

Backward simulation - A tool for designing more efficient mechatronic systems

Matthias Liermann

American University of Beirut, Department of Mechanical Engineering

www.aub.edu.lb

matthias.liermann@aub.edu.lb

Abstract

This paper proposes the use of backward simulation with Modelica as a tool to improve system design. The aim is to introduce system simulation into early design stages of mechatronic systems and to use the same software tools and model libraries that are also used in later stages for dynamic analysis and control design. It seems that the necessity of a control design is one of the main obstacles against the use of conventional dynamic system simulation in early design stages. The main benefit of backward simulation is that it does not require an implementation of feedback control.

The backward simulation approach is explained using the example of a servo-hydraulic drive. The paper shows that it can help to significantly reduce the energy consumption of this system. It is possible to simulate typical duty cycles of the drive without the need to redesign the control for each change.

Keywords: backward simulation; forward simulation; model inverse; hydraulics; mechatronics; servo-drive; efficiency optimization; servo drives; design process

1 Introduction

Dynamic system simulation is useful to analyse the dynamic behavior of systems, to design controllers or to determine cumulative system characteristics. Cumulative characteristics, such as for example energy consumption, depend on the definition of a typical load cycle. Simulation can be used to determine the state variables of components of the system for this load cycle and to predict the expected losses and total energy consumption. This is attractive for the designer in the initial stages of system development. A typical example is the design of a servo-drive. The designer has to make many choices in the initial design stages. Choices include the appropriate type of drives, such

as electrical, hydraulic or pneumatic, drive configuration. For each drive type the designer has to size its components. While dynamic simulation could be very helpful in making these choices, it is, however, rarely used. One of the main reasons is the need for control design, which often requires expert knowledge, [8].

Today, engineers use mainly steady state relationships to size components of a mechatronic system. This can be done with spreadsheet calculations. Some manufacturers move to offer specialized software for the dimensioning and analysis of a drive solution such as the SIZER configuration tool [1] for electric drives. Such tools take some dynamic forces into account, but only for predefined, typical scenarios.

This paper presents the method of backward simulation which allows the use of dynamic system simulation to study different system configurations and to size components. Backward simulation in the context of this paper is synonymous with (forward) simulation of the inverse system model. It means that input and output of the simulation are switched and that the direction of computation goes backward from the physical outputs to required control inputs. As will be explained in the paper, the main benefit of backward simulation is the fact that a control does not have to be implemented. Another benefit is, that implemented with Modelica, the backward simulation approach could be used with the same models and simulation tools used for the conventional (forward) simulation approach. This would lead to a better communication between design and control engineers and improve the product development.

The rest of the paper is organized as follows. The role of support of dynamic system simulation in product design and the benefits that are coming with the additional use of backward simulation are described in section 2. Section 3 reviews the use of backward simulation in literature and presents two simple examples to explain the idea and concept. Section 4

presents the mathematical model of a hydraulic drive as an exemplary application of the backward simulation approach. The backward simulation approach is demonstrated by using it to optimize the hydraulic drive efficiency for a certain load cycle in section 5. Section 6 provides the conclusion of this study.

2 Dynamic simulation support in product development

Figure 1 illustrates the conventional support of dynamic simulation in the design of a servo drive in the left flowchart. Usually the design starts by specifying the desired motion and the expected load. This provides the necessary information about required torque and speed which can be used to configure the system and size its components. The designer depends on analytic and empirical design formulas which he can solve for the unknown parameters. This approach makes sure that the hard requirements can be met. But it may be difficult to include some other important aspects of the design, such as average power consumption or required cooling power. These are cumulative characteristics which depend on the average use. For those aspects to be included, respective empirical or analytical design formulas are harder to define. Some sizing tools such as mentioned in [1] can actually compute power consumption for standard drive cycles on the basis of steady state simulation.

Dynamic system simulation is usually used only in later stages of product development. It is used to accelerate commissioning by setting up control hardware with hardware-in-the-loop simulation. It is also used in commissioning or to trouble-shoot unexpected system behavior. The design and implementation of a feedback control is a characteristic part of dynamic simulation, certainly of servo-drives which operate in closed loop. The control design verifies whether the requirement specifications can be met. Once the control is working, also the cumulative characteristics such as power consumption and required cooling can be assessed. If the investigations identify the need to make changes at this stage in product development, it is clear that the costs of making those changes will be high compared to changes made during conceptual design phase.

An alternative design process is shown in the right side of Figure 1. The initial design is found from the same knowledge and experience as in the conventional design process. The main difference is that backward dynamic simulation is used right from the beginning.

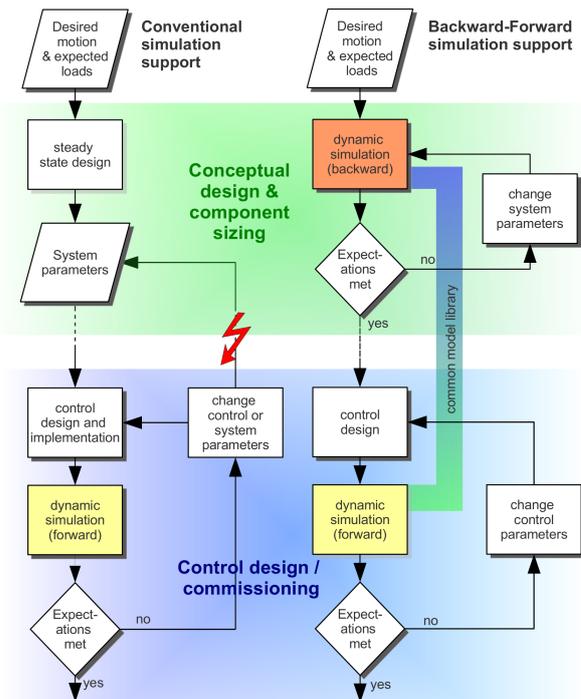


Figure 1: Backward/Forward vs. conventional simulation support of the design process

The (dynamic) simulation model is built from a component library, with the the same models used later for dynamic analysis and control design. However, no control is implemented and the simulation is run in backward mode with the required motion and external forces as boundary conditions. The backward simulation shows if any component runs into physical limitations. Also the energy efficiency over a representative duty cycle can be assessed. Different configurations can be tested to minimize the energy consumption. The backward simulation helps to detect and address dynamic performance issues. As a result it reduces the risk of costly design changes in late stages of the product development. The control design and dynamic analysis of the closed loop can be done at a later stage in the product development. The key advantage of backward simulation for the conceptual design is that a perfect control system is used, where the measured signal is always identical to the desired signal.

Another advantage is that the same dynamic model can be used in later stages of system development for the control design and hardware-in-the-loop simulation. The only difference between backward and forward simulation is in the definition of inputs and outputs and that backward simulation does not need a control to work. Whereas traditionally there is only small overlap of the fields of expertises of design and control engineers, combined backward/forward simu-

lation would enable them to use and update a common tool. This can improve cross-departmental communication and lead to faster and better product development. The use of the backward simulation approach is illustrated in this paper at the example of sizing of a hydraulic servo-axis.

3 Backward simulation

Backward simulation basically is forward simulation of the inverse model. It is to switch cause and effect of a system simulation. The model used for backward simulation is the same model used for forward simulation. The difference is in the definitions of inputs and outputs. The input to the model in forward simulation becomes the output in backward simulation. Forward simulation follows physical principles from cause to effect. Backward simulation can be used to compute the required input for a given output. Backward simulation, or simulation of the model inverse, can be well automated with equation based modeling languages such as Modelica. Dymola, as a simulation tool for Modelica models, is able to calculate the nonlinear model inverse. This capability can be used effectively for system configuration and sizing, but also for nonlinear control. The approach of this paper is closely related to the approach taken in [2], which uses the inverse simulation approach for the optimal selection of drive components in aircraft design. The use of inverse model simulation for nonlinear control schemes is presented in [14, 12].

This paper uses the term backward simulation synonymously for simulating the model inverse. The term backward simulation has been used also by other research groups. The program Advisor, a Matlab/Simulink implementation of model libraries used for optimization of hybrid vehicle drive trains, uses a combined backward-forward simulation approach, [17, 9]. The motivation of using the combined backward-forward approach is to be able to focus on system design. The optimization of drivetrain configurations can be approached without the need for control design. The problem of Advisor is that the way in which the models of this library can be used is predetermined, either backward or forward. Equation based modeling languages such as Modelica have the advantage that the causality of their use is not predefined.

The term backward simulation is also used in the context of backward planning (for example [5]) or for simulation of dynamic systems backward in time (see [6, 13, 16, 15]). In these cases, the simulation aims to

help find the system parameters and initial conditions which lead to a certain result.

3.1 Backward simulation in Modelica

In many components of the Modelica libraries, signal inputs are used, to apply external forces or other constraints or to make changes in component parameters. Signal inputs put restrictions on the connection structure, e.g. an "input" cannot be connected to an "input". It has to be connected to an output. However, it does not define the computational causality as in other approaches like Simulink. A typical example is a hydraulic control valve modeled as a turbulent resistance, see **Figure 2**. The volumetric flow rate Q is propor-

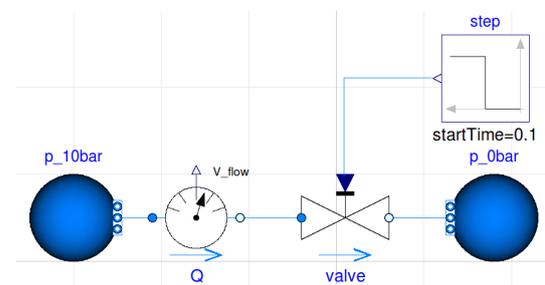


Figure 2: Valve from the Hydraulics library

tional to the partial opening x_V and to the square root of the pressure difference $p_1 - p_2$ across its ports [10]. With the flow gain c_V the flow through a control valve is expressed as

$$Q = c_V x_V \sqrt{p_1 - p_2} \quad (1)$$

Since the square root function is not defined for negative pressure differences and not differentiable for $\Delta p = 0$, often an approximate solution for the square root function is used to implement the flow-pressure relationship in a model [3]. Such a function is implemented as RegRoot in the standard Modelica Library in Modelica.Fluid.Utilities. It is strictly monotonically increasing, continuously differentiable and therefore invertible.

$$Q = c_V x_V \cdot \text{RegRoot}(p_1 - p_2, \Delta p_{\text{small}}) \quad (2)$$

For the implementation of control valves as for example in Modelica.Fluid.Valves.- ValveIncompressible, it is assumed that the partial opening x_V is not affected by the pressure difference or flow through a valve. The valve opening is therefore defined as a signal input.

For backward simulation, the partial valve opening x_V needs to be solved for from given flow Q and pressure difference $p_1 - p_2$.

$$x_V = \frac{Q}{c_V \sqrt{p_1 - p_2}} \quad (3)$$

With simulators such as Simulink, where the causality of a model is predefined, assigning pressure and flow as given from boundary conditions leads to an error since the valve opening is defined as a signal input. With Modelica this is possible, as **Figure 3** shows. With the block `Blocks.Math.Inverse-`

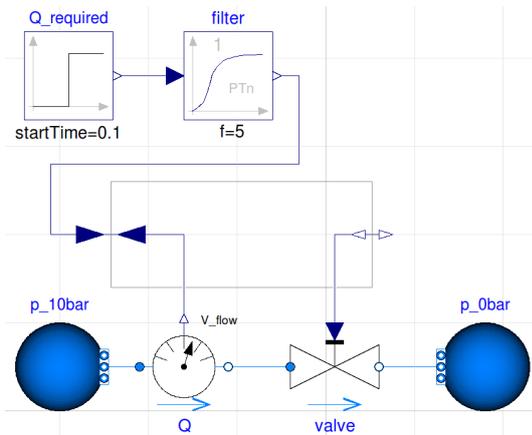


Figure 3: Valve from the Hydraulics library

BlockConstraints it is possible to connect an input function to the volumetric flow sensor and to impose a required flow on the computation while the signal input of the valve can be interpreted as a signal output. What this component does, is, to simply connect the two input signal connectors with each other as well as the two output signal connectors. The effect is that the model inverse is automatically derived by the Modelica translation engine. We see how Modelica allows to simulate the system 'backwards' simply by changing the boundary conditions for inputs and outputs.

3.2 Simple backward simulation example

The input step function in Figure 3 is filtered with a first order filter, without which the simulation would fail. As already stated, backward simulation is forward simulation of the model inverse. Inverting a dynamic model usually requires the derivatives of the input function. This is illustrated at the example of a simple linear system expressed by the transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^2 + 2s + 1}. \quad (4)$$

The input-output dynamics written in state differential form is

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (5)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (6)$$

The inverse of this system can be expressed as transfer function

$$G^{-1}(s) = \frac{U(s)}{Y(s)} = \frac{s^2 + 2s + 1}{1}. \quad (7)$$

However, there is no equivalent expression in state differential form. To express the inverse dynamics, the states would be functions of their derivatives. In Simulink and other assignment based simulation languages, it is important that each model element can be expressed in state differential form. For the inverse dynamics element this is impossible.

The state differential form is required also for the simulation of Modelica models. However, this is reached through automated rearrangement of all sub-model equations. This is a difference to other simulator concepts where each element or sub-model must be represented in state differential form initially.

Figure 4 illustrates how simulation of the inverse model dynamics is possible when the whole system is considered. To implement the simulation of the inverse model, the derivatives of the input to the inverse model must exist. Generating the input through a 2nd order filter assures that two derivatives exist.

$$G_f(s) = \frac{Y(s)}{R(s)} = \frac{25}{s^2 + 10s + 25}. \quad (8)$$

Applying the filtered signal $Y(s) = G_f(s)R(s)$ to the inverse model Eq. 7 yields

$$U(s) = R(s) \cdot \frac{25(s^2 + 2s + 1)}{s^2 + 10s + 25} \quad (9)$$

The combined system can be expressed in state differential form.

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -25 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 25 \end{bmatrix} r \quad (10)$$

$$u = \begin{bmatrix} -24 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 25r \quad (11)$$

Figure 4 shows that the results of backward and forward simulation match. Feeding the forward simulation with the result u of the backward simulations yields the desired system response y that was given as input to the backward simulation.

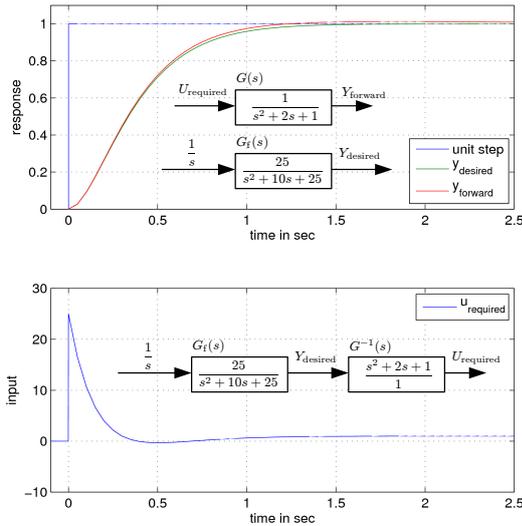


Figure 4: Backward simulation example

3.3 Limitations of backward simulation

The backward simulation approach is limited to systems for which the model inverse exists and is stable. Coulomb friction for example depends on the sign of velocity. For zero velocity the coulomb friction is undefined and depends on the history of motion. This function is not invertible without adjustments. Another challenge are physical limitations implemented in the models. If, during backward simulation, one component reaches a physical limitation, the states which cause the behavior of the model in limitation are not clearly defined anymore. There are infinitely possible combinations of states which cause the limited model to be in its limit. The cases for which the model inverse cannot be obtained are further elaborated in [14, 12]. It is subject of future research to show how relevant these issues are for typical configuration and sizing problems and how they can be addressed appropriately. The next section explains the mathematical model of the example application for which the advantage of the backward simulation approach is demonstrated.

4 Model of example application

A typical model for a servo-hydraulic drive is presented as given in many text books such as [7, 10, 11]. The drive consists of a servo-valve which connects the two ports of a cylinder to a constant pressure supply and a tank, see **Figure 5**.

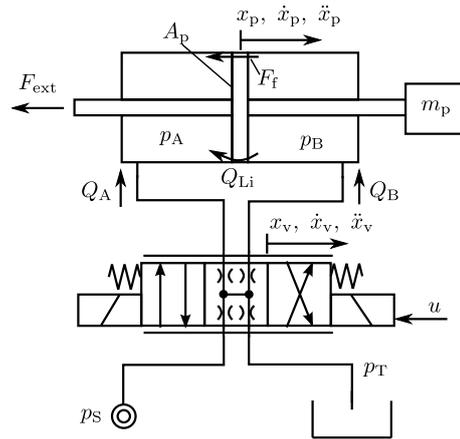


Figure 5: Hydraulic scheme

The model can be described by a system of nonlinear state differential equations of dimension 6.

$$\ddot{x}_p = \frac{1}{m_t(x_p)} [(p_A - \alpha p_B)A_p - F_f(\dot{x}_p) - F_{ext}] \quad (12)$$

$$\dot{p}_A = \frac{1}{C_{h,A}} [Q_A(p_A, x_v) - A_p \dot{x}_p + Q_{Li}(p_A, p_B)] \quad (13)$$

$$\dot{p}_B = \frac{1}{C_{h,B}} [Q_B(p_B, x_v) + \alpha A_p \dot{x}_p - Q_{Li}(p_A, p_B)] \quad (14)$$

$$\ddot{x}_v = -\omega_v^2 x_v - 2D_v \omega_v \dot{x}_v + \omega_v^2 u \quad (15)$$

Where the states and parameters are listed in **Table 1**.

The flow equations are nonlinearly dependent on the valve partial opening x_v and the pressure difference. It has to be defined for different cases depending on which ports are connected with each other.

$$Q_A = c_v \operatorname{sg}(x_v - x_0) \operatorname{sign}(p_S - p_A) \sqrt{|p_S - p_A|} \dots$$

$$\dots - c_v \operatorname{sg}(-x_v - x_0) \operatorname{sign}(p_A - p_T) \sqrt{|p_A - p_T|} \quad (16)$$

$$Q_B = c_v \operatorname{sg}(-x_v - x_0) \operatorname{sign}(p_S - p_B) \sqrt{|p_S - p_B|} \dots$$

$$\dots - c_v \operatorname{sg}(x_v - x_0) \operatorname{sign}(p_B - p_T) \sqrt{|p_B - p_T|} \quad (17)$$

As already explained in section 3.1, the term $\operatorname{sign}(\Delta p) \sqrt{|\Delta p|}$ does not work reliable in a Modelica simulator since there is an infinite derivative whenever Δp becomes zero. For practical implementation the approximate function `RegRoot` can be used. The different switching conditions are realized using the function `sg`, which is defined as:

$$\operatorname{sg}(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases} \quad (18)$$

Table 1: States and parameters of model

Symbol	Comment	Unit
C_h	capacity of chamber	$\frac{\text{m}^3}{\text{Pa}}$
$E'_{A,B}$	effective bulk modulus	Pa
F_{ext}	external force	N
F_f	friction force	N
$p_{A,B}$	pressure in A,A	Pa
$Q_{A,B}$	flow into chamber A,B	$\frac{\text{m}^3}{\text{s}}$
Q_{Li}	leakage from chamber B into A	$\frac{\text{m}^3}{\text{s}}$
u	$\frac{\text{valve signal}}{\text{maximum valve signal}}$	-
$V_{A,B}$	Volume chamber A,B	m^3
x_p	piston position	m
x_v	valve spool partial opening	-
$A_p = 7.6 \cdot 10^{-4}$	piston face side surface area	m^2
$C_{\text{Li}} = 1.6 \cdot 10^{-13}$	leakage coefficient	$\frac{\text{m}^3}{\text{Pa} \cdot \text{s}}$
$c_v = 8.9 \cdot 10^{-8}$	valve flow gain	-
$c_s = 0.01$	Stribeck velocity	$\frac{\text{m}}{\text{s}}$
$D_v = 0.9$	damping ratio of valve	-
$E_{\text{max}} = 1.7 \cdot 10^9$	bulk modulus at infinite pressure	Pa
$F_{c0} = 100$	Coulomb friction force	N
$F_{s0} = 100$	Static friction force	N
$m_t = 50$	total mass of piston	kg
$p_s = 200 \cdot 10^5$	supply pressure	Pa
$p_T = 2 \cdot 10^5$	reservoir pressure	Pa
$s = 0.8$	stroke	m
$x_o = -1\%$	fractional valve overlap	-
$\alpha = 1$	piston surface ratio	-
$\gamma = 800$	Approximation factor	-
$\omega_v = 628$	natural undamped frequency of valve	$\frac{\text{rad}}{\text{s}}$
$\sigma = 1000$	viscous friction coefficient	$\frac{\text{N} \cdot \text{s}}{\text{m}}$

According to the manufacturing of the valve, the spool can have over- or underlap with the sleeve in the middle position. The overlap parameter x_o takes this into account. If it is negative, it means that the valve is underlapped and therefore all valve ports are connected with each other in the middle position of the valve.

The leakage flow across the piston Q_{Li} is the cylinder pressure difference multiplied by the leakage coefficient C_{Li} .

$$Q_{\text{Li}} = C_{\text{Li}}(p_B - p_A) \quad (19)$$

The pressure gradient \dot{p} in a cylinder chamber is char-

acterized by the hydraulic capacity

$$C_h = \frac{V}{E'} \quad (20)$$

which is the quotient of Volume over effective bulk modulus of the respective chamber. The volumes change with position of the piston, while the bulk modulus varies with the chamber pressure. An empirical model proposed by [4] is

$$E' = E_{\text{max}} \left[1 - e^{-0.4 - 2 \cdot 10^{-7} p} \right] \quad (21)$$

where E_{max} is the bulk modulus at infinite pressure.

The friction of the hydraulic cylinder can be represented by the stribeck curve, which is a piecewise defined function.

$$F_f(\dot{x}_p) = \sigma \dot{x}_p + \text{sign}(\dot{x}_p) \left[F_{c0} + F_{s0} e^{-\frac{|\dot{x}_p|}{c_s}} \right] \quad (22)$$

where σ is the viscous friction coefficient, F_{c0} the coulomb friction, F_{s0} the static friction and c_s the so-called Stribeck velocity. Since the friction model as given by Eq. (22) is not invertible, an approximation of it is used by replacing the sign function.

$$\text{sign}(\dot{x}_p) \approx \frac{2}{\pi} \arctan(\gamma \dot{x}_p) \quad (23)$$

And therefore:

$$|\dot{x}_p| \approx \dot{x}_p \frac{2}{\pi} \arctan(\gamma \dot{x}_p) \quad (24)$$

With the values given in Table 1, the friction function is plotted in **Figure 6**.

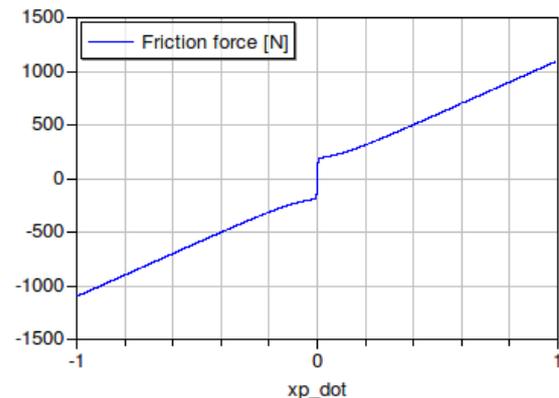


Figure 6: Friction model

The set of differential algebraic equations Eq. (12-15) is given in the standard form of ordinary differential equations, where the state derivatives are expressed as a function of the states and inputs. For

backward simulation the model inverse needs to be expressed by rearranging the equations. This is not possible algebraically in this case. However, Modelica tools, such as Dymola, can generate nonlinear inverse models automatically as explained in section 3. The next section explains how backward simulation can help in sizing a hydraulic system with respect to static and dynamic requirement specifications.

5 Efficiency study using dynamic backward simulation

Backward simulation allows to study the drive's performance for a whole duty cycle without the need to design a controller. In fact, perfect control is assumed because the expected output is forced on the system as a boundary condition. This is an advantage, since sometimes it is the necessity of control design which discourages the early use of system simulation. Often, in early stages of system development, issues such as architecture configuration and component sizing is important. System simulation in forward manner may then be impractical if changes in the system always require re-design of the controller. It is interesting to note that Modelica allows to use the same model for forward and backward simulation. This means that the same model used in backward manner for system configuration and component sizing may be used in forward manner later for the control design.

The idea of backward simulation is to force the prescribed duty cycle as boundary conditions on the physical outputs of the system, see **Figure 7**. Consequently, by simulating the inverse model, the corresponding physical inputs are calculated. To do this, no control has to be implemented. The advantage of this approach is demonstrated at the example of a hydraulic servo drive as modeled in the previous section. The model 'HSS' in **Figure 7** is described by Eqs. 12-15. For the sake of clarity the system was not put together by the commercial Modelon Hydraulics library. The components of this library include some effects which cause problems for the backward simulation approach. For example, an interpolation function is used to calculate the average density within a resistance. This interpolation function causes to fa

Figure 8 shows the required (filtered) duty cycle, the position and velocity trajectories and the external force impact. In this duty cycle the hydraulic drive moves out with a constant velocity of $0.23 \frac{\text{m}}{\text{s}}$ while applying a constant force on a workpiece of 10kN. The return stroke takes place with high velocity of $0.8 \frac{\text{m}}{\text{s}}$.

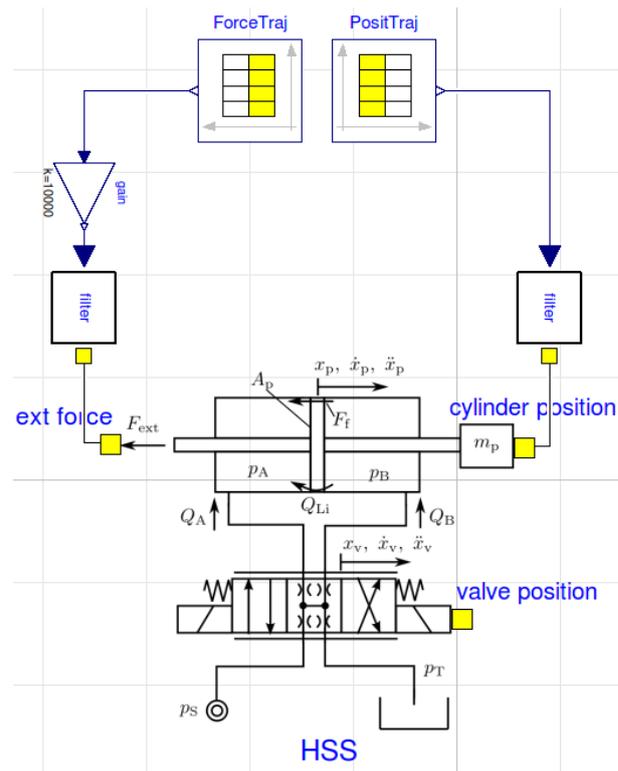


Figure 7: Dymola backward simulation of hydraulic servo system

The objective of optimization is to find the right sizes of cylinder and valve as well as choosing the operating pressure. To do this in conventional forward simulation, a control has to be designed. In case the requirement specifications cannot be met, it is unclear in forward simulation whether the suboptimal control limits the performance or whether the components just don't allow for a better performance.

Simulation of the setup is performed in backward simulation with the parameters as listed in Table 1. The design engineer can examine from the results whether component limitations were violated. It is also possible to examine the total energy consumption. **Figure 9** shows the required valve signal input HSS.xv, the cylinder pressures HSS.pA, HSS.pB and the cumulated amounts of energy E_{loss} , E_{mech} , and $E_{\text{mech+fric}}$. The top plot indicates that the valve size is too small since it opens beyond 100%. The cylinder pressures are within the range between supply and tank pressure. At the beginning of the force impact, the pressure in chamber A has a peak of 185 bar. The load pressure during the working stroke is $\Delta p_{A,B} = (167 - 34) \text{ bar}$ which is 67% of the available pressure difference. According to [11] this is the operating point of optimal efficiency for this type of servo drive. This can be seen well by looking at the bottom plot of

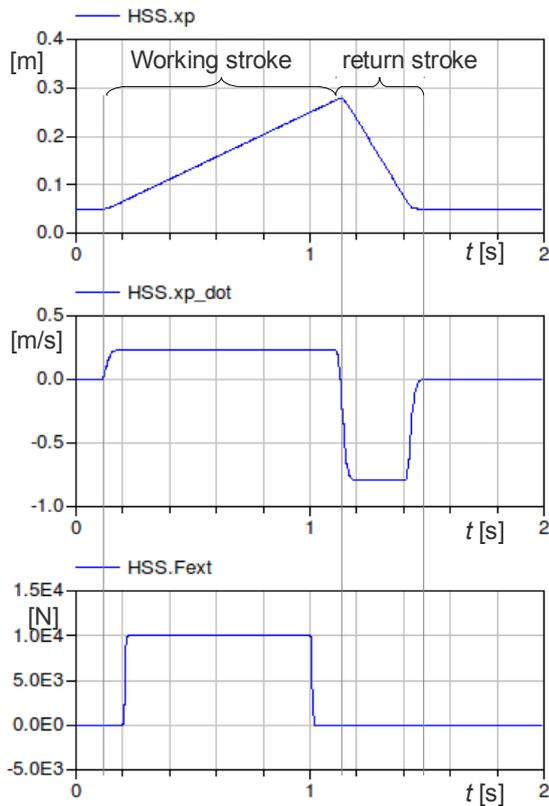


Figure 8: Duty cycle of drive

Figure 9. It compares the total hydraulic input energy E_{loss} , the mechanical output energy E_{mech} and the cumulative curve of mechanical output energy and friction energy $E_{\text{loss+fric}}$. In this example, the friction energy is negligible compared to losses in the valve. During the working stroke, the efficiency is approximately 67% which is optimal according to [11]. The total energy consumption for the working stroke is 3487J. The backward simulation reveals that the energy consumption for the return stroke is equal. This result is interesting although obvious. One might expect that the return stroke should consume less energy because no load is applied. However, the same flow is consumed at the same pressure level. Therefore the power is equal. The backward simulation can now be used to alter the design to achieve a higher efficiency while not violating the valve limitations at the same time.

The losses during the return stroke can be reduced by changing the area ratio of the cylinder. Choosing a faster valve reduces the dynamic peak in the valve and pressure signal at the moment when the load is suddenly applied. Reducing the cylinder area decreases the overall losses and increases the load pressure. The changes according to **Table 2** are found through few

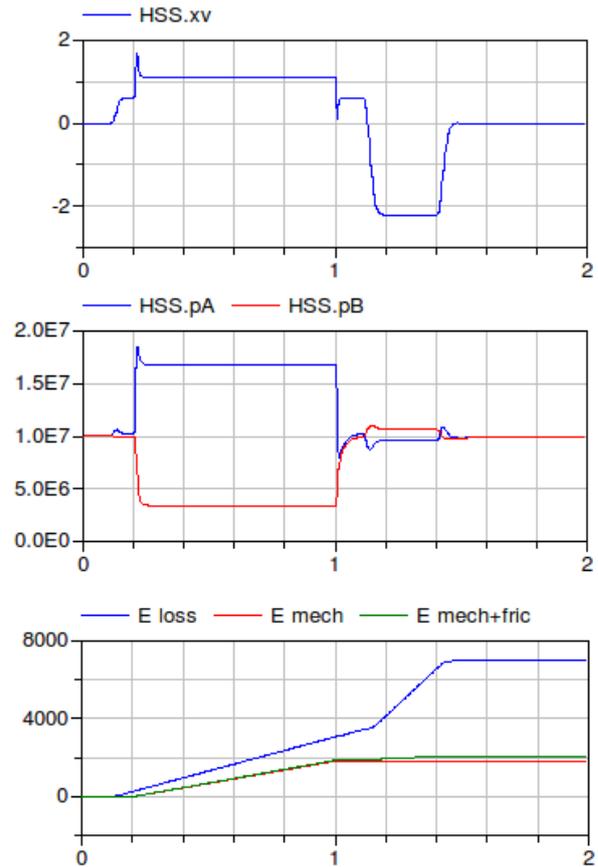


Figure 9: Valve signal, pressures and loss curves for duty cycle

iterative steps and do not represent an optimum solution. But the effect in terms of reduction of energy consumption is significant, as **Figure 10** shows.

The energy consumption of the improved system could be reduced by 38% from 7029J to 4323J for the example duty cycle. The valve was doubled in size and does not run into limitations anymore. This example demonstrates the advantage of the use of backward simulation for the design of servo drives. Through the use of backward simulation the energy efficiency of the systems could be analyzed for a representative duty cycle without the need to design a control. The control

Table 2: Modified parameters

Symbol	Comment	Unit
$A_p = 6.08 \cdot 10^{-4}$	piston face side surface area	m^2
$c_v 17.8 \cdot 10^{-8}$	valve flow gain	-
$\alpha = 0.5$	piston surface ratio	-
$\omega_v = 1256$	natural undamped frequency of valve	$\frac{\text{rad}}{\text{s}}$

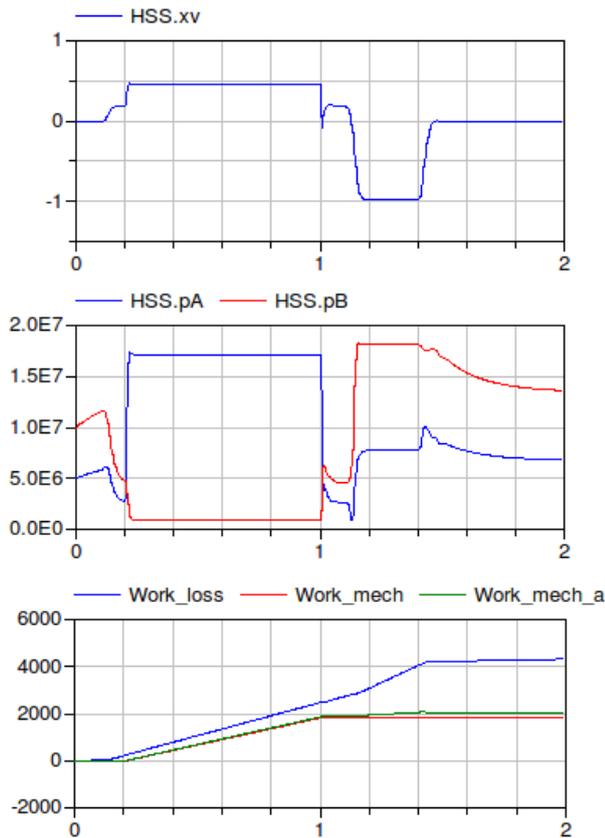


Figure 10: Valve signal, pressures and loss curves for optimized system

design is the next step after the dimensions of the drive have been determined.

6 Conclusion

This paper explains the idea of backward simulation, which is basically forward simulation of the inverse model. It was shown at the example of the mathematical model of a hydraulic servo-drive that building the model inverse by hand is not a trivial task. Depending on the system under study, the model inverse can only be determined numerically. Modelica tools such as Dymola provide this capability and therefore facilitate this new simulation technique. It is explained that the backward simulation approach only works if the system inverse can be build from the model. This may not be possible for systems with backlash or hysteresis. Phenomena like coulomb friction, which are discontinuous, need to be approximated.

The advantage of backward simulation is demonstrated in this paper at the example of a hydraulic servo drive for which a typical duty cycle was given. With little effort, new system parameters are found for

which the energy consumption is reduced by nearly 40%.

This study did not make use of already available libraries.

7 Acknowledgements

The author expresses thanks to the University Research Board of the American University of Beirut for funding this research.

References

- [1] Michael Ambros. Engineeringtool sizer. In *Antriebstag 2011, Siemens Kundenveranstaltung*, 2011.
- [2] Johann Bals, Gerhard Hofer, Andreas Pfeiffer, and Christian Schallert. Object-oriented inverse modelling of multi-domain aircraft equipment systems with modelica. In Peter Fritzson, editor, *Proceedings of the 3 International Modelica Conference, Linkping, November 3-4, 2003*, pages 377–384, 2003.
- [3] Francesco Casella, Martin Otter, Katrin Proelss, Christoph Richter, and Hubertus Tummescheit. The modelica fluid and media library for modeling of incompressible and compressible thermo-fluid pipe networks. In *Proceedings of the 5 International Modelica Conference, September 4-5th, 2006*, volume 2, pages 631–640, 2006.
- [4] W. Hoffmann. *Dynamisches Verhalten hydraulischer Systeme, automatischer Modellaufbau und digitale Simulation (Diss.)*. PhD thesis, RWTH Aachen University, 1981.
- [5] Chueng-Chiu Huang and Hsi-Kuang Wang. Backward simulation with multiple objectives control. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 (IMECS)*, volume 2, 2009.
- [6] Amor V.M. Ines and Peter Droogers. Inverse modelling in estimating soil hydraulic functions: a genetic algorithm approach. *Hydrology and Earth System Sciences*, 6(1):49–65, 2002.
- [7] Mohieddine Jelali. *Hydraulic Servo Systems: Modelling, identification and control*. Springer, 2003.

- [8] Matthias Liermann and Hubertus Murrenhoff. Knowledge based tools for the design of servo-hydraulic closed loop control. In *International Symposium on Power Transmission and Motion Control (PTMC)*, pages 17–28, Bath, England, 2005.
- [9] T. Markel, A. Brooker, T. Hendricks, V. Johnson, K. Kelly, B. Kramer, M. O’Keefe, S. Sprik, and K. Wipke. Advisor: a systems analysis tool for advanced vehicle modeling. *Journal of Power Sources*, 110(2):255 – 266, 2002.
- [10] Herbert E. Merrit. *Hydraulic control systems*. John Wiley & Sons, Inc., 1967.
- [11] Hubertus Murrenhoff. *Servohydraulik - geregelte hydraulische Antriebe [Servo-hydraulics - closed loop controlled hydraulic drives]*. Shaker, Aachen, 2008.
- [12] Martin Otter. *Modeling, simulation and control with Modelica 3.0 and Dymola 7.0 (Preliminary Draft, Jan 21, 2009)*. DLR, 2009.
- [13] A. Stohl, C. Forster, S. Eckhardt, N. Spichtinger, H. Huntrieser, J. Heland, H. Schlager, S. Wilhelm, F. Arnold, and O. Cooper. A backward modeling study of intercontinental pollution transport using aircraft measurements. *Journal of Geophysical Research*, 108(D12):Ach 8 1–18, 2003.
- [14] M. Thümmel, G. Looye, M. Kurze, M. Otter, and J. Bals. Nonlinear inverse models for control. In Gerhard Schmitz, editor, *Proceedings of the 4th International Modelica Conference, Hamburg, March 7-8, 2005*, pages 267–279, 2005.
- [15] Christopher D. Twigg and Doug L. James. Backward steps in rigid body simulation. *ACM Transactions on Graphics*, 27:25:1–10, 2008.
- [16] Eric A. Wan, Er A. Bogdanov, Richard Kiebertz, Antonio Baptista, Magnus Carlsson, Yinglong Zhang, and Mike Zulauf. Model predictive neural control for aggressive helicopter maneuvers. In *Software Enabled Control: Information Technologies for Dynamical Systems, chapter 10*, pages 175–200. IEEE Press, John Wiley & Sons, 2003.
- [17] K.B. Wipke, M.R. Cuddy, and S.D. Burch. Advisor 2.1: a user-friendly advanced powertrain simulation using a combined backward/forward approach. *Vehicular Technology, IEEE Transactions on*, 48(6):1751 –1761, November 1999.