chain Systems”, *Proc. of the 9th European Simulation Symposium & Exhibition ESS’97, Simulation in Industry, Oct. 19-22,* 1997, SCS, Passau, Germany, pp. 705-712.

6. Grossschmidt, G. and Harf, M., 2009, “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.

7. Grigorenko, P., Saabas, A. and Tyugu, E., 2005, “COCOVILA – Compiler-Compiler for Visual Languages,” *Proc. of the 5th Workshop on Language Descriptions, Tools and Applications, (Edinburgh, 2005), v. 141, n. 4 of Electron.* Notes in Theor. Comput. Sci., pp. 137-142, Elsevier

8. ADDITIONAL DATA ABOUT AUTHORS

Gunnar Grossschmidt
Assoc. Prof. emer.
Senior researcher
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