PERGAMON

Annual Reviews in Control 27 (2003) 87–117

Annual Reviews in Control

www.elsevier.com/locate/arcontrol

IFAC Professional Brief

Modelling of physical systems for the design and control of mechatronic systems

Job van Amerongen*, Peter Breedveld

Control Engineering, Drebbel Institute for Mechatronics and Faculty of Electrical Engineering, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

Mechatronic design requires that a mechanical system and its control system be designed as an integrated system. This contribution covers the background and tools for modelling and simulation of physical systems and their controllers, with parameters that are directly related to the real-world system. The theory will be illustrated with examples of typical mechatronic systems such as servo systems and a mobile robot. Hands-on experience is realised by means of exercises with the 20-sim software package (a demo version is freely available on the Internet).

In mechatronics, where a controlled system has to be designed as a whole, it is advantageous that model structure and parameters are directly related to physical components. In addition, it is desired that (sub-)models be reusable. Common block-diagram- or equation-based simulation packages hardly support these features. The energy-based approach towards modelling of physical systems allows the construction of reusable and easily extendible models. This contribution starts with an overview of mechatronic design problems and the various ways to solve such problems. A few examples will be discussed that show the use of such a tool in various stages of the design. The examples include a typical mechatronic system with a flexible transmission and a mobile robot. The energy-based approach towards modelling is treated in some detail. This will give the reader sufficient insight in order to exercise it with the aid of modelling and simulation software (20-sim). Such a tool allows high level input of models in the form of iconic diagrams, equations, block diagrams or bond graphs and supports efficient symbolic and numerical analysis as well as simulation and visualisation. Components in various physical domains (e.g. mechanical or electrical) can easily be selected from a library and combined into a process that can be controlled by block-diagram-based (digital) controllers.

This contribution is based on object-oriented modelling: each object is determined by constitutive relations at the one hand and its interface, the power and signal ports to and from the outside world, at the other hand. Other realizations of an object may contain different or more detailed descriptions, but as long as the interface (number and type of ports) is identical, they can be exchanged in a straightforward manner. This allows top–down modelling as well as bottom–up modelling. Straightforward interconnection of (empty) submodels supports the actual decision process of modelling, not just model input and output manipulation. Empty submodel types may be filled with specific descriptions with various degrees of complexity (models can be polymorphic) to support evolutionary and iterative modelling and design approaches. Additionally, submodels may be constructed from other submodels in hierarchical structures.

An introduction to the design of controllers based on these models is also given. Modelling and controller design as well as the use of 20-sim may be exercised in hands-on experience assignments, available at the Internet (http://www.ce.utwente.nl/IFACBrief/). A demonstration copy of 20-sim that allows the reader to use the ideas presented in this contribution may be downloaded from the Internet (http://www.20sim.com).

Keywords: Mechatronics; Physical systems; Design and control; Modelling and simulation; Bond graphs; Energy-based modelling

---

1. Introduction

Mechatronic design deals with the integrated design of a mechanical system and its embedded control system. This definition implies that it is important, as far as possible, that the system be designed as a whole. This requires a systems approach to the design problem. Because in mechatronics the scope is limited to controlled mechanical systems, it will be possible to come up with more or less standard solutions. An important aspect of mechatronic systems is that the synergy realised by a clever combination of a mechanical system and its embedded control system leads to superior solutions and performances that could not be obtained by solutions in one domain. Because the embedded control system is often realised in software, the final system will be flexible with respect to the ability to be adjusted for varying tasks.

---

* Corresponding author. Tel.: +31-53-4895791; fax: +31-53-4892223.
E-mail addresses: J.vanAmerongen@utwente.nl (J. van Amerongen), P.C.Breedveld@utwente.nl (P. Breedveld).

1367-5788/$ – see front matter
doi:10.1016/S1367-5788(03)00010-5

The interdisciplinary field of mechatronics requires tools that enable the simultaneous design of the different parts of the system. The most important disciplines playing a role in mechatronics are mechanical engineering, electrical engineering and software engineering. One of the ideas behind mechatronics is that functionality can be achieved either by solutions in the (physical) mechanical domain, or by information processing in electronics or software. This implies that models for mechatronic systems should be closely related to the physical components in the system. It also requires software that supports such an approach. In an early stage of the design, simple models are required to make some major design decisions. In a later stage (parts of the), models can be more detailed to investigate certain phenomena more in depth. The relation to physical parameters like inertia, compliance and friction is important in all stages of the design. Because specialists from various disciplines are involved in mechatronic design, it would be advantageous if each specialist would be able to see the performance of the system in his or her own domain. It should be possible to see the performance of the mechatronic system in multiple views. Typical views that are important in this respect are:

- Physical models
- Bond graphs
- Control engineering models
  - Block diagrams
  - Bode plots
  - Nyquist plots
  - State space description
- Time domain
- Animation
- C-code of the controller

Often modelling, simulation and identification is done for systems that already exist. The design of a controller has to be done for an already realised and given 'process'. When we talk about design the system does not yet exist, there may be a lot unknown in the beginning, but there is also much more freedom to modify the system, not only the controller, but also the 'process', the mechanical construction itself.

In a typical design, the following phases can be distinguished. This is an iterative process and it may be necessary to go back to an earlier phase when problems arise in a later phase of the design process:

Phase 1: A concept is made of the system that has to be constructed and taking into account the tasks that have to be performed, the major components and their dominant dynamic behaviour are identified and modelled.

Phase 2: Controller concepts can be evaluated on this simple model. This requires that the model be available, e.g. as a transfer function, or a state space description.

Phase 3: When the evaluation is successful, the different components in the system can be selected and a more detailed model can be made. The controller designed in phase 2 can be evaluated with the more detailed model and controller and component selection can be changed.

Phase 4: When phase 3 has been successfully completed the mechanical system can be built and the controller can be realised electronically or in software. It is to be preferred that the translation from the controller tested in simulations is automatically transferred to, e.g. C-code, without manually coding; not only because of efficiency reasons, but especially to prevent coding errors.

This professional brief will address these issues. In the first part some representative examples of mechatronic design problems will be treated, showing how modelling and controller design are closely interacting during the design process of a mechatronic system. This part is written from the perspective of a control engineer. These cases will make clear that physical models are essential when besides the design of a controller also design of the mechanical part of the system is being considered. In the second part, port-based modelling of physical systems is introduced.

Physical system modelling provides *insight*, not only in the behaviour of systems that an engineer working on multidisciplinary problems wishes to design, build, troubleshoot or modify, but also in the behaviour of the environment of that system. A key aspect of the physical world around us is that 'nature does not know domains'. In other words, all boundaries between disciplines are man-made, but highly influence the way humans interact with their environment. A key point each modeller should be aware of is that any property of a model that is a result of the modeller's choice, should not affect the results of the model. Examples of modeller's choices are:

- Coordinates
- Metric
- Domain boundaries
- System boundaries
- Relevance of time and space scales

Several attempts to unified or systematic approaches of modelling have been launched in the past. In the upcoming era of the large-scale application of the steam-engine, the optimization of this multi-domain device (thermal, pneumatic, mechanical translation, mechanical rotation, etc.) created the need for the first attempt to a systems approach. This need for such a '*mechathermics*' approach was then named *thermodynamics*. Although many will not recognise the current treatment of thermodynamics as the first systems theory, it certainly was aimed originally in trying to describe the behaviour of such a system independently of the involved domains. However, it required a paradigm shift or 'scientific revolution' in the sense of Kuhn (1962) due to the fact that the concept of *entropy* had to be introduced for reasons of consistency, i.e. to be able to properly 'glue' these domains together with the concept of a conserved quantity called *energy*. The rather abstract nature of the concept of entropy has caused that students have considered thermodynamics a difficult subject ever since, resulting in only a relatively limited number of engineers and scientists actively using the thermodynamic approach in modelling of behaviour.

Despite the fact that the first evidence of the use of feedback dates back to 200–100 B. C. when water clocks required the water level in a reservoir to be kept constant, followed by Drebbel's thermostat and James Watt's fly-ball governor, it was only in the late nineteen twenties that feedback was realised by means of electric signals (Harold Stephen Black's 1927 famous patent that he wrote on a copy of the *New York Times*). At first, electronic feedback was used internally, to reduce distortion in electric amplifiers but later, especially during World War II, this concept was used in radar control and missile guidance. One might say that the multidomain approach to feedback was transferred to a signal approach in which the external power supply did not need to be part of the behavioural analysis. However, a more important paradigm shift was still to come, viz. the idea that the use of feedback allowed the construction of components, viz. operational amplifiers, with which basic mathematical operations could be mimicked, leading to analogue computers. This gave a new meaning to the terminology 'analogue simulation' that until then was conceived as mimicking behaviour by means of analogue circuits or mechanisms.

Just after World War II, due to the rapidly increasing demand for electric power, the USA was in great need for power, in particular hydropower, plants that should be able to deal with large and sometimes rapid fluctuations in the power grid. Obviously, the success of control theory (cybernetics) during World War II inspired many to apply control theory to the dynamic problems involved in electric power production. One such a civil engineer by the name of Henry Paynter (http://www.hankpaynter.com/) tried to use the early analogue computers that he invented together with James Philbrick, to simulate the dynamics of the power plants to be built (http://www.me.utexas.edu/~lotario/paynter/paynterbio.html). He used the common description of block diagrams that display the computational structure of the differential and algebraic equations being used, as these mathematical operations were to be mapped directly on the basic components of the analogue computer. However, for reasons that will become clear in the course of this contribution (viz. related to the concept of computational causality) he ran into formulation problems. At the beginning of the fifties, he realised himself that the concept of a 'port' introduced in electrical circuit theory by Wheeler and Dettinger (1949) a few years earlier, should be extended to arbitrary power ports that can be applied domain-independently. Power ports include mechanical ports, hydraulic ports, thermal ports, electric ports, etc., i.e. everything Paynter needed for the description of the dynamic behaviour of power plants (http://www.hankpaynter.com/Bondgraphs.html).

In the following decade, after moving to the MIT mechanical engineering department, he designed an efficient notation based on the efficient representation of the relation between two ports by just one line that he called a 'bond'. This so-called 'bond graph' notation was completed when he finally introduced the concept of the junction in 1959 (Paynter, 1961). Junctions not only make bond graph a powerful tool, but they are rather abstract concepts. Just like thermodynamics, bond graphs never became widely popular, although they spread over the whole world and are still alive after more than forty years. By contrast, signal processing, analogue and later digital computing, were not constrained to physical reality. This allowed people to mimic virtually everything, from physically correct or incorrect models to arbitrary mathematical relations that described imaginary systems. In the previous decennium, this even led to concepts like a 'cyber world', etc. even though the level of physical modelling in most virtual environments is rather low, as demonstrated by the unnatural features of much virtual behaviour.

Nevertheless, the introduction of rapid and flexible machinery for production, assembly, manipulation (including surgery), etc. that has truly taken off in the nineties, introducing again the need for a systems approach. In these application areas, physical constraints still limit imagination. The dynamics of such devices heavily leans on the application of digital electronics (microcomputers) and software. But a domain-independent description of the parts in which power plays a role is crucial to make a designer aware of the fact that a considerable part of these systems is constrained by the limits of the physical world. This mix of mechanics, or rather physical system engineering in general at the one hand and digital electronics, software and control at the other hand has been named 'mechatronics'.

Obviously, a smooth connection is needed between the information-theoretical descriptions of the behaviour of digital systems and physical systems theory. From its introduction, bond graphs have allowed the use of signal ports, both in- and output, and a corresponding mix with block diagrams. As all digital operations can be successfully represented by block diagrams similar to mathematical operations, the common bond graph/block diagram representation is applicable. This graphical view supports a hierarchical organization of a model, supporting reusability of its parts.

However, many systems that are studied by (mechatronic) engineers differ from the engineering systems that were previously studied in the sense that the spatial description of complex geometries often plays an important role in the dynamic behaviour, thus including the control of these systems. This shows the need for a consistent aggregation of at the one hand the description of the *configuration* of a mechanism and at the other hand the *displacements* in a system that in some way are related to the *storage* of potential or elastic energy.

Another aspect of these systems is that only few realistic models can be solved analytically, emphasizing the important role of a numerical solution (simulation). The aggregation of numerical properties in the representation of dynamic systems allows that a proper trade-off is made between numerical and conceptual complexity of a model. The approach discussed herein offers a basis for making a trade-off between numerical and conceptual complexity, resulting in both a higher modelling efficiency and numerical simulation efficiency.

Motivated by the problems encountered in the examples, the energy-based approach towards modelling is treated in some detail necessary for the description of simple mechatronic systems. This will give the reader sufficient insight in order to exercise the approach with the aid of freely available demo version of the modelling and simulation software 20-sim. For more advanced issues the interested reader is referred to the references. The modelling and simulation tool 20-sim allows high level input of models in the form of iconic diagrams, equations, block diagrams or bond graphs and supports efficient symbolic and numerical analysis as well as simulation and visualisation. It is based on an approach to formulate mechanical constraints primarily in terms of velocities, not displacements. Basic elements and generic domain-dependent components in various physical domains (in particular the mechanical and electrical domain) can easily be selected from a library and combined into a process that can be controlled by block-diagram-based (digital) controllers as demonstrated in the case studies of the next sections.

2. Context

'Mechatronic design deals with the integrated and optimal design of a mechanical system and its embedded control system'. This definition implies that the mechanical system is enhanced with electronic components in order to achieve a better performance, a more flexible system or just reduce the cost of the system. In many cases the electronics are present in the form of a computer-based embedded (control) system. This does not imply that every controlled mechanical system is a mechatronic system because in many cases the control is just an add-on to the mechanical system in a sequential design procedure. A real mechatronic approach requires that an optimal choice be made with respect to the realization of the design specifications in the different domains.

In control engineering, the design of an optimal control system is well understood and for linear systems standard methods exist. The optimization problem is formulated as follows: given a process to be controlled, and given a performance index (cost function), find optimal controller parameters such that the cost function is minimized (Fig. 1). With a state feedback controller and a quadratic cost function, solutions for the optimal controller gains can be found with standard controller design software, such as Matlab (Mathworks, 2001).

Mechatronic design on the contrary requires that not only the controller be optimized. It requires optimization of the system as a whole. In the ideal case all the components in the system: the process itself, the controller, as well as the sensors and actuators, should be optimized simultaneously (Fig. 2).

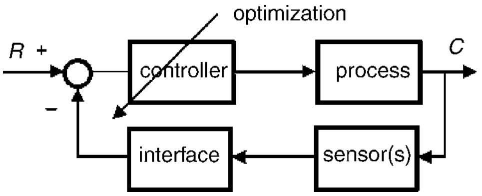


Fig. 1. Optimization of the controller.

In general, this is not feasible. The problem is ill-defined and has to be split into smaller problems that can be optimized separately. Later on the partial solutions have to be combined and the performance of the complete system has to be evaluated. After eventually readjusting some parts of the system this leads to a sub optimal solution.

In the initial conceptual design phase it has to be decided which problems should be solved mechanically and which problems electronically. In this stage decisions about the dominant mechanical properties have to be made, yielding a simple model that can be used for controller design. Also a rough idea about the necessary sensors, actuators and interfaces has to be available in this stage. When the different partial designs are worked out into some detail, information about these designs can be used for evaluation of the complete system and be exchanged for a more realistic and detailed design of the different parts.

A mechatronic system consists by definition of a mechanical part that has to perform certain motions and an electronic part (in many cases an embedded computer system) that adds intelligence to the system. In the mechanical part of the system power plays a major role. In the electronic part of the system information processing is the main issue. Sensors convert the mechanical motions into electrical signals where only the information content is important or even into pure information in the form of numbers (if necessary, through an AD-converter). Power amplifiers convert signals into modulated power. In most cases the power supply is electrical, but other sources such as hydraulic and pneumatic power supplies are possible as well. A controlled mechanical motion system thus typically consists of a mechanical construction, one or more actuators to generate the desired motions and a controller that steers the actuators based on feed-forward and sensor-based feedback control (Fig. 3).

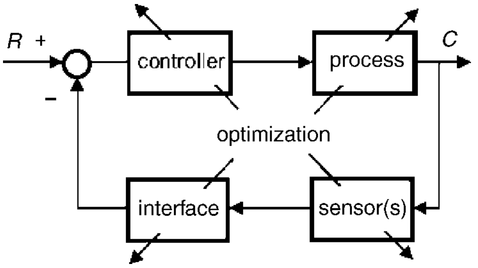


Fig. 2. Optimization of all system components simultaneously.

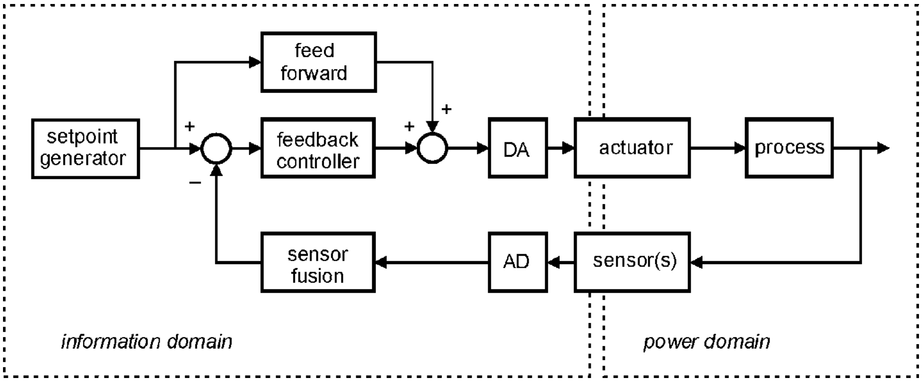


Fig. 3. Mechatronic system.

3. Integrated modelling, design and control of a servo system

3.1. Modelling

During the design of mechatronic systems, it is important that changes in the construction and the controller be evaluated simultaneously. Although a proper controller enables building a cheaper construction, a badly designed mechanical system will never be able to give a good performance by adding a sophisticated controller. Therefore, it is important that during an early stage of the design a proper choice can be made with respect to the mechanical properties needed to achieve a good performance of the controlled system. On the other hand, knowledge about the abilities of the controller to compensate for mechanic imperfections may enable that a cheaper mechanical construction be built. This requires that in an early stage of the design a simple model is available, that reveals the performance limiting factors of the system. Still there is a gap between modelling and simulation software used for evaluation of mechanical constructions and software used for controller design. Mechanical engineers are used to finite element packages to examine the dynamic properties of mechanical constructions. It is only after reduction to low-order models (modal analysis) that these models can be used for controller design. On the other hand, typical control engineering software does not directly support the mechatronic design process either; in the modelling process the commonly used transfer functions and state space descriptions often have lost the relation with the physical parameters of the mechanical construction. Tools are required that allow modelling of mechanical systems in a way that the dominant physical parameters (like mass and dominant stiffness) are preserved in the model and simultaneously provide an interface to the controller design and simulation tools control engineers are used to (Coelingh, 2000; Coelingh, de Vries, & van Amerongen, 2000).

Simulation is an important tool to evaluate the design of mechatronic systems. Most simulation programs like Simulink (Mathworks, 2001) use block diagram representations and do not support physical modelling in a way that direct tuning of the physical parameters of the mechanical construction and those of the controller is possible as required in the design of mechatronic systems. Recently programs that allow physical modelling in *various physical domains* became available. They use an object-oriented approach that allows hierarchical modelling and reuse of models. The order of computation is only fixed after combining the sub systems. Examples of these programs are 20-sim (Controllab Products; Broenink, 1990), and Dymola (Dynasim).

In this section, the modelling and simulation program 20-sim (pronounce: Twente Sim) will be used to illustrate the simultaneous design of construction and controller in mechatronic systems. 20-sim supports object-oriented modelling. Power and signal ports to and from the outside world determine each object (Weustink, de Vries, & Breedveld, 1998). Inside the object there can be other objects or, on the lowest level, equations. Various *realizations* of an object can contain different or more detailed descriptions as long as the interface (number and type of ports) is identical. Modelling can start by a simple interconnection of (empty) submodels. Later they can be filled with realistic descriptions with various degrees of complexity. de Vries (1994) refers to this as *polymorphic* modelling. Submodels can be constructed from other submodels in hierarchical structures. Proper physical modelling is achieved by coupling the submodels by means of the *flow of energy*, rather than by *signals* such as voltage, current, force and speed. At the lowest level all models are described by equations. Signal models can be described at a higher level by block diagram elements that are coupled to other elements by means of signal ports. Models of physical components can be described at a higher level by means of bond graphs and iconic diagrams. These submodels are coupled to other submodels by means of power ports. Actuator and sensor elements typically have a signal port and a power port and enable, e.g. the connection of signal-based controllers with power-based physical parts of the mechatronic system.

This way of modelling is well suited for mechatronic system design as will be illustrated with a few cases.

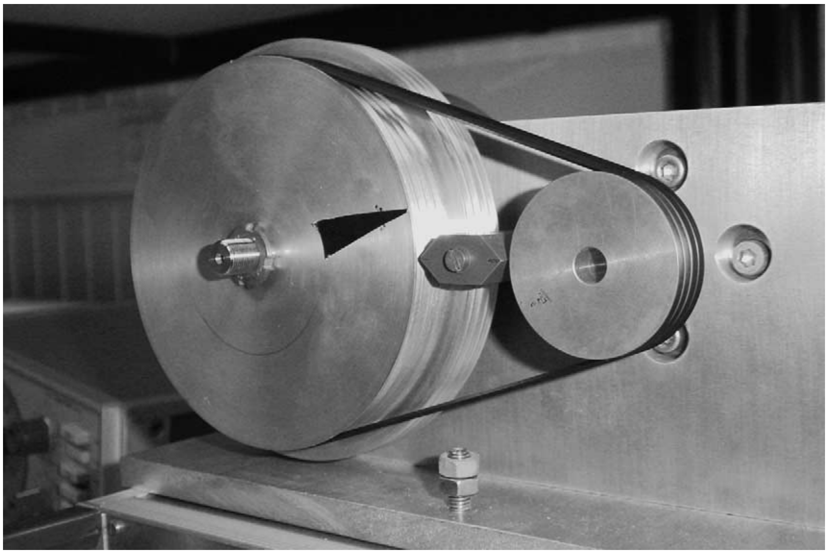


Fig. 4. Simple DC-servo system.

3.2. Modelling and design of a servo system

We want to consider the design of a simple servo system, consisting of a current or voltage source, a DC-motor and a mechanical load driven through a transmission (Fig. 4).

For the time being the transmission is disregarded. The belt is considered as infinitely stiff and the transformation ratio is taken care for by changing the motor constant. If a power amplifier driven by a signal generator describes the current or voltage source, we can draw the iconic diagram of Fig. 5 . (All the drawings in the example have been made with the modelling and simulation package 20-sim. The software and (most of) the examples can be downloaded from the Internet.) At this stage, the different *components* in this model are still empty. But all components have electrical and/or mechanical 'ports'. With the proper interfaces (ports) defined, the components can be connected to each other.

To derive equations, we have to fill in the details of the different components. The first question is whether the motor will be controlled by means of voltage or current. Let us assume for the moment that we will apply current steering. In that case the power amplifier delivers a current, proportional to the input signal of the amplifier. If the input of the power amplifier is assumed to be a constant, the power amplifier can be modelled as a current source. The motor converts the electrical current into a torque. We will initially assume that the dynamics of the motor (due to its electrical and mechanical parts) are negligible. The transmission is in its most simple form a transformation ratio. The load will have some inertia and friction. This leads to the more detailed model of Fig. 6, where the model is built up from a number of ideal basic elements. 'Ideal' means that each element represents only one single physical phenomenon, like inertia, friction, etcetera.

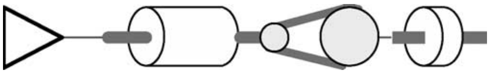


Fig. 5. Iconic diagram of the simple servo system.

From this diagram, the basic equations can be derived:

Current source: $i = I_0$
Motor: $T = K_m i$ and $u = K_m \omega$
Transmission: $T_{motor} = n T_{load}$ and $\omega_{load} = n \omega_{motor}$
Friction: $T_f = f\omega$
Inertia: $T_J = J\frac{d\omega}{dt}$

where I is the motor current; K_m is the motor constant; T is the torque delivered by the motor; n is the transmission ratio (for simplicity assumed to be 1, such that $T_{motor} = T_{load} = T$); T_f is the torque loss due to friction; T_J is the torque used to accelerate the load; J is the inertia of the load; f is the coefficient of viscous friction; ω_1 is the motor angular velocity; ω_2 is the load angular velocity; φ is the load angle.

With $T - T_f + T_J = 0$ it follows that:

$$\frac{K_m}{n} i = f\omega + J\frac{d\omega}{dt}$$

$$\omega = \frac{d\varphi}{dt}$$

After Laplace transformation this system can be transferred into the following block diagram of Fig. 7 with:

$$\frac{\omega}{u} = \frac{K_m}{n}\frac{1/f}{(J/f)s + 1} = \frac{K}{s\tau + 1}$$

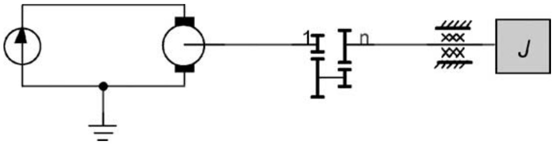


Fig. 6. Schematic diagram modelled with basic elements.

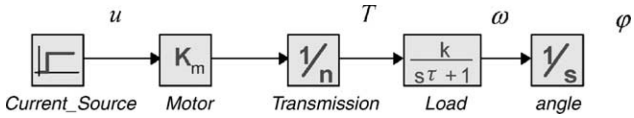


Fig. 7. Block diagram of a current controlled servo system.

or

$$\frac{\varphi}{u} = \frac{K_\mathrm{m}}{n}\frac{1/f}{s((J/f)s+1)} = \frac{K}{s(s\tau+1)}$$

with $K = K_\mathrm{m}/nf$ and $\tau = J/f$.

When the parameters are known, standard controller design tools like bode plots, root loci and so on can be used to design a controller for this system. A disadvantage is that in the process of converting the ‘physical model’ in the form of an iconic diagram with physical meaningful parameters into a block diagram with gain factors K and time constants, the relation with the physical reality is easily lost.

Based on the model of Fig. 7, we can design a controller, e.g. by means of using root locus. When the system would be available, we could do an experiment and determine the parameters K and τ. Using measurements from the real system, we could identify the parameters K and τ with a configuration as given in Fig. 8.

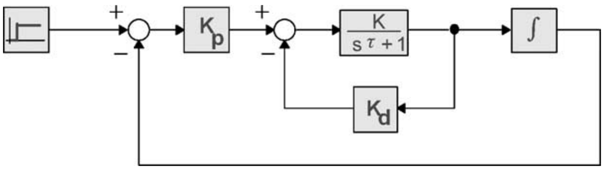


Fig. 10. PD-controlled model.

After optimization of the difference between the angle of the load and the model output we find the following responses for the two velocities (Fig. 9).

We find that $K \approx 10$ and $\tau \approx 0.85$.

Based on the model developed, a PD-type controller with proportional and tacho feedback, seems appropriate (Fig. 10).

After tuning K_p and K_d we find the following responses (Fig. 11).

However, when we apply this controller to the real system, we see that the system is unstable, indicating that our initial model was too simple, or in other words, not competent for this problem. Disregarded dynamics of the motor and, even more likely, of the flexibility in the belt of the transmission, which is visible in the open loop response of the real system could be responsible for this problem.

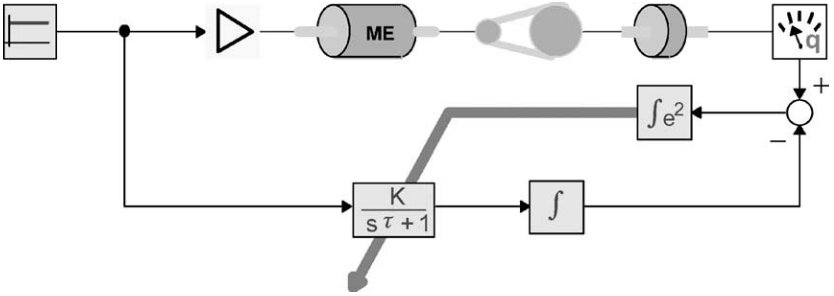


Fig. 8. Structure for parameter identification.

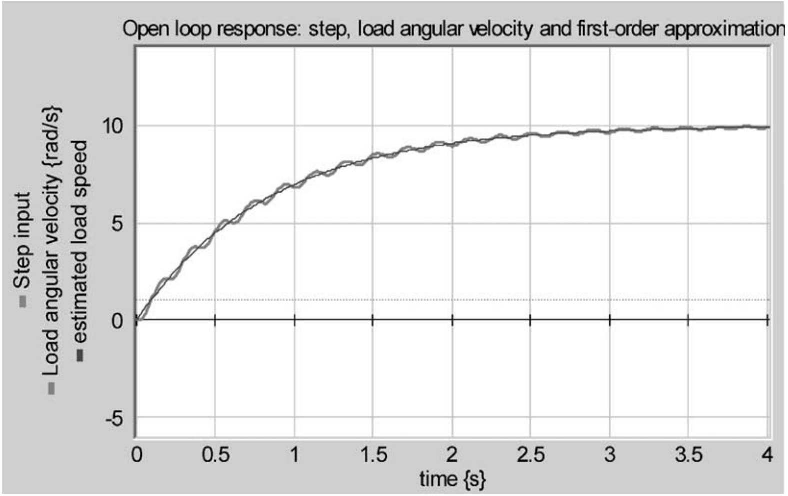


Fig. 9. Open loop responses of the real and identified system.

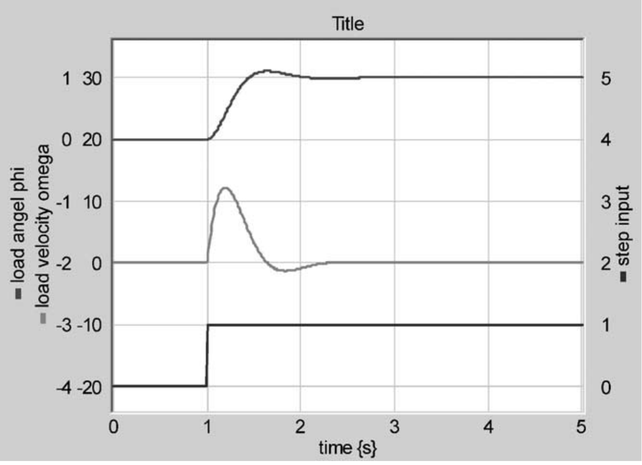


Fig. 11. Responses of the PD-controlled model.

We will reconsider the modelling process and follow an approach that preserves the relation with the physical parameters throughout the controller design and enables the choice between solutions in the controller (signal) domain or by alternative constructions.

The model of Figs. 6 and 7 is still very basic. This model enables the choice of a proper motor and transmission ratio, given the characteristics of the load. Once a candidate motor has been selected it may be desired to further detail the motor, e.g. by modelling the electrical properties of the motor and to take into account the compliance of the transmission when this is realised by means of two pulleys and a belt. This leads to three alternative models for the motor in Fig. 12. This indicates that various models can be meaningful, depending on the context. Because all models have the same interface (ports to the environment), submodels can easily be exchanged, allowing for polymorphic modelling (de Vries, 1994).

We will start with the description of Fig. 12c and assume that we use voltage steering. The motor is described by a number of *ideal physical elements*, each representing a basic physical relation. The motor has an electrical (EL) as well as a mechanical port (MECH) (Fig. 13).

Each of the *elements* in this figure can be described as an element with an electrical and/or mechanical port as explained before. The idea of ports is made more explicit in so-called bond graphs (Breedveld, 1989; Cellier, Elmqvist, & Otter, 1996; Gawthrop & Smith, 1996; van Amerongen, 2000). For the electrical elements these are the voltage difference over the element and the current through the element. For the mechanical elements these are the torque and the (angular) velocity. The products of these conjugated variables ($P = ui$ or $P = T\omega$) represent power.

If we go down a step further into the hierarchy, we arrive at the level of equations. For instance, an electrical resistor can be described by the equation:

$$u - iR = 0$$

or in the notation used in 20-sim:

$$p \cdot u = R * p \cdot i$$

where the variables $p{\cdot}u$ and $p{\cdot}i$ indicate the conjugated variables u and i of the electrical port p. Note that this is an *equation* and not an *assignment statement*. It could have been written equally well in the form:

$$p \cdot i = \frac{1}{R * p \cdot u}$$

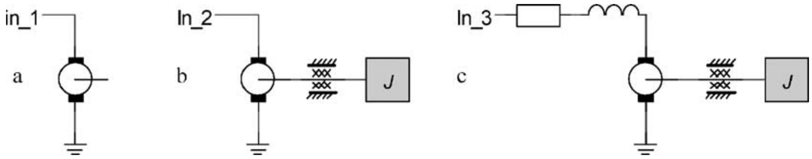


Fig. 12. Three different models of a DC motor: (a) basic power conversion from the electrical to the mechanical domain; (b) mechanical properties: motor mass and friction added; (c) electrical properties: resistance and inductance of armature coils added.

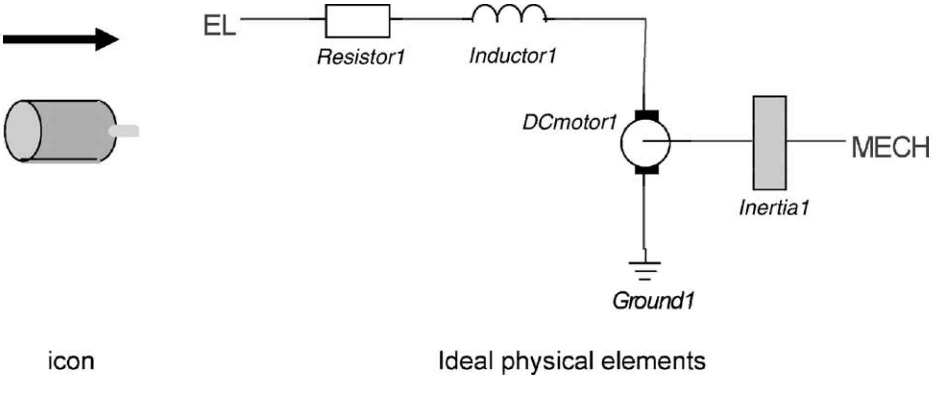


Fig. 13. Icon of the motor expanded to ideal physical elements.

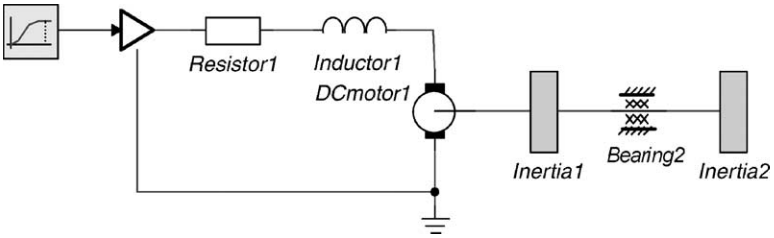


Fig. 14. Complete model in the form of ideal physical elements.

In a similar form, a mechanical inertia can be described by the equation:

$$\omega - \frac{1}{J}\int F\,\mathrm{d}t = 0$$

or

$$p \cdot T = J * \mathrm{ddt}(p \cdot \omega) \quad \text{or} \quad (p \cdot \omega) = \frac{1}{J \times \mathrm{int}(p \cdot T)}$$

where $d\,\mathrm{d}t(p{\cdot}\omega)$ denotes $\mathrm{d}\omega/\mathrm{d}t$ and $\mathrm{int}(p{\cdot}T)$ denotes $\int T\,\mathrm{d}t$. In case of an *R*-element there is no preference for one of the two forms. For the inertia-element, the integral form is preferred in the simulations. 20-sim determines the preferred causal form and derives the equations automatically. A warning can be generated when the preferred form cannot be used.

The energy flow or *power P* is the product of *two conjugated signals*, called effort (e) and flow (f):

$$P = ef$$

Examples of this expression in the mechanical and electrical domain are:

$$P = Fv \quad \text{or} \quad P = T\omega$$

$$P = ui$$

where F is force, v is velocity, T is torque, ω is angular velocity, u is voltage and i is current. The idea of ports and domain-independent modelling, will be treated into more detail in the second part of this professional brief.

When we expand the complete Fig. 6 we obtain Fig. 14.

When this model is processed a message pops up that indicates that inertia 2 has a dependent state. The two inertias in this model always have the same speed and therefore, they are dependent. They cannot have independent initial conditions. The message indicates that this element can only be written in the non-preferred derivative form:

$$T = J\frac{\mathrm{d}\omega}{\mathrm{d}t}$$

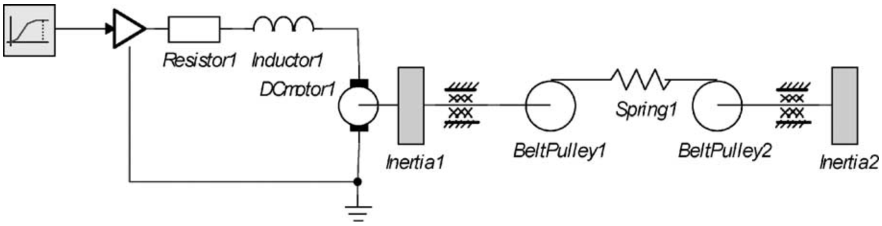


Fig. 15. Model extended with transmission.

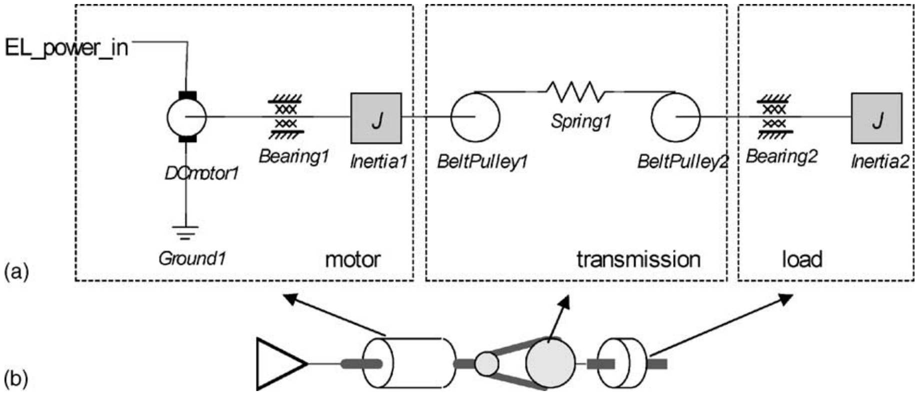


Fig. 16. More detailed model of the servo system: (a) all details shown as ideal physical models; (b) details hidden after creating a higher level model.

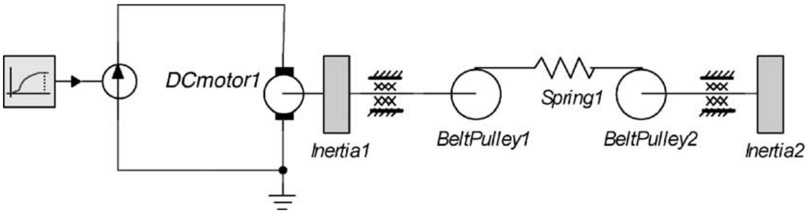


Fig. 17. Model with current amplifier.

There are several ways to deal with this problem.

1. The two inertias can be combined into one inertia (if possible this is done automatically in 20-sim. A message pops up that the dependency of the two inertias has been solved symbolically).
2. The transmission can be added, including some flexibility in the belt.
3. Dealing with the derivative causality can be done in the 20-sim simulator by means of an implicit integration algorithm.

If the flexibility is negligible solution 1 leads to the simplest model. On the other hand, the warning raises the question whether the flexibility of the belt can be disregarded indeed. If not, the model has to be extended with a spring element. It should be noted that this should not be done for numerical reasons only. In that case solutions 1 and 3 are to be preferred. If a very stiff transmission is added, this would result in high-frequency dynamics and lead to unnecessary slow simulations. On the other hand, if the flexibility is important, as it is in this system, the warning draws the designer's attention to the fact that the model may be oversimplified. And indeed, we observed already in the open loop response that resonances due to the flexible transmission were visible in the response and that a controller design based on the oversimplified system leads to an unstable closed-loop behaviour.

In Fig. 15, the transmission, including a spring element, has been added. Processing of this model does not produce any warnings.

This model could also be built up from a number of ‘submodels’ which can be combined into a higher-level model, where the model components contain these submodels (Fig. 16). Note that the latter model could have the same appearance as Fig. 5. But where Fig. 5 only presented the concept, the model of Fig. 16b is a high-level model which components are filled with submodels.

It is well known that application of a current amplifier has advantages compared with a voltage amplifier. When the voltage amplifier of Fig. 15 is replaced by a current amplifier we get a warning again: *Solved dependent state symbolically* (*Inductor1/p·i*).

Close inspection of the electrical circuit reveals that the resistor and the inductor in the electrical circuit can be left out of the model without changing the system behaviour. An (ideal) current amplifier completely determines the current in the electrical circuit. After removal of these two elements, the warning disappears. This leads to the diagram of Fig. 17.

This example illustrates how modern software can help to come up with a model that has the complexity that is needed for a particular problem. The user is warned for over simplification and can, e.g. choose whether to explicitly consider the flexibility in the system or not, or to consider the current amplifier really as an ideal current amplifier. Adding the flexibility makes the model a bit more complex, but also more realistic. Leaving out the electrical components simplifies the model, while keeping it competent for the problems we want to consider.

Physical models, in the form of an iconic diagram, based on connecting elements by means of power ports, may help in this modelling process. The user can select the preferred view, whether this is a bond graph, an iconic diagram with ideal physical element or a view using higher-lever submodels, like in Fig. 16. In the next section, the use of this model for the design of controllers will be shown.

3.3. Control system design methodologies

Many processes can be reasonably well controlled by means of PD-controllers. This is due to the fact that these processes can be more or less accurately described by means of a second-order model. Tuning rules, like those of Ziegler Nichols, enable less experienced people to tune such controllers. Relatively simple models can also describe many mechatronic systems. A mechatronic system mostly consists of an actuator, some form of transmission and a load. A fourth-order model can properly describe such a system. The performance-limiting factor in these systems is the resonance frequency. A combination of position and tacho feedback (basically a PD-controller) can be applied here as well. But due to the resonant poles proper selection of the signals to be used in the feedback is essential. Efforts have been made (Coelingh, 2000; Coelingh et al., 2000; Groenhuis, 1991) to derive recipes for tuning such systems, in addition to selecting the proper feedback signals. Computer support tools are essential to enable less experienced designers to use these recipes (van Amerongen, Coelingh, & de Vries, 2000–2002).

Coelingh (2000) and Coelingh et al. (2000) describe a structural design method for mechatronic systems. The method starts with reducing the conceptual design to a fourth-order model that represents the dominant properties

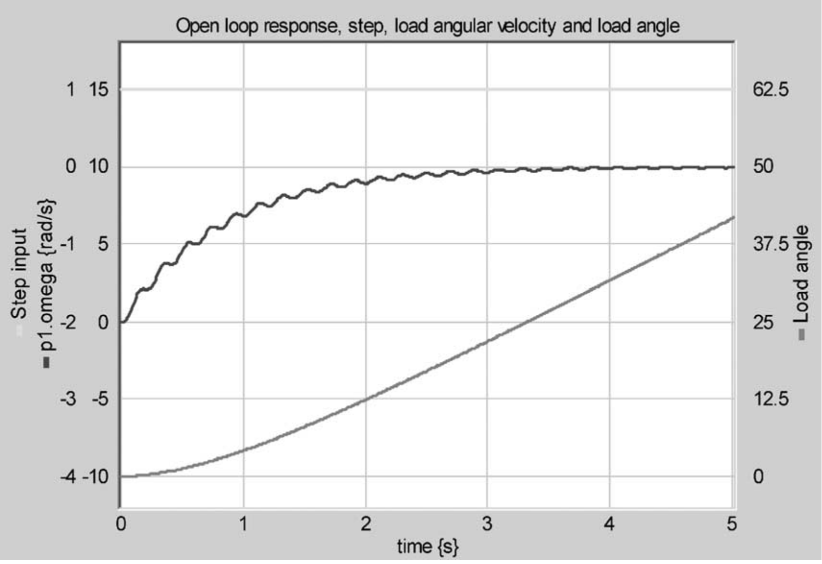


Fig. 18. Open loop responses.

of the system in terms of the total mass to be moved and the dominant stiffness. This model still has physical meaningful parameters. In this model, appropriate sensors are chosen, as well as a path generator. In the conceptual design phase, a simple controller is developed and mechanical properties are changed if necessary. Then a more detailed design phase follows where also parameter uncertainties are taken into account. The servo-system example used here is also a fourth-order system and as such it is representative for many mechatronic systems.

3.4. Controller design

Here we will consider some simple aspects of the design of a controller for the servo system in order to illustrate the advantage of the use of physical models and to illustrate the need for an integrated design approach. We consider the model discussed before, a load driven by an electric motor, through a flexible transmission. The iconic diagram of this model is given in Fig. 17. After selecting candidate components, the model elements can be enhanced with realistic parameter values. The selected values are given in the table.

Element	Value	Units
Mechanical_Load1\Bearing1	$f = 0.1152$	N ms/rad
Mechanical_Load1\Inertia1	$J = 0.056$	kg m^2
Axis1\Spring1	$k = 3750$	N/m
Axis1\BeltPulley1	Radius = 0. 02	m
Axis1\BeltPulley2	Radius = 0. 08	m
DC_motor1\Dcmotor1	$K_m = 0.292$	N m/A or V s
DC_motor1\Inertia1	$J = 0.00262$	kg m^2
DC_motor1\Bearing1	$f = 0.0001$	N ms/rad

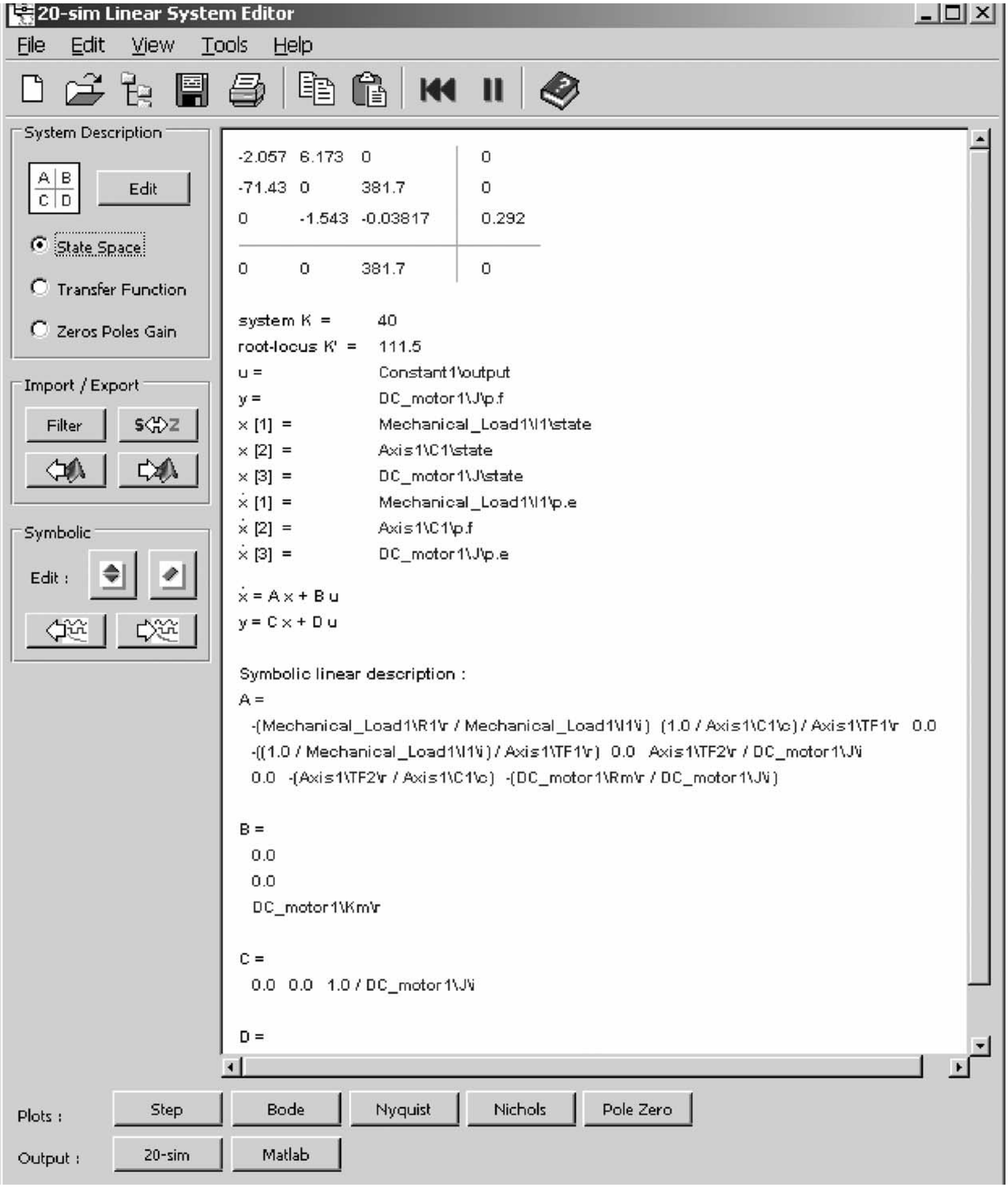
This enables that the system be simulated. Simulation equations are automatically generated from the graphical input in the form of iconic diagrams, like in Fig. 17.

In the step responses of Fig. 18, the resonance due to the flexible transmission is clearly visible. The numerical values in this example correspond with the values of the realised servo system in Fig. 4.

From the equations used for the simulation, 20-sim can automatically derive a model in a form suitable for controller design, such as a state-space description, a transfer function, or poles and zeros. A result of this automatic model procedure is given in Fig. 19. Note that also a state-space description in symbolic form is generated, thus making all the physical variables available in the state-space description.

An interface is provided to Matlab enabling, for instance, to use Matlab algorithms to compute the gains of advanced controllers like an LQR (optimal state feedback) or LQG controller (with a Kalman filter for state estimation and optimal state feedback). In fact these computer-supported design methods lead to a quick solution in this case. From the physical model, the A, B, C and D matrices are automatically generated. After selection of a proper performance index for the LQR-controller, the computation of the optimal feedback gains is easily obtained, either from Matlab, or by solving the equations in 20-sim itself. The same holds for the Kalman filter, which in addition to the state-space description of the process requires values for the variances of system and observation noise. Application of the Kalman filter enables full state feedback based on measurements of the load angle only. The diagram of the process together with an LQG-controller is given in Fig. 20 and responses in Fig. 21.

A properly designed P(I)D controller is able to perform almost similar, especially when the amount of noise is small. A first attempt could be to use only measurements of the load angle and load speed. This attempt fails, because feedback



```
-2.057   6.173    0          |  0
-71.43   0        381.7      |  0
0        -1.543   -0.03817   |  0.292
-----------------------------+---------
0        0        381.7      |  0

system K =        40
root-locus K' =   111.5
u =               Constant1\output
y =               DC_motor1\J\p.f
x [1] =           Mechanical_Load1\I1\state
x [2] =           Axis1\C1\state
x [3] =           DC_motor1\J\state
ẋ [1] =           Mechanical_Load1\I1\p.e
ẋ [2] =           Axis1\C1\p.f
ẋ [3] =           DC_motor1\J\p.e

ẋ = A x + B u
y = C x + D u

Symbolic linear description :
A =
  -(Mechanical_Load1\R1\r / Mechanical_Load1\I1\i)  (1.0 / Axis1\C1\c) / Axis1\TF1\r   0.0
  -((1.0 / Mechanical_Load1\I1\i) / Axis1\TF1\r)  0.0   Axis1\TF2\r / DC_motor1\J\i
  0.0   -(Axis1\TF2\r / Axis1\C1\c)  -(DC_motor1\Rm\r / DC_motor1\J\i)

B =
  0.0
  0.0
  DC_motor1\Km\r

C =
  0.0   0.0   1.0 / DC_motor1\J\i

D =
```

Fig. 19. Result of the ‘linearization’ procedure: numerical and symbolic state space description, that can be converted into a transfer function, pole-zero description and can be exported to Matlab.

of the load speed leads almost immediately to an unstable system as we saw before in the real system. Having a more complete model available now, this can be seen from the root locus for variations in the gain of the velocity feedback. From the responses of Fig. 18, 20-sim can easily determine the transfer function between the motor current and the load speed and plot the root locus (Fig. 22).

Fig. 22 clearly shows that even the smallest amount of velocity feedback will lead to an unstable system. This can also be observed in the Bode plot (Fig. 23) and Nyquist plot (Fig. 24). For very low values of the loop gain, the closed loop system will be unstable.

It is well known that feedback of the motor speed is a better solution. Using again the model of Fig. 17 to

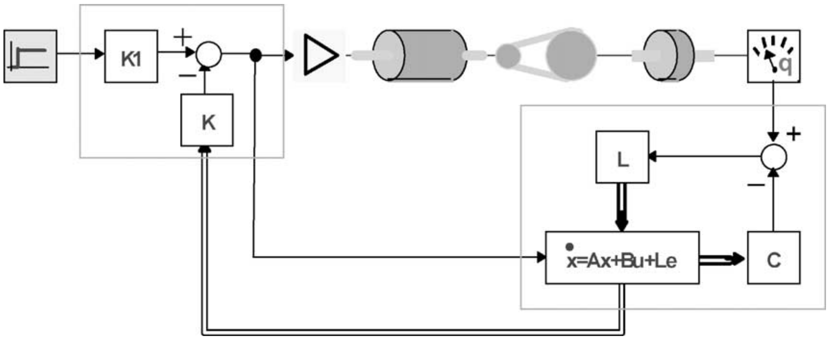


Fig. 20. Process with Kalman filter and state feedback.

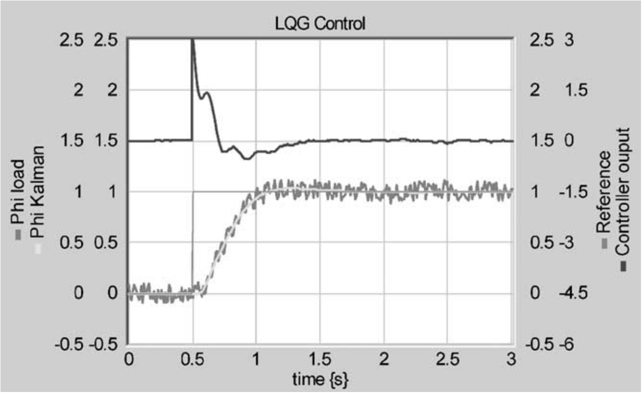


Fig. 21. Response of the LQG-controlled system.

determine the transfer from input current to motor speed yields the root locus of Fig. 25.

Complex zeros now accompany the complex poles and because they are close together their influence on the response will be almost negligible. The branch of the root locus on the real axis now shows the desired behaviour: moving the

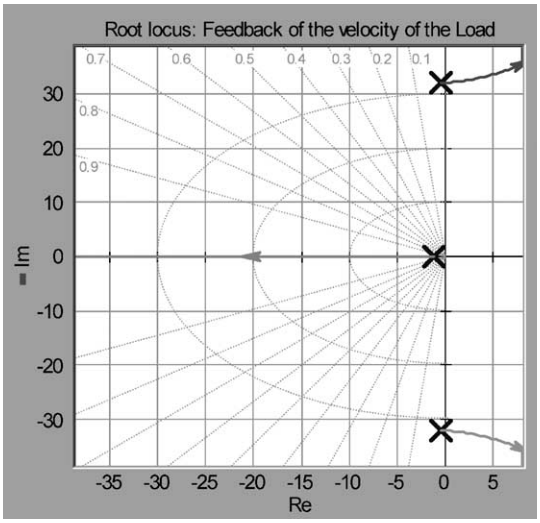


Fig. 22. Root locus for velocity feedback of load axis.

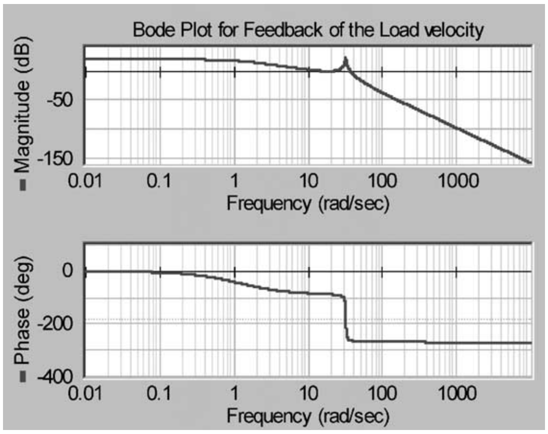


Fig. 23. Bode plot of the open velocity feedback loop.

dominant pole on the real axis to the left in the *s*-plane. The observations made here are generally applicable. A system with two resonant (complex) poles and no zeros, such as in Fig. 22 is difficult to control by means of a simple controller. If complex zeros accompany the resonant poles with an imaginary part smaller than that of the poles, stable control is easily achieved. In the frequency domain this is seen

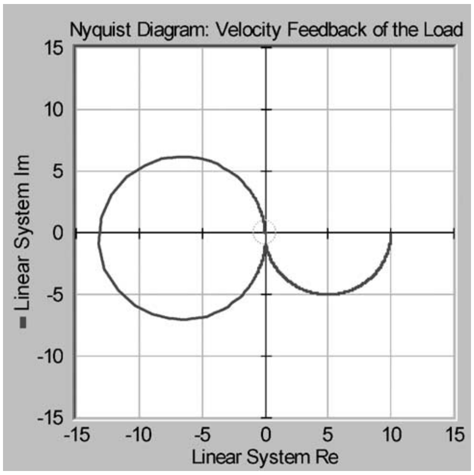


Fig. 24. Nyquist plot of the open velocity feedback loop.

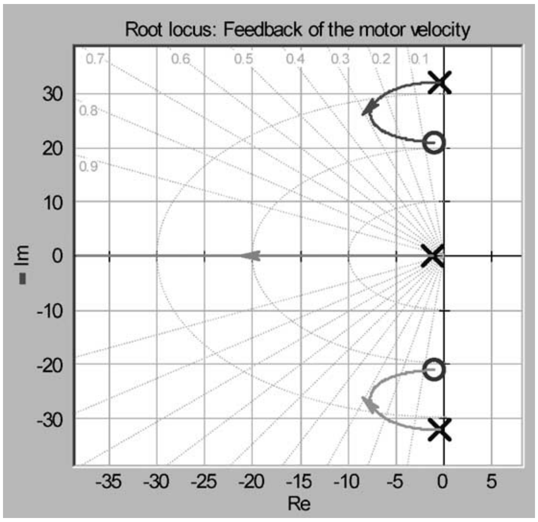


Fig. 25. Root locus for velocity feedback of motor axis.

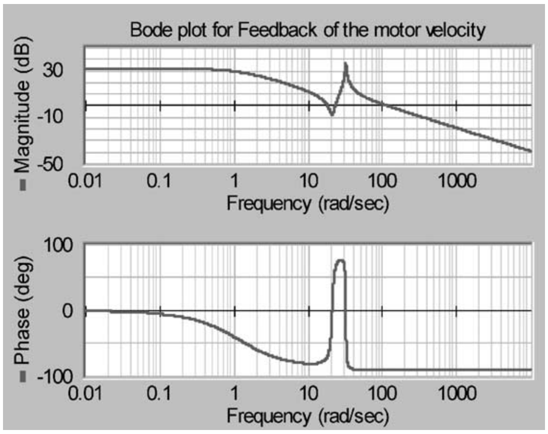


Fig. 26. Bode plot of the open loop transfer from input to motor speed: AR system. An anti-resonance is followed by a resonance. The phase lag never exceeds the ±90°, yielding always a stable feedback system.

as an anti-resonance (A), followed by a resonance (R) (type AR, Figs. 26 and 27).

Combining the feedback of the motor speed with feedback of the load angle yields the PD-controller structure of Fig. 28 and the responses of Fig. 29. Except for the fact that noise is not considered here, there is not much difference with the responses of the system with the Kalman filter, although the PD-controlled system is simpler.

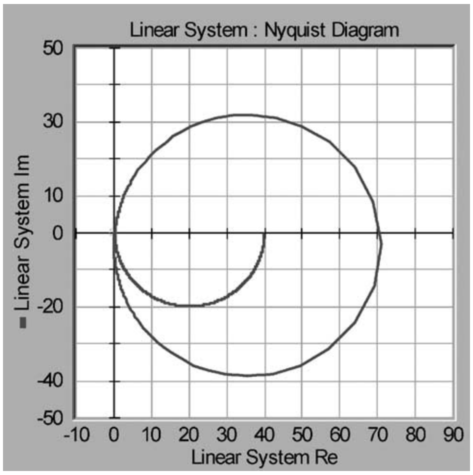


Fig. 27. Nyquist plot of the open loop transfer from input to motor speed.

An RA-system, where the resonance frequency is lower than the anti-resonance frequency (the imaginary part of the poles is smaller than that of the zeros), is just as difficult to control as in the case of only resonant poles. The existence and location of resonant zero's is completely determined by the (geometrical) location of the sensors in the mechanical system. A careful choice of these sensor locations is, therefore, crucial for the successful application of a controller. This choice can easily be verified with the types of models described here.

3.5. Increasing the bandwidth

The bandwidth of the controlled system in Fig. 29 has been chosen such that a reasonable response was obtained, without exciting the complex poles due to the flexible transmission. This implied a bandwidth of the close system of approximately 6 rad/s as the resonant poles are located approximately in $\pm 30j$. When we want to further increase the bandwidth, this cannot be done by simply increasing the gains of the controller. The only way to increase the bandwidth is to remove the bandwidth limiting factor, i.e. the resonant poles should be moved to a higher frequency. Let us assume that we select a stiffer transmission and change the compliance of the flexible belt from 3.75×10^3 N/m to 2.5×10^5 N/m. Using the 'model linearization' feature, we observe that the resonant poles move to $\pm 260j$, resulting in the Bode plot of Fig. 30.

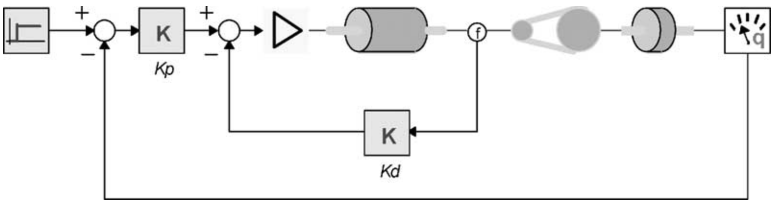


Fig. 28. Servo system with PD-controller.

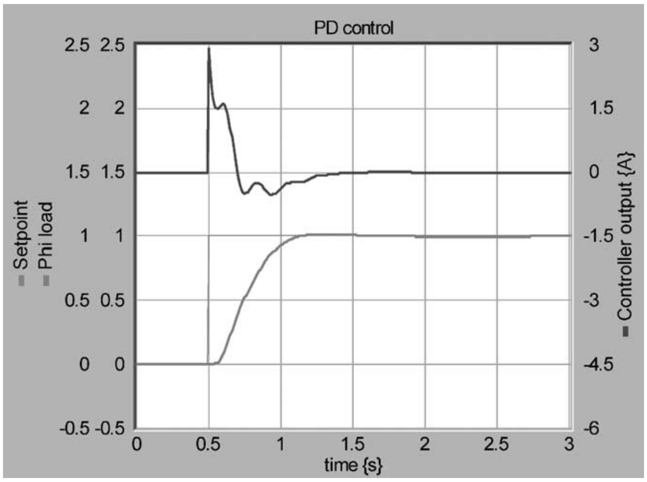


Fig. 29. Responses of the system of Fig. 15.

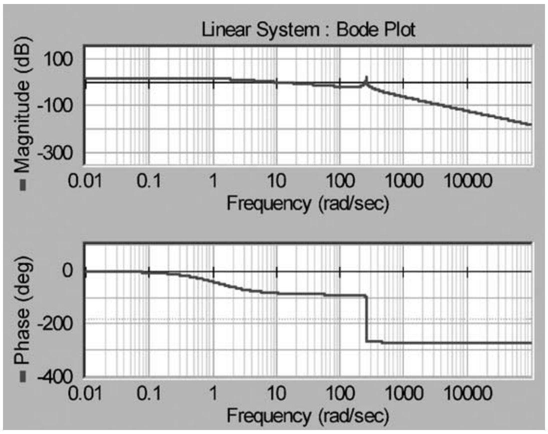


Fig. 30. Bode plot for a system with stiffer transmission.

The controller gains can now be increased from $K_\mathrm{p} = 3$ and $K_\mathrm{d} = 0.15$ to $K_\mathrm{p} = 200$ and $K_\mathrm{d} = 1.7$, resulting in the step response of Fig. 31. As a comparison also the closed loop Bode plots of the compliant and stiffer system are given in Fig. 32, where the increase in the bandwidth of the stiffer system is clearly visible.

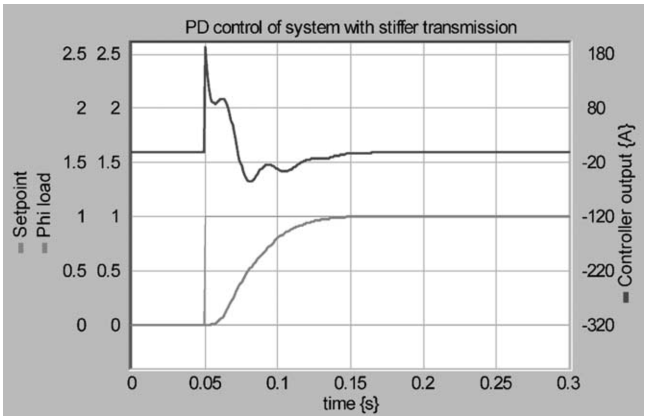


Fig. 31. PD control of system with stiffer transmission.

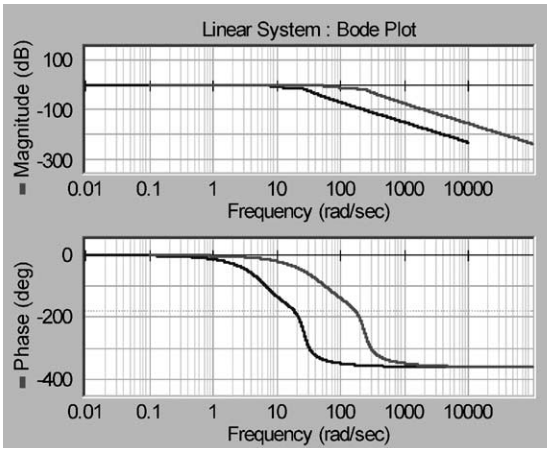


Fig. 32. Closed-loop Bode plots of the low- and high-bandwidth systems.

4. Direct optimization of a physical variable

Suppose that the belt in the transmission has a limited strength. Using the state-event feature of 20-sim the moment of breaking of the belt can be exactly determined. This is shown in Fig. 33. Notice that because of the use of a physical model, the force signal is easily available.

Using a path generator can prevent excessive forces in the belt. In its basic form the path generator smoothes the reference step and filters out frequencies that excite the resonance frequencies. Further improvement is possible by adding zeros to the path generator that suppress those frequencies explicitly. The optimal situation with this configuration is achieved when an optimization algorithm explicitly minimizes the forces in the belt by tuning the location of the zeros. The model used for optimization is shown in Fig. 34 and the forces in the belt before and after optimization are shown in Fig. 35.

5. Design of a mobile robot

A typical example of the early design procedure is the conceptual design of a mobile assembly robot. Already in a very early stage of the design conflicting demands have to be resolved. Such a robot should be able to collect parts all around a production facility and do the assembly while driving. Because a high accuracy is required between the gripper of the robot and the surface where the parts are located, it is important that floor irregularities and vibration modes of the structure do not prevent proper assembly. On the other hand, the path controller, partly based on dead reckoning (i.e. measuring of the wheel speed and orientation) requires that the wheels be very stiff. Damping of disturbances has to be realised by another means of suspension. This has led to the concept of an upper frame and a lower frame, connected by means of springs (Fig. 36).

The robot can be mounted at the upper frame and should have sufficiently bandwidth such that the position error

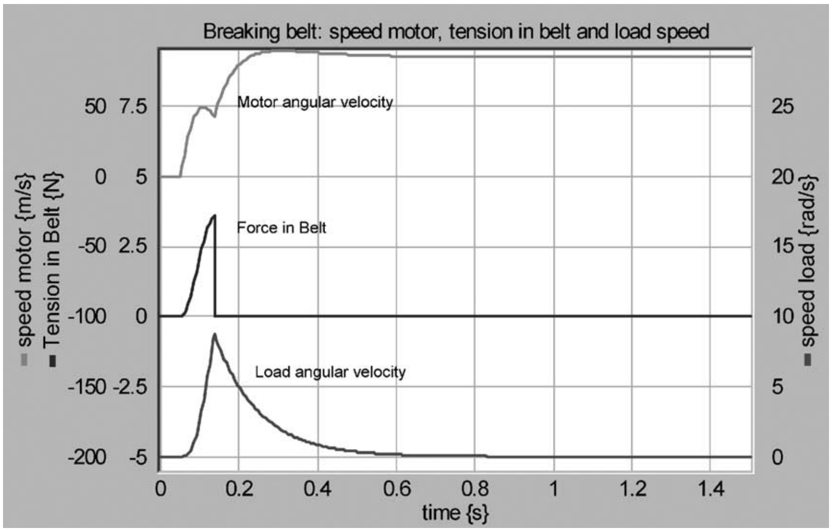


Fig. 33. Breaking of the belt.

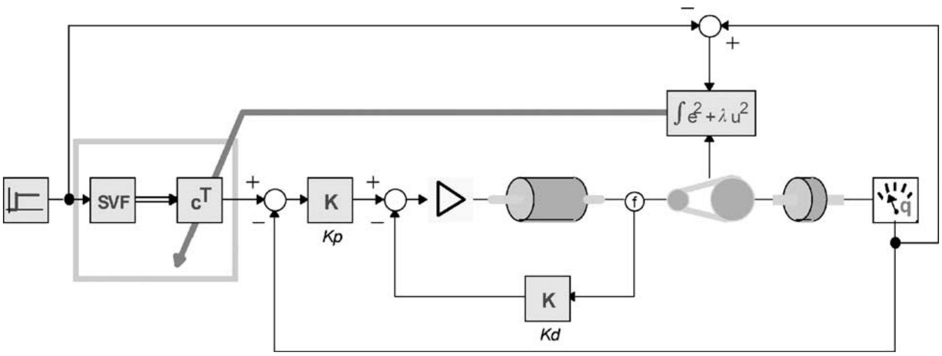


Fig. 34. Structure used for optimization.

($e_{\text{tip}} = z_{\text{tip}} - z_{\text{upper frame}}$) between the tip of the robot (z_{tip}) and the upper frame ($z_{\text{upper frame}}$) is small enough.

The next step is to derive a simple model, in order to have some parameters for the weight distribution and the stiffness and damping of the springs. In the model of Fig. 37, the robot is confronted with a bump in the floor at a speed of 1 m/s.

Based upon the payload—mainly the weight of the batteries—the total mass of the vehicle was estimated to be

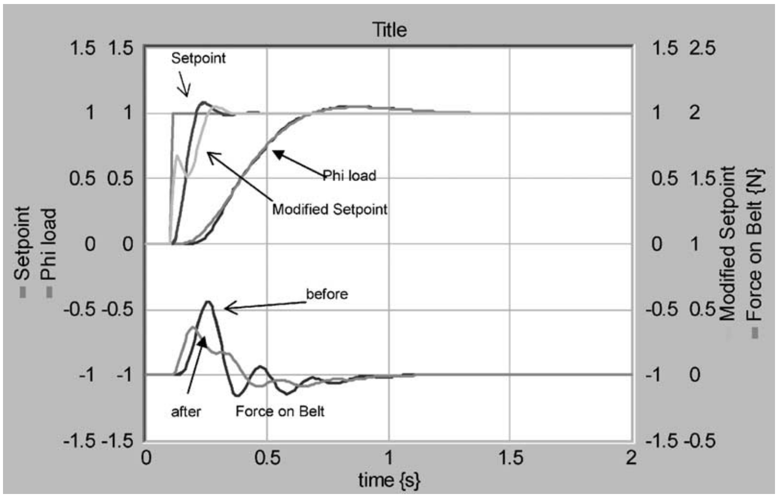


Fig. 35. Simulation results before and after optimization.

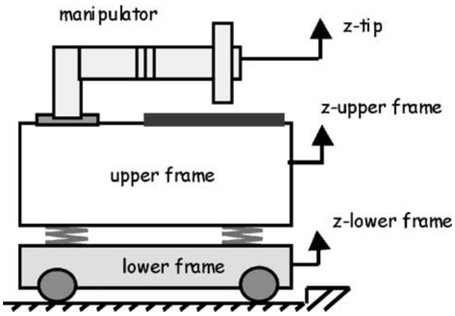


Fig. 36. Conceptual design of the mobile robot.

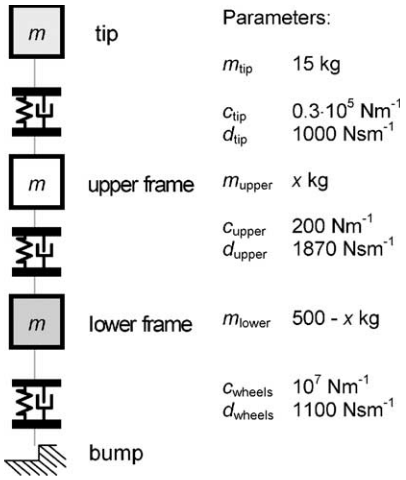


Fig. 37. Simple model with ideal physical elements to compute the error e_{tip}.

500 kg. Stiffness and damping of the wheels follow from the demands for the accuracy of the position estimation. The mass and bandwidth of the controlled manipulator were already known from other studies, yielding the effective stiffness and damping for the robot tip. When also initial estimates of the stiffness and damping of the springs between the upper and lower frame are made, the only parameter to be varied is the weight distribution between the upper frame and lower frame. By using the optimization feature of 20-sim, the optimal weight distribution can easily be found. In order to minimize the error between the tip of the robot and the upper frame, the weight has to be placed as much as possible in the upper frame (Fig. 38).

This example illustrates how the mechanical configuration of the system is determined by the requirements for good path control and accurate control of the assembly task.

A next step could be to optimize the properties of the suspension between upper and lower frame. This will further improve the error. This decision made in a very early stage of the design directed other design decisions. After completion of the project it appeared that the different parameters of the final construction were close to these early estimates (Fig. 39).

6. Port-based modelling of dynamic systems

If modelling, design and simulation of (controlled) systems is to be discussed, some initial remarks at the meta-level are required. It should be clear and it probably is, due to the way it is phrased next, that no global methodology can exist that would deal with each problem that might emerge. In other words, no theory or model can be constructed independent of some problem context. Nevertheless, in practice, models are often considered as constructs that can be independently manipulated, for instance in a so-called model library. Without some reference to a problem context, such a library would be useless, unless there is an implicit agreement about some generic problem context. However, it needs no further explanation that such a foundation is rather weak,

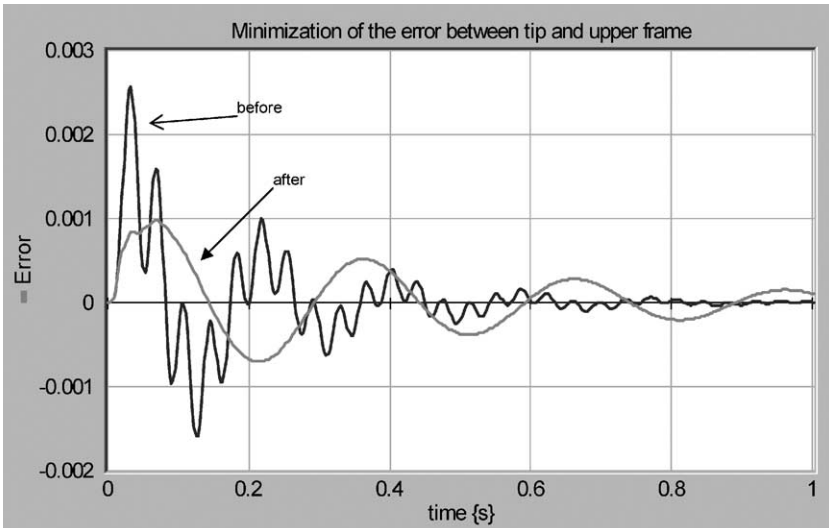


Fig. 38. Error of the tip before and after optimization of the weight distribution between upper and lower frame.

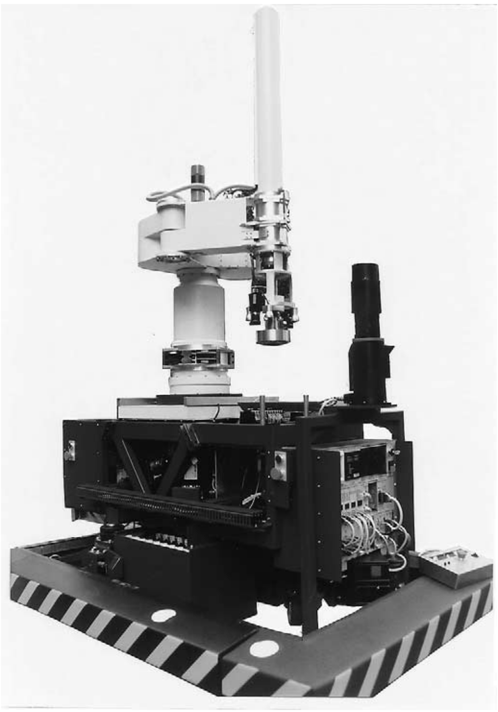


Fig. 39. The mobile robot (MART) after completion.

as implicit agreements tend to diverge, especially in case of real world problems.

Herein, we will focus on the generic problem context of the dynamic (i. e. in time) behaviour of systems that belong to the area of the engineer and the mechatronic engineer in particular, like the ones discussed in the case studies. These systems can be roughly characterised as systems that consist for a large part of subsystems for which it is relevant to the dynamic behaviour that they obey the basic principles of physics, like conservation laws and the positive-entropy-production principle. The other part consists of submodels for which the energy bookkeeping is generally not considered relevant for the dynamic behaviour. Such parts are generally addressed as the signal processing part (‘controller’) that is commonly for a large part realised in digital form. This part of this contribution focuses on the description of the part for which energy bookkeeping is relevant for the dynamic behaviour, while keeping a more than open eye for the connection to the signal part, either in digital or in analogue form. The cases studies discussed earlier emphasise the connection with the signal part.

It is argued that port-based modelling is ideally suited for the description of the energic part of a multidomain, sometimes also called multi-physics, system or subsystem. This means that the approach deals with mechatronic systems by definition and even beyond those.

7. Multiple view approach

It was already mentioned that commonly energy plays only a role in a part of a system. Furthermore, it is often fruitful to be able to look at all parts of a system from more than one point of view. This has been formalised as the so-called *multiple view approach* that is particularly well supported by window-based computer tools: a number of graphical representations like iconic diagrams (domain dependent), linear graphs (more or less domain-independent, but limited to the existence of analogue electric circuits (Shearer, Murphy, & Richardson, 1971)), block diagrams (computational structure), bond graphs (domain-independent), etc. as well as equations (mathematical structure) can serve as model representations in different windows. The tool in which all examples of this contribution are treated, 20-sim, has been designed on the basis of the multiple view approach. Possible views are: equations, block diagrams, (multi-)bond graphs, transfer functions, state-space representations (system, in- and output matrices), time responses, phase planes, functional relationships, step responses, bode plots, pole-zero plots, Nyquist diagrams, Nichols charts and 3D-animation. Where possible, automatic transformation is provided and results are linked.

The port concept allows a domain-independent notation called bond graphs. The port-based approach has been taken as the underlying structure of 20-sim, formulated in the internal language SIDOPS (Breunese, 1996), which makes it the ideal tool for demonstration of the port-based and multiple view approaches. A more detailed introduction to ports, bonds and the bond graphs representation is given in Section 9. However, first some generic remarks about modelling should be made.

8. Modelling philosophy

8.1. Every model is wrong

This paradoxical statement seeks to emphasise that any model that perfectly represents all aspects of an original system is not a model, but an exact copy of that system (identity). When modelling one looks for simple but *relevant* analogies, not for complex identities. As a result, a model is much simpler than reality. This is its power and its weakness at the same time. The weakness is that its validity is constrained to the problem context it was constructed in, whereas its strength is the gain of insight that may be obtained in the key behaviours that play a role. In other words: ‘no model has absolute validity’. The resulting advice is that one should always keep the limitations of a model in mind.

8.2. A model depends on its problem context

Models should be *competent* to support the solution of a specific problem. This also means that any type of archiving of a model or submodel should always include archiving of the corresponding problem context. Without this context, the model has no meaning in principle.

Note that training of specialists and experts is often related to what is sometimes called a 'culture' and that they are said to speak a 'jargon'. This culture and jargon reflect the existence of a particular (global) problem context, even though this context is not explicitly described when models are made. For electrical circuit designers, this problem context consists of the behaviour of electric charges and in particular of the voltages and currents related to this behaviour, in a specific part of the space–time scale. This behaviour is such that electromagnetic radiation plays no dominant role. Mechanical systems mostly belong to another part of the space–time scale, although there may be considerable overlap.

These cultures and jargons easily lead to implicit assumptions, which, in turn, may lead to model extrapolations that have no validity in the specific problem context at hand due to the danger of ignoring earlier assumptions. These extrapolations often start from well-known class-room problems with analytical solutions like the model of a pendulum. In other words: 'implicit assumptions and model extrapolations should be avoided'. The resulting advice is that one should focus at the competence, not at the 'truth content' of a model.

8.3. Physical components versus conceptual elements

In all cases, it should be clear that (physical) components, i.e. identifiable system parts that can be physically disconnected and form a so-called physical structure, are to be clearly distinguished from (conceptual) elements, i.e. entities that represent some basic behaviour, even though they are sometimes given the same name. For example, a *resistor* may be an electrical *component* with two connection wires and some colour code (cf. Fig. 40a), while the same name is used for the conceptual *element* (commonly represented by Fig. 40b) that represents the dominant behaviour of the component with the same name, but also of a piece of wire through which a relatively large current flows.

Note that this model requires that the problem context is such that the component 'resistor' is part of a current loop in a network in which the behaviour of the voltages and currents plays a role. By contrast, other realistic problem contexts exist in which the dominant behaviour of the component 'resistor' is not represented by the element 'resistor', but by the element 'mass' or a combination of mechanical conceptual elements like mass, spring and damper. For example, when this component is to be rapidly manipulated in assembly processes, i.e. before it is part of an active circuit, this could be a competent model.

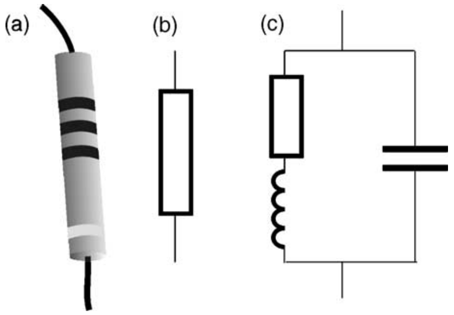


Fig. 40. Component electrical resistor (a) with two different conceptual models (b and c).

Often, not only the dominant behaviour of a component has to be described but also some other properties that are often called '*parasitic*' (cf. Fig. 40c), because they generate a *conceptual structure* and destroy the one-to-one mapping between components and elements that misleadingly seems to simplify modelling and design. Note that those areas of engineering (like electrical engineering) in which materials could be manipulated as to suppress all other behaviours than the dominant one, have been the first to apply network style dynamic models successfully. Also note that in our daily life we have learned to make quick intuitive decisions about dominant behaviours ('survival of the fittest'). This type of learning stimulates implicit and intuitive decisions that may fail in more complex engineering situations (counter-intuitive solutions).

Implicit assumptions are commonly not only made about the problem context, but also about the reference, the orientation, the coordinates, the metric and about 'negligible' phenomena. Famous class-room examples may have an impact on the understanding of real behaviour for generations, especially due to the textbook copying culture that is the result of what may be called a 'quest for truth' motivation, ignoring competence. A famous example is the false explanation of the lift of an aircraft wing due to the air speed differences and according dynamic pressure differences generated by its profile. This explanation has survived many textbooks, even though the simple observation that aeroplanes with such wing profiles can fly upside down falsifies this explanation in an extremely simple and evident way.

Other examples are models of which the behaviour changes after a change of coordinates: as coordinates are a modeller's choice, they cannot have any impact on the behaviour of the described system.

Not keeping an open eye for these aspects of modelling may lead to exercises that are documented in the scientific literature in which controllers are designed to deal with model behaviours that are due to imperfections of the model and that are not observed at all in the real system.

9. Use of ports in dynamic system models

The concept of a port is generated by the fact that submodels in a model have to interact with each other by definition and accordingly need some form of interface. In physical systems, such an interaction is always (assumed to be) coupled to an exchange of energy, i.e. a power. In domain-independent terminology, such a relation is called a power *bond* accordingly. This *bilateral* relation or bond connects two (power) *ports* of the elements or submodels that are interacting (Fig. 41).

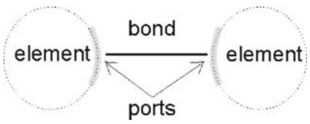


Fig. 41. Bond connecting two ports.

In the signal domain, this power is assumed to be negligible compared to the powers that do play a role, such that a signal relation may be considered a '*unilateral*' relation. Note that *ideal* operational amplifiers have an infinite input impedance and a zero output impedance in order to suppress the back-effect and to be purely unilateral, but can only be approximated by adding external power. The bilateral nature of the power relations (as opposed to unilateral signal relations) suggest the presence of *two* variables that have some relation to the power represented by the bond. These so-called power-conjugate variables can be defined in different ways, but commonly they are related by a *product operation* to a power P and in this case named effort e and flow f:

$$P = e \times f$$

In principle, the f- or flow-variable can be seen as the rate of change of some state, whereas the e- or effort-variable can be seen as the equilibrium-determining variable. Examples are the voltage and the current as the effort and the flow, respectively, of the electrical domain and the force and the velocity as the effort and the flow, respectively, of the mechanical domain. Note that the common approach to use two types of storage, C- and I-type, prevents that this distinction between flow and effort as rate of change of state and equilibrium determining variable, respectively, can be used in modelling. The so-called Generalised Bond Graph or GBG approach resolves this problem, leaving the discussion about the force–voltage versus force–current analogy a non-issue (Breedveld, 1982, 1984). This is not further discussed in order to adapt to the conventional background of the reader, despite the fact that the 'rate of change' versus 'equilibrium-determining' aspects of these variables are powerful tools to support initial modelling decisions.

In this second part only an intuitive introduction to bond graphs is given mainly focussing on those aspects required to introduce the port concept in such a way that it can also be applied to iconic diagrams. In the first part, a port-based modelling and control problem was already extensively described using the iconic diagram notation mixed with block diagrams. For more extensive information about the bond graph notation the reader is referred to the literature.

9.1. Dynamic conjugation versus power conjugation

The two signals of the bilateral signal flow representing a physical interaction are dynamically conjugated in the sense that one variable represents the rate of change of the characteristic physical property, like electric charge, amount of moles, momentum, while the other variable represents the equilibrium-determining variable. This is called dynamic conjugation. As long as no other domains are of interest, the concept of energy is not particularly relevant, such that these variables do not need to be related to a power like the effort and flow discussed earlier. Examples are: temperature and heat flow (product is not a power, heat is not a proper state if other domains are involved), molar flow and concentration or mole fraction (product is not a power), etc. The power-conjugated variables are a subset of the dynamically conjugated variables.

9.2. Multidomain modelling and the role of energy

The previous subsection illustrates that the concept of a domain-independent conserved quantity, the energy, is crucial for the consistent interconnection of physical phenomena in different domains. The discussion of basic behaviours in Section 13 is based on this and thus requires either the consistent use of power-conjugated variables or carefully defined domain transitions that are power continuous and energy conserving.

10. Computational causality

Purely mathematically speaking one can state that a subsystem with a number of ports, called multiport, is a multiple-input–multiple-output or MIMO system, of which the set of inputs and the set of outputs is not a priori chosen. The relation between the input and output variables, the so-called constitutive relation, determines the nature of the multiport. If the number of input variables is not equal to the number of output variables, this means that there has to be at least one unilateral *signal* port as opposed to a bilateral *power* port as the latter is by definition characterised by one input and one output. If this signal port is an input signal, the multiport is called modulated. Modulation does not affect the power balance, in other words: no energy can be exchanged via a signal port.

Although ports and bonds illustrate that two bilateral signals are involved in a relation, *no a priori choice* about the *direction* of the corresponding signals needs to be made. This is an important distinction with a conventional MIMO system (Fig. 42). A particular choice of this computational direction or *causality* is needed before a set of computable relations can be found or some particular analysis may be performed. Often, such a 'causality assignment' leads to computational forms that are not obvious and would have led to modelling problems in conventional approaches, in particular when domain boundaries are crossed (cf. the remarks about Paynter's motivation in the introduction). As a result, bond causality does not only support the solution of computational and analytical issues, it also gives the modeller immediate feedback about the physical meaning of his modelling decisions and the trade-off he has to make

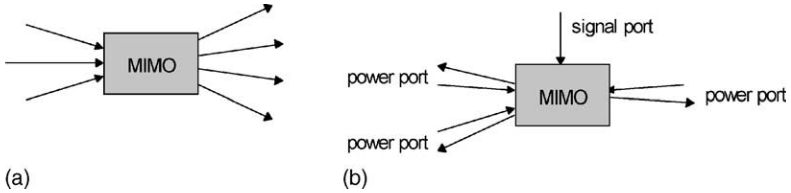


Fig. 42. (a) Conventional MIMO; (b) MIMO system with bilateral power ports and modulating signal ports.

between conceptual and computational complexity. If information about its causality is represented on a bond in a bond graph by means of a so-called 'causal stroke' (Fig. 43), the bond graph simultaneously represents both conceptual and computational structure. From the latter point of view a bond graph can be seen as a condensed block diagram. However, although any causal bond graph can be converted into a block diagram, not any block diagram can be converted into a bond graph, as conceptual structure is lost in the first transformation.

The trade-off may be illustrated by the simple example of a rigid constraint between two rigid bodies. Conceptual simplicity leads to a causal problem—the DC-servo system example already showed that a loop emerges containing an integration and a differentiation, i.e. a 'net' algebraic loop (cf. Fig. 14)—and consequently to numerical complexity (DAEs). A DAE is a mixed set of differential and algebraic equations that cannot be solved straightforwardly by means of explicit numerical integration (e.g. with the common Runge–Kutta 4th order method). However, the way in which the causal problem emerges in the model during causal analysis (to be discussed in Section 15) clearly suggests how the model can be modified in order to prevent the causal problem. In this case the rigid constraint can be replaced by an elastic element, i.e. a finite rigidity. Although this gives the model some more conceptual complexity, the numerical (structural) complexity is reduced, due to the fact that the resulting equations are a set of ordinary differential equations (ODEs) that can be solved by explicit numerical integration schemes (Ascher & Petzold, 1998), see also the course slides at http://www.npac.syr.edu/users/gcf/CPS615NI95/. Note that the model still needs a stiff constraint and thus introduces dynamics at a time scale that is not of interest. This means not only means that both options can be a solution depending on the problem context, the available tools, etc. but also that a third solution can be obtained, viz. a symbolic transformation of the model as to eliminate the dependent inertia. In other words: two rigidly connected rigid bodies may be considered as one rigid body. This possibility is also induced by the causal analysis of the bond graph model that will be discussed in Section 15.4.

11. Example of the use of the port concept

Only the actual use of the port concept can fully clarify its importance. Therefore, a simple example is discussed to illustrate the port concept.

A component that may be used in mechatronic systems, but in which the control is not realised by (digital) electronic signal processing, but physically, i.e. as an energetic process, is taken as an example. This choice is made in order to focus on the multidisciplinary modelling part on the basis of power ports. In the second part, the examples will be focussed on the combination of the power and signal ports.

11.1. Problem context

Under some operating conditions of a *low-vacuum control valve* spontaneous, self-sustained oscillations occur. Given the purpose of the valve, viz. to maintain a constant 'low' vacuum in particular in medical applications this behaviour is clearly undesired. In order to solve this problem insight is to be obtained in the source(s) of this behaviour and the design parameters of the system that should be modified in order to prevent this behaviour.

Some simple oscilloscope measurements of these oscillations, mainly showing shape and frequency, are available. A construction drawing of the valve with data on geometry and used materials is available.

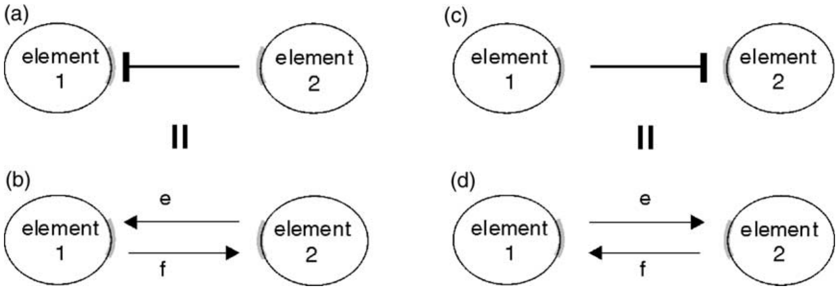
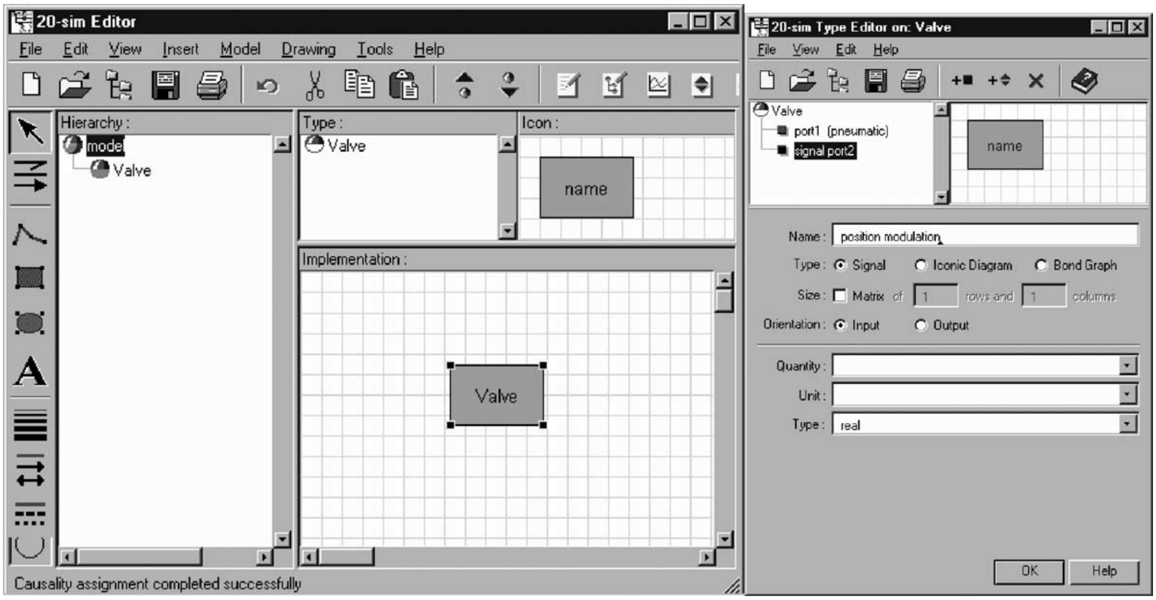


Fig. 43. Causal stroke (red in (a) and (c)) showing computational direction of effort signal (red in (b) and (d)).



Hierarchy: model, Valve

Type: Valve

Valve
- port1 (pneumatic)
- signal port2

Name: position modulation

Type: ● Signal ○ Iconic Diagram ○ Bond Graph

Size: ☐ Matrix of 1 rows and 1 columns

Orientation: ● Input ○ Output

Quantity:

Unit:

Type: real

Causality assignment completed successfully

Fig. 44. Definition of power and signal ports in 20-sim.

11.2. Functional description of the valve

The intended basic operation of this control valve is that an orifice can be opened and closed by a valve that is connected to a diaphragm loaded by a coil spring. Changing the position of the other end of this spring with a screw knob can set its pretension. The diaphragm is part of the wall of the valve chamber that is at one end connected to the 'supply' pressure (a relatively high under pressure or 'vacuum') via the valve opening and at the other end via an orifice and a hose to the 'mouth piece' to suck superfluous body fluids away as used by a dentist or during operations. Given some desired low under pressure or 'low-vacuum', the pressure difference over the diaphragm will cause the valve opening to get smaller if the desired pressure gets too low. Due to the increasing flow resistance of the variable orifice the pressure difference with the supply pressure ('high' vacuum) will increase again vice versa.

11.3. Analysis

(1) In a regular valve, a screw modulates the position of the body of the valve. Note that the fluid acts with a force on this body, trying to move it out of the valve seat. The reason that the fluid cannot displace the valve body while the human hand can, is the presence of the transforming action of the screw/spindle. This amplifies the static friction of the screw seen from the translating port of the screw/spindle. However, as this static friction is only overcome during a hand turning the valve and the dynamics of this process are at a completely different time scale than the flow phenomena in the valve, a change in position of the valve body is commonly modelled as a modulation of the flow resistance of the valve. Hence, a position-modulated resistor can describe the dominant behaviour of a valve (Fig. 44).

(2) Feedback can be introduced by a diaphragm (membrane) that transforms the difference in pressure at its sides into a force that can cause a displacement. By connecting the body of the valve to the membrane such that an increasing pressure difference will close the valve and a decreasing pressure difference will open it, it will thus have a counteraction in both cases, i.e. a negative feedback. The relation between force and displacement is characterised by the stiffness of the diaphragm. It needs to be increased in order to attenuate the position changes of the valve body. This is achieved by connecting a spring. By connecting the other end of the spring to the screw, the screw can be used to change the setpoint for the pressure difference. The model of the valve has to be at least extended by an ideal transformer (TF) to represent the dominant behaviour of the diaphragm, an ideal spring to represent the elasticity of the spring and the diaphragm and a modulated force source to introduce the pretension of the setpoint. Fig. 45 shows this with a mixed use of bond graph (TF, valve), block diagram (modulation and signal generator) and

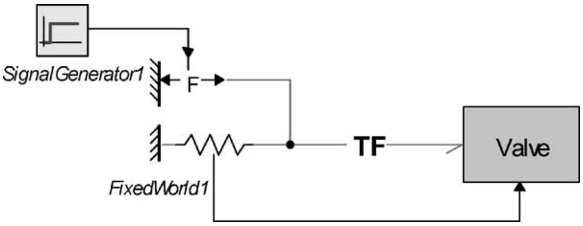


Fig. 45. Mix of block diagram, iconic diagram and bond graph representation (20-sim editor screen).

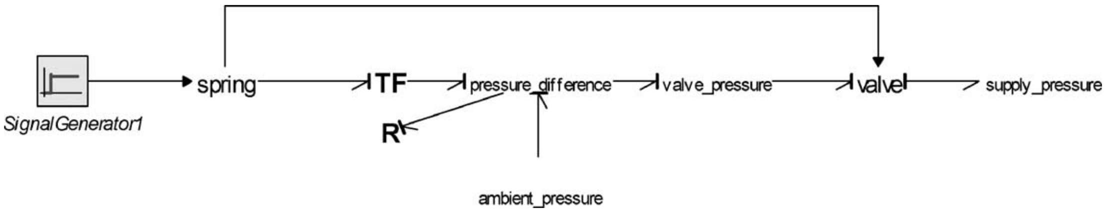


Fig. 46. Addition of boundary conditions, flow resistor and structural details (pressures) in a mixed notation.

iconic diagram elements (spring, force source and fixed world).

(3) Note that the source of the pressure difference described in step 2 has not been accurately defined. One might conclude that the pressure difference between some supply pressure and the ambient pressure is meant as these are the two obviously present pressures. However, this would cause the output pressure to fluctuate with the supply pressure, which is commonly not desired. Furthermore, the output pressure is required to cause some fluid exchange with the environment, i.e. some flow connection to the environment. As a consequence one is usually interested in setting the pressure difference between the output pressure and the supply pressure. This means that the valve needs to contain a more or less closed volume, the so-called valve chamber, in which the output pressure is allowed to be different from both the supply and the ambient pressure. Obviously, some opening needs to connect this chamber to the environment in order to allow the desired flow. The dominant behaviour of this restriction is that of an ideal (fluid) resistor, whether a hose is attached to the orifice or not. Parasitic behaviour as fluid inertia (in case of a long hose) may be added later when fine-tuning the model. Summarising, the following ideal elements are required in the model: a position-modulated resistor, a transformer, a spring and a resistor (Fig. 46). As the spring is the only dynamic element (containing an integration with respect to time) in this model, oscillatory solutions are not likely.

(4) At this point one might be inclined to bring the possibility of oscillatory behaviour into the model by adding an ideal mass to represent the dominant behaviour of the valve body. Together with the ideal spring it forms a (damped) second order system that has the potential of oscillatory solutions. However, such oscillations are not self-exciting and not self-sustained, unless the system would contain negative damping which would violate the laws of physics. In the sequel, it will turn out that the mass of the valve body is indeed required to capture the behaviour, but not in combination with the spring (Fig. 47). Note the change of position of some of the strokes (automatically generated by 20-sim) and the fact that the strokes can be followed from the R to the valve.

(5) The crucial element that needs to be added to the model is the compressibility of the air in the valve chamber (Fig. 48). The position that modulates the valve is (inversely) proportional to the flow through the valve. The capacitance of the valve chamber relates the displaced volume (first integration!) of this flow to the pressure in the chamber. Via the diaphragm this pressure acts with a force on the valve body. The resulting change of its momentum (second integration!) results in a change of its velocity. Finally this velocity causes its displacement (third integration!) and thus results in the position that modulates the valve resistor (closure of the loop). This loop contains three integrations that under certain conditions may have unstable solutions that are bounded by the nonlinearities of the model, like the valve body hitting the valve seat. As will be demonstrated in Section 15. 4, a bond graph representation of a port-based analysis provides this insight. At this point, this example should illustrate that modelling should be focussed on the relevant elementary behaviours present in a system, not merely on a (one-to-one) translation of the functional relations as the designer of the valve intended them, because this approach would never bring in the compressibility of the air in the valve chamber. The key elements in this model to represent the observed behaviour are: the nonlinear, position-modulated resistor, the valve body, the diaphragm and the capacitance of the valve chamber to create the third-order loop, but also the spring with its adjustable pretension, the fluid resistor at the inlet, the supply pressure and the valve body hitting the valve seat. The number of elementary one- and multiports is relatively small (8 linear and 1 nonlinear element; cf. the bond graph expansion in Fig. 49).

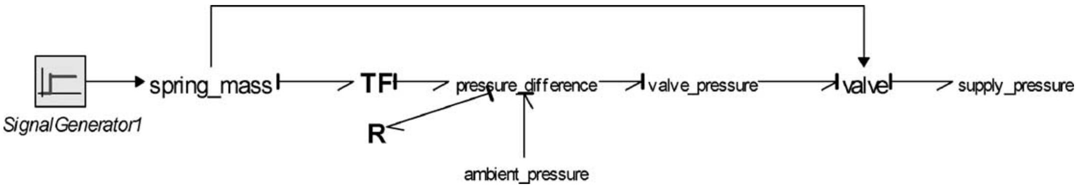
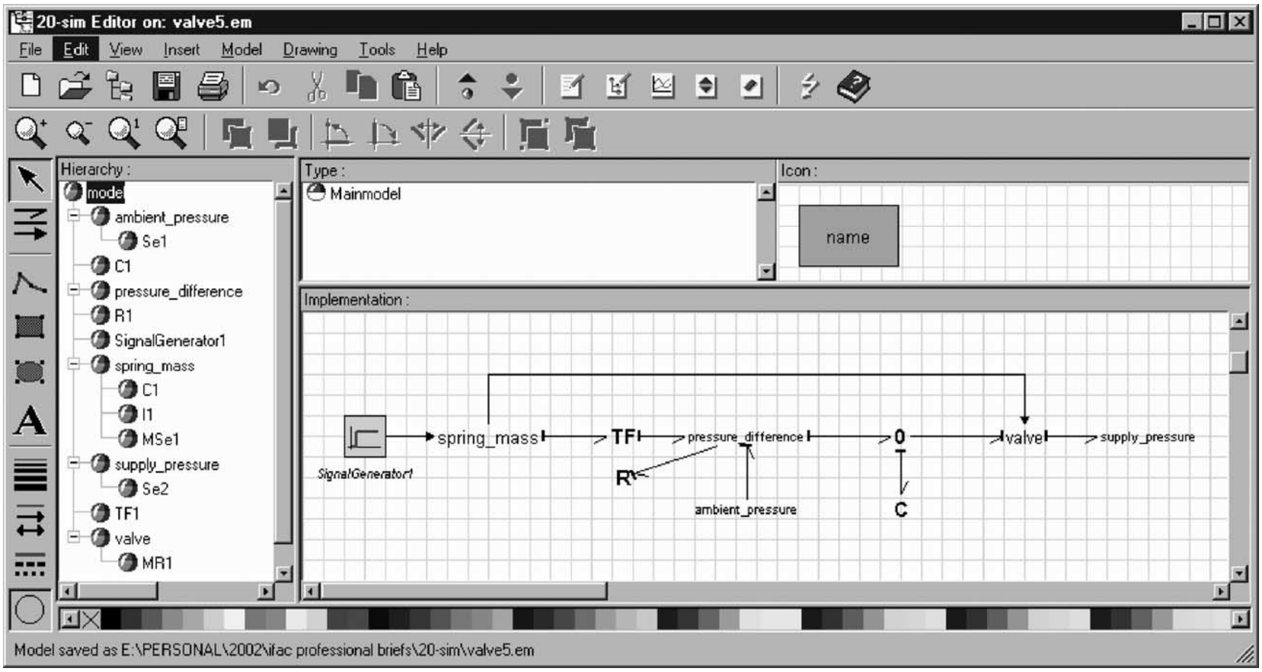


Fig. 47. Addition of the valve body mass.



Model saved as E:\PERSONAL\2002\ifac professional briefs\20-sim\valve5.em

Fig. 48. Addition of the compressibility of the air in the valve chamber (C).

(6) After identification of the proper parameter values from the provided measurement data, first simulation runs showed indeed self-starting and self-sustained oscillation with a shape that coincided with the shapes observed on the oscilloscope. The frequency of these first results was only 10% off the observed frequency. Fine-tuning of the model allowed these frequencies to be matched. However, the problem was in fact already solved before the parameter identification phase, because the process of setting up the *model structure* already indicated the crucial role of the valve chamber that was confirmed by an experienced senior craftsman at the work floor where these valves were produced and assembled. He then remembered that long ago the role of this valve chamber was identified by trial and error. A result that had been forgotten over the years and didn't play a role in the design of the new valve that was causing the oscillation problems.

Fig. 49 shows an exploded view of Fig. 48. This shows that a bond graph notation was used at a lower level. Even though this notation has not been introduced it is indicated that the little strokes (so-called causal strokes) also show the user here explicitly and in a simple way the existence of a third-order loop. The causality of the ports is derived automatically by 20-sim to obtain a computable set of equations for simulation. The same procedure is used in case of iconic diagrams are other representation that contain the concept of a port.

Note that the possibility of these oscillations is inherent to this particular design. None of the parts can be omitted or changed as to break the third-order loop. For this reason, every designer of such valves should have the insights discussed above in order to be able to choose the dimensions of the valve such that it never displays undesired behaviour in or near the range of operation. This insight is more related to model structure than to particular simulation results, although simulation results can help to identify the influence of the valve chamber size on the modes of operation.

Similar types of valves are not only used as low-vacuum control valves, but also as fuel-injection valves, pressure reduction valves, etc.

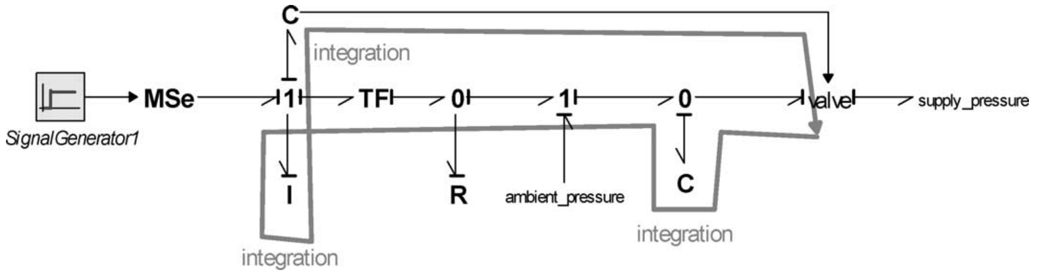


Fig. 49. 'Exploded view' of Fig. 48 showing the third-order loop via the causal stroke directions in a bond graph.

This example should give the reader some motivation to try and understand why a port-based approach provides this insight quite easily, although its use should be supported by sufficient knowledge of engineering physics. Furthermore, a user of this approach should be willing to see that straightforward application of library models does not result in a solution of the original problem, although it largely supports it.

After discussing the role of the choice of the system boundary, the basic elements of the port-based approach will be shortly introduced together with its domain-independent representation. Some more attention will have to be paid to the junctions, as they require the paradigm shift mentioned earlier. Next causal analysis will be discussed and illustrated on the immediate identification of the third-order loop in the example.

12. System versus environment: system boundary

The distinction between system and environment is determined by the role of these parts: the environment can influence the system, but not dynamically interact with it. In signal terminology: the environment may influence the system via inputs and observe the system via outputs, but the inputs cannot depend on these outputs, at least not at the time scale that is of interest. In case of normal use, a car battery, for example, may be considered the environment of a dashboard signal light, as the discharge will not affect the voltage in a considerable way. In other words, the car battery in this problem context (regular car use) can be modelled by a voltage source. However, in a context of a car being idle for three months (other time scale!) the car battery has to be made part of the system and dominantly interacts with the resistance of the bulb like a discharging capacitor. Such an RC-model is competent in this problem context to predict the time-constant of the discharge process. In severe winter conditions the thermal port of this capacitor will have to be made part of the system, etc.

Note that, after a particular choice of the separation between environment and system, the influence of the environment on the system may be concentrated in this system boundary by means of so-called sources and sinks, also called boundary conditions or constraints, depending on the domain background. They are part of the ideal conceptual elements to be discussed next.

13. Elementary behaviours and basic concepts

This section introduces the conceptual elementary behaviours that can be distinguished in the common description of the behaviour of physical systems, in particular from a port-based point of view. Before the individual elements are discussed, first the notation for the positive orientation in the form of the so-called *half-arrow* is introduced.

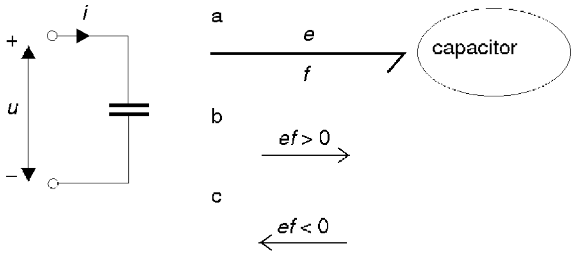


Fig. 50. Positive orientation of a bond represented by a half-arrow.

13.1. Positive orientation and half-arrow

Each bond represents a connection between two ports. However, if one end is kept open, it can be used to visualise the port it is connected to. The three variables involved, effort, flow and power, may have different signs with respect to this port. In order to be able to indicate this, a half-arrow is attached to the bond, expressing the positive orientation of these variables, similar to the plus and minus signs and the arrow that are used for an electric two-pole to represent the positive orientation of the voltage and the current, respectively (Fig. 50). Note that the half-arrow does not indicate the direction of the flow or the power: the direction can be opposite, in case the corresponding variable has a negative value.

13.2. Storage

The most elementary behaviour that needs to be present in a system in order to be dynamic is 'storage'. In mathematical terms, one can describe this behaviour by the integration of the rate of change of some conserved quantity, viz. the stored quantity or state, and by the relation with the equilibrium determining variable, the so-called constitutive relation. Note that in the common classification of domains, many domains are characterised by two types of state, viz. the generalised displacement and the generalised momentum, following the structure of the so-called mechanical domain. It has been noted before that another classification of domains that for instance separates the mechanical domain in a kinetic domain and a potential or elastic domain can easily resolve the paradoxical situation that results from the common choice (Breedveld, 1984), but this would be beyond the scope of this contribution. This means that the common two types of storage are used:

- the C-type storage element in which the flow is integrated into a generalised displacement and related to the conjugate effort;
- the I-type storage element in which the effort is integrated into a generalised momentum and related to the conjugate flow.

Note that both are dual in the sense that they can be transformed into each other by interchanging the roles of the conjugate variables effort and flow. Simple examples of C-type storage elements are:

- ideal spring (mechanical domain)
- ideal capacitor (electric domain)
- ideal reservoir (hydraulic/pneumatic domain)
- ideal heat capacitor (thermal domain)

Note that the use of the adjective ideal tries to emphasise that the difference between elements and components although the naming is usually based on the component that dominantly displays a particular elementary behaviour.

Simple examples of I-type storage elements are:

- ideal mass (mechanical domain)
- ideal inductor (electric domain)
- ideal fluid inertia (hydraulic/pneumatic domain)

Storage elements can be used in a domain-independent way due to the built in representation of the energy conservation principle. Not only the stored quantity, e.g. charge, matter, momentum, flux linkage, etc. is stored but also the energy related to this storage. In case more than one quantity is stored (multi-port storage) the principle of energy conservation supports the description of the potential power transfer from one domain into the other by means of cycle processes. Note that all other parts of a system have to satisfy the principle of energy conservation too. However, no storage takes place there, so it can be concluded at this point that all have to be power continuous in principle, apart from external sources and sinks that represent the interaction with the environment.

13.3. Irreversible transformation

Next to the first law of thermodynamics, the second law of thermodynamics has to be satisfied. However, the entropy production is assumed to take place only in the two-port irreversible transformers that are usually addressed as one-port 'dissipators' or 'resistors' due to the fact that the thermal port can be omitted if the temperature is assumed to be homogenous and constant. Note that this implicit assumption is often not explicitly mentioned, which may lead to modelling inconsistencies, as these one-ports are clearly power discontinuous.

As the rest of the system has to satisfy the second principle too, all entropy production is assumed zero there, which results in entropy continuity for all elements except for the storage elements where *reversible* storage of entropy is allowed. Note that reversible storage is a tautology, as irreversibilities would violate the basic concept of storage. The common acronym for an irreversible transducer is RS, derived from the common acronym in the isothermal case, R, to which an S for source is added to represent the entropy *production*.

Simple examples of irreversible transforming (resistive) elements are:

- ideal electric resistor
- ideal friction
- ideal fluid resistor
- ideal heat resistance

Due to the second principle of thermodynamics (positive entropy production), the relation between the conjugate variables at the R-port can be linear or nonlinear as long as the relation remains in the 1st and 3rd quadrant. However, the relation at the S-port (always in the thermal domain) is intrinsically nonlinear, due to the absolute zero-point of temperature.

13.4. Reversible transformation

Irreversible transformation more or less suggests the 'possibility' or rather the need for the ideal concept of a *reversible* transducer. As they cannot store or produce entropy, as these properties are already concentrated in other basic elements, they have to be power continuous. Their most elementary form is the two-port. It can be formally proven that, independent of the domain, only two types of port-asymmetric, i.e. with non-exchangeable ports, power-continuous two-ports can exist, at the one hand the so-called transformer (acronym: TF) that relates the efforts of both ports and also the flows of both ports and at the other hand the so-called gyrator (acronym: GY) that relates the flow of one port with the effort of the other vice versa. Furthermore, the nature of the relation is multiplicative, either by a constant (regular TF and GY) or by a time-dependent variable, the so-called modulating signal (acronyms: MTF and MGY). The notion of a port-asymmetric multiport will be clarified when port-symmetric multiports are discussed.

Simple examples of reversible transforming elements are:

- ideal (or perfect) electric transformer
- ideal lever
- ideal gear box
- ideal piston–cylinder combination
- ideal positive displacement pump

Simple examples of reversible gyrating elements are:

- ideal centrifugal pump
- ideal turbine
- ideal electric motor

An ideal continuously variable transmission is a simple example of a reversible, modulated transforming element, while an ideal turbine with adjustable blades is a simple example of a reversible, modulated gyrating element.

13.5. Supply and demand (sources and sinks/boundary conditions)

As already announced, the supply and demand from and to the environment can be concentrated in the (conceptual!) system boundary and represented by sources or sinks. As sinks may be considered negative sources, only ideal sources are used as ideal elements. Given that a port has two kinds of variables, effort and flow, two kinds of sources may exist, effort- and flow-sources (acronyms: Se and Sf). Generally speaking, all storage elements that are large compared to the dynamics of interest (note that this cannot be considered independently of the resistance of its connection to the rest of the system) may be approximated by infinitely large storage elements that are identical to sources. An infinitely large C-type storage element becomes an Se, an infinitely large I-type storage element becomes a Sf. However, feedback control may turn a port into a source, cf. a stabilized voltage source too. As the voltage may be adapted or modulated, this kind of sources are called modulated sources (MSe, MSf).

Simple examples are of (modulated) effort sources are:

- ideal (controlled) voltage source
- ideal (controlled) pressure source

Simple examples are of (modulated) flow sources are:

- ideal (controlled) current source
- ideal (controlled) fluid flow source

13.6. Distribution

In order to be able to distribute power between subsystems in an arbitrary way, at least three-ports are required. This is synonymous to the statement that, in order to be able to interconnect subsystems at will at least three-ports are required. By assigning all energy storage to the storage elements, all entropy production to the irreversible transducers (‘dissipators’) and all exchange with the environment to the sources, only power continuity remains. Furthermore, the requirement that port should be connectable at will, requires that an interchange of ports of these interconnection elements has no influence. This is the so-called property of port-symmetry. It is important to note that it can be formally proven that only the requirements of power continuity and port symmetry result in two solutions, i.e. two types of multiports (i.e. interconnection elements with two or more ports) with linear constitutive relations, the so-called junctions. The constitutive relations (one per port) of the first type require all efforts to be identical and the flows to sum up zero with the choice of sign related to their positive orientation, similar to a Kirchhoff current law. Paynter called this junction a 0-junction, due to the similarity between the symbol for zero and the shape of a node, because like a node in an electric circuit (at that time the only network type notation) the 0-junction has a common effort and the adjacent flows sum to zero.

The constitutive relations of the second type are dual: all flows should be identical and the efforts sum to zero with the choice of sign related to their positive orientation, similar to a Kirchhoff voltage law.

However, the mistake to say that the junctions represent the generalised Kirchhoff laws is wrong, as the junctions at the same time represent the ‘commonness’ of the conjugate variable, such that they can be used at the same time to represent that variable.

As mentioned before, really manipulating the concept of the junction in a way that supports the modelling process, i.e. without using other modelling techniques and translation first, requires some skill as the true understanding of the junctions requires the paradigm shift mentioned earlier. Nevertheless, the results are powerful, as will be demonstrated after the discussion of the causal port properties.

14. Causal port properties

Each of the nine basic elements (C, I, R(S), TF, GY, Se, Sf, 0, 1) introduced above has its own causal port properties, that can be categorised as follows: fixed causality, preferred causality, arbitrary causality and causal constraints. The meaning of these categories will become clear when the basic elements are discussed. For reasons of clarity, the sources are discussed first, after introduction of the notation by means of the so-called *causal stroke* (Karnopp & Rosenberg, 1975).

14.1. Causal stroke

Like the half-arrow the causal stroke is an additional label to the bond, but they do not influence each other. The causal stroke merely fixates the direction of the individual signal flows in the bilateral signal flow pair. The causal stroke is attached to that end of the bond where the effort signal comes out, i.e. where it enters the connected port. This automatically means that the so-called open end of the bond represents the computational direction of the flow signal (cf. Fig. 43).

14.2. Fixed causality

It needs no explanation that a source of effort always has an effort as output signal, in other words, the causal stroke is attached to the end of the bond that is connected to the rest of the system (Figs. 51 and 52a). Mutatis mutandis the causal stroke of the flow source is connected at the end of the bond connected to the source (Fig. 52b). These causalities are called ‘fixed causalities’ accordingly. Apart from these fundamentally fixed causalities, all ports of elements to may become nonlinear and non-invertible, i.e. all but the

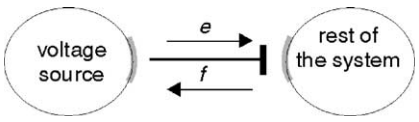


Fig. 51. Fixed effort-out causality of an effort (voltage) source.

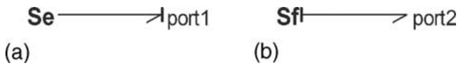


Fig. 52. Fixed causality of sources as assigned by 20-sim.

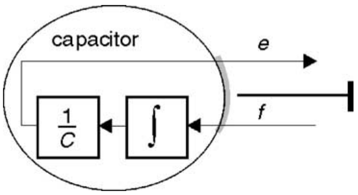


Fig. 53. Preferred integral causality of a capacitor or condenser.

junctions, may become fixed due to the fact that the constitutive relation may only take one form. In more advanced causal analysis procedures, the distinction between these two types of fixed causalities is used. Herein, this distinction will not be made for the sake of clarity.

14.3. Preferred causality

A less strict causal port property is that one of the two possibilities is, for some reason, preferred over the other. Commonly this kind of property is assigned to storage ports, as the two forms of the constitutive relation of a storage port require either differentiation with respect to time or integration with respect to time (Fig. 53). Only on the basis of numerical arguments the integral form is preferred, due to the fact that numerical differentiation amplifies numerical noise, but there is more. A first indication is found in the fact that the integral form allows the use of an initial condition, while the differential form does not. Obviously, an initial state or content of some storage element is a physically relevant property that illustrates the statement that integration 'exists' in nature, whereas differentiation does not. Although one should be careful with the concept existence when discussing modelling, this statement seeks to emphasise that differentiation with respect to time requires information about future states in principle, whereas integration with respect to time does not. The discussion of causal analysis will make clear that violation of a preferred causality gives important feedback to the modeller about his modelling decisions. Note that some forms of analysis require that the differential form is preferred, but this is never used as a preparation for numerical simulation.

14.4. Arbitrary causality

The expected next possibility in the sequence is that the causality of a port is neither fixed nor preferred. Hence, it can only be arbitrary. Examples of arbitrary port causality are linear, thus invertible, resistive ports. For example, the acausal form of the constitutive relation of an ohmic resistor is $u - Ri = 0$, the effort-out causal form is $u = Ri$, while the flow-out causal form is $i = u/R$ (Fig. 54).

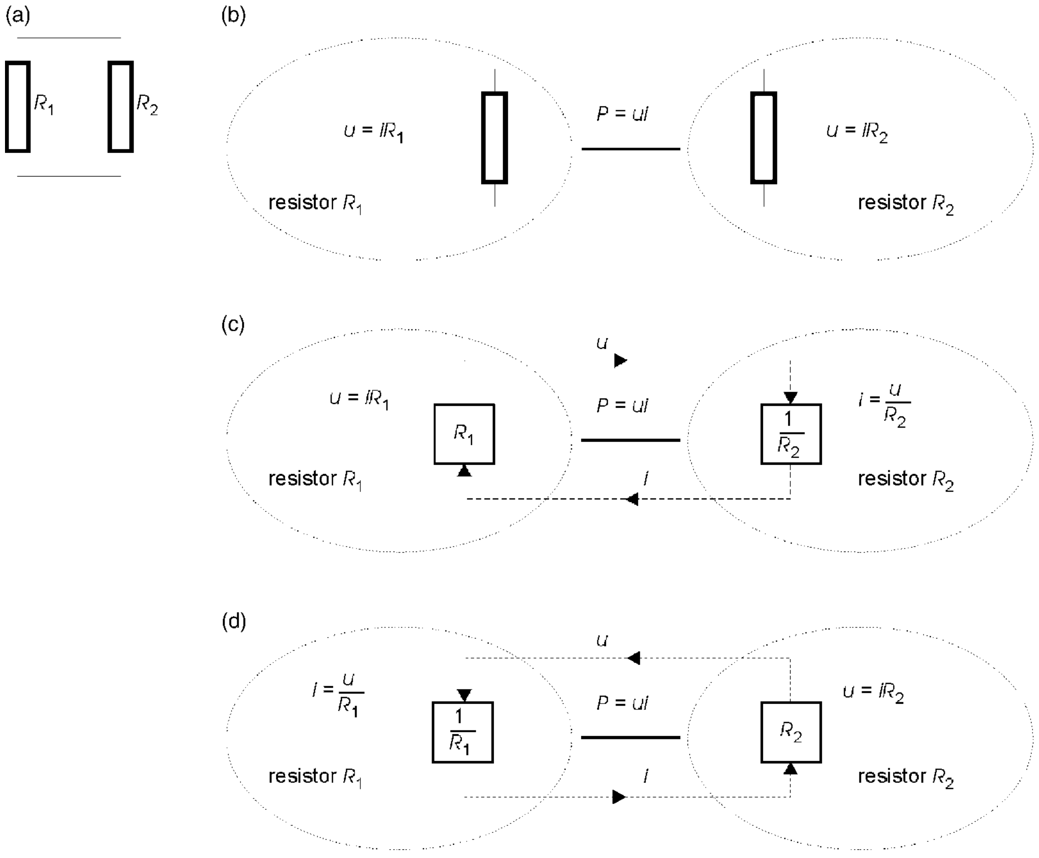


Fig. 54. Arbitrary causality of two resistors causing an algebraic loop.

14.5. Causal constraints

Causal constraints only exists for multiports, i.e. elements with two or more ports like the transducers (TF, GY) and the junctions (0, 1). For instance, if the constitutive relation of the two-port transducers is linear (junctions are always linear), the first port to which causality is assigned is arbitrary, but the causality of the second port is immediately fixed. For instance, the two-port transformer always has one port with effort-out causality and one with flow-out causality. By contrast, the causalities of the ports of a two-port gyrator always have the same type of causality. In graphical terms: a TF has only one causal stroke directed to it, while a GY has either both causal strokes directed to it or none.

The fundamental feature of the junctions that either all efforts are common (0-junction) or all flows are common (1-junction) shows that only one port of a 0-junction can have 'effort-in causality', i.e. flow-out causality, the result of the flow-balance. By contrast, only one port of a 1-junction can have 'flow-in causality', i.e. effort-out causality, the result of the effortbalance. In graphical terms: only one causal stroke can be directed towards a 0-junction, while only one open end can be directed towards a 1-junction.

15. Causal analysis: feedback on modelling decisions

Causal analysis, also called causality assignment or causal augmentation, is the *algorithmic* process of putting the causal strokes at the bonds on the basis of the causal port properties induced by the nature of the constitutive relations.

15.1. Fixed causality

Obviously, the first step in this process is to assign fixed causalities and immediately propagate them via the *causal constraints*. For instance, if a flow source is connected to a 1-junction, the source-port immediately gets flow-out causality, i.e. the 1-junction gets flow-in causality, which means that all other ports of the 1-junction get flow-out causality, etc. (Fig. 55). Conflicts at this stage of the causality assignment procedure indicate that the problem is ill posed, e.g. two voltage sources in parallel or two force sources trying to impose the same force (mechanically 'in series'). Note that the causality propagation may lead to violation of preferred causalities, e.g. a voltage source in parallel to a capacitor or a velocity source on a mass. This violation gives the modeller the feedback that no independent state is related to the storage element as its content is imposed by a source, which also means that it is dynamically inactive.

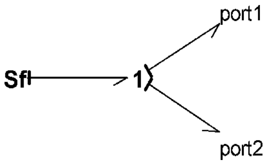


Fig. 55. Propagation of a fixed causality via a 1-junction.

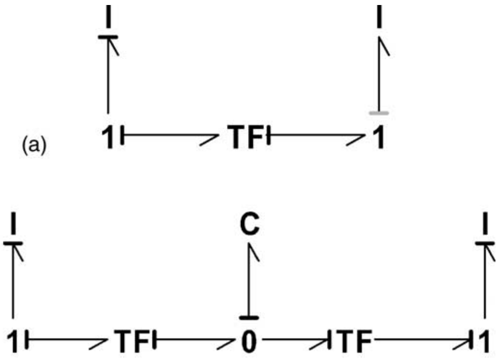


Fig. 56. (a) Dependent inertia's via the causal constraints of 1-junctions and transformer (cf. Fig. 14). (b) Independent inertia's by adding the elasticity of the transmission (cf. Fig. 15).

15.2. Preferred causality

Naturally, the fixed causalities are followed by the preferred causalities that are also propagated via the causal constraints. Conflicts at this stage indicate that a port may get differential causality as a result of another port getting preferred integral causality. Fig. 56 shows the bond graph of the two inertia's in the servo system, including the transmission, but without flexibility. This shows the modeller that he has chosen a model in which two storage ports depend on each other and form a signal loop with an integration that is compensated by a differentiation, i.e. a net algebraic loop, e.g. the two rigidly connected rigid bodies mentioned earlier. The computational problem may be solved by the application of implicit integration, by changing the model (the sequence of putting the causal strokes hints the modeller where a model change should be made, e.g. adding a spring between the two rigid bodies), or by symbolic manipulation (either manually or automatically) of the model. A technique to deal with this problem my adding some advanced control schemes to the model is under investigation. This also changes the model, but not in a way that can be physically interpreted.

15.3. Arbitrary causality

Commonly all ports are causal at this point, but if this is not the case, it means that at least two ports with arbitrary causality are present. If an arbitrary choice is made for one of the ports, this means that at least one other port will obtain its causality as a result of propagation via the causal constraints. The dual choice would have the same effect. This shows the modeller that this situation always results in an

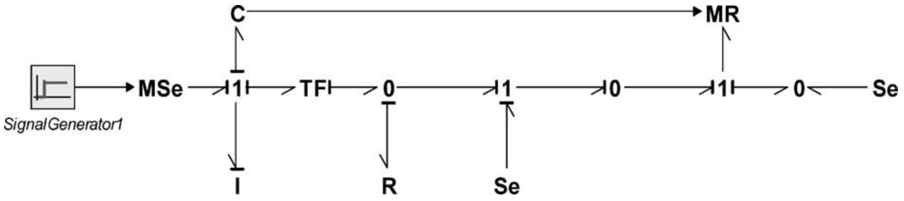


Fig. 57. Exploded view of Fig. 47 demonstrating the need for model changes in order to create a third-order loop via modulation.

algebraic loop that may cause numerical difficulties. In the same way as in case of differential causality, the assignment procedure itself hints the modeller how to change the model in order to prevent the loop.

In the example of the low-vacuum control valve, the omission of the air compressibility of the valve chamber results in an algebraic loop between the position-modulated resistor and the resistor representing the connection between chamber and environment. The assignment process hints the modeller to put a C-type storage element at the 0-junction representing the pressure in the chamber in order to prevent this algebraic loop (Fig. 54).

Note that the causality assignment process is completely algorithmic and more advanced variations on this algorithm exists and are implemented that can handle all possible situations. As a result, it can be used without using the notation itself, e.g. by replacing the bond graph by the more common iconic diagram representation. The amount of feedback that can be given to the modeller about his modelling decisions is largely reduced and the effect of model modifications becomes less obvious. But if one is merely interested in converting a simple iconic diagram into code ready for simulation, this is a powerful option. It will be exploited in the second part as the scope of this brief only allows some intuitive exposure to the bond graph notation.

15.4. Causal analysis of the control valve example

In the control valve example, Fig. 49 already showed an exploded view with bond graph causality. After the discussion of casualty in this section, it should be clear that a bond graph without modulating signals can never result in three integrations in a loop. A causal path can only exist between at most two storage elements, such that the number of integrations in the corresponding signal loop is at most two. Hence, the modulating signal of the valve that contains a third integration is crucial to create a model that is competent to represent the instabilities. Note that the causal strokes also provide the modeller information about the necessity of the compressibility of the air: if it is omitted, the third-order loop does not occur. Fig. 57 shows a full exploded view of Fig. 47 resulting in only bond graph elements. Now an algebraic loop occurs between the MR and the R and no causal path between the MR and the two storage elements (C, I) exists.

16. Conclusion

In this professional brief, it has been shown how modern object-oriented modelling of physical systems can help to make proper decisions during the design of mechatronic systems. In the cases it has been demonstrated that this approach enables to easily change from finding solutions in the controller domain or in the mechanical structure itself. Software tools that cover the different domains support this process.

This second part emphasised the background of physical modelling in general. It discussed in particular the paradigm shift to the port-based approach via the introduction of the concepts of a *port* and a *junction*. An example demonstrated that one of the major achievements is that a multiple-view approach is supported that provides insight into the nature and background of observed behaviour. A bond graph representation gives the user who gained some expertise in this language feedback about his modelling decisions via the representation of computational causality by the causal stroke. Note that the port-based computational causality assignment procedure is also used in the iconic diagram notation, but cannot provide the graphical insight that is obtained via a bond graph, as the control valve example illustrated. All sorts of generalisations like multi-bond graphs, Generalised Bond Graphs, bi-causality, hybrid systems, etc. exist, but are beyond the scope of this brief. The interested reader is referred to the extensive literature on these topics.

This text could only provide some background and a flavour of the design process by means of a few examples. In order to do some exercises yourself, the reader is encouraged to download the software from the Internet as well as the ‘hands on experience’ document that accompanies this professional brief.

The software is available at: http://www.20sim.com.

The ‘hands on experience’ document at: http://www.ce.utwente.nl/IFACbrief.

References

Ascher, U. M., & Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, ISBN 0-89871-412-5.

Breedveld, P.C. (1982). Thermodynamic Bond Graphs and the problem of thermal inertance. *Journal of The Franklin Institute, 314*(1), 15–40.

Breedveld, P. C. (1984). *Physical systems theory in terms of bond graphs*. ISBN 90-9000599-4 (distributed by author).

Breedveld, P. C. (1989). Fundamentals of bond graphs. In *IMACS annals of computing and applied mathematics, Vol. 3: Modelling and simulation of systems* (pp. 7–14). Basel: Blackwell.

Broenink, J. F. (1990). *Computer-aided physical-systems modeling and simulation: A bondgraph approach*. Ph.D. thesis, University of Twente, Enschede, NL.

Breunese, A. P. J. (1996). *Automated support in mechatronic systems modelling*. University of Twente, Enschede, NL, ISBN 90-365-0872-X.

Cellier, F. E., Elmqvist, H., & Otter, M. (1996). In W. S. Levine (Ed.). *Modelling from physical principles in the control handbook* (pp. 99–108). Boca Raton: CRC Press.

Coelingh, H. J. (2000). *Design support for motion control systems*. Ph.D. thesis, University of Twente, Enschede, NL, also http://www.ce.el.utwente.nl/clh/.

Coelingh, H. J., de Vries, T. J. A., & van Amerongen, J. (2000). Design support for motion control systems—Application to the Philips fast component mounter. In *Proceedings of the Mechatronics Forum 7th International Conference, Mechatronics 2000*. Atlanta, GA.

Controllab Products. (2003). 20-sim, http://www.20sim.com.

de Vries, T. J. A. (1994). *Conceptual design of controlled electro-mechanical systems*. Ph.D. thesis, University of Twente, Enschede, NL.

Dynasim. (2003). *Dymola*, http://www.dynasim.se/.

Gawthrop, P., & Lorcan Smith, L. (1996). *Metamodelling: Bond graphs and dynamic systems*. New York: Prentice-Hall.

Groenhuis, H. (1991). *A design tool for electromechanical servo systems*. Ph.D. thesis, University of Twente, Enschede, NL.

Karnopp, D. C., & Rosenberg, R. C. (1975). *System dynamics: A unified approach*. New York: Wiley.

Kuhn, T. S. (1962). *The structure of scientific revolutions*. Chicago: University of Chicago Press, http://www.emory.edu/EDUCATION/mfp/Kuhnsnap.html.

Mathworks. (2001). *The Mathworks: Developers of Matlab and Simulink*, http://www.mathworks.com.

Paynter, H. M. (1961). *Analysis and design of engineering systems*. Reading, MA: MIT Press.

Shearer, J. L., Murphy, A. T., & Richardson, H. H. (1971). *Introduction to system dynamics*. Reading, MA: Addison-Wesley.

van Amerongen, J. (2000). Modelling, simulation and controller design for mechatronic systems with 20-sim 3.0. In *Proceedings of the 1st IFAC Conference on Mechatronic Systems* (pp. 831–836). Darmstadt, Germany.

van Amerongen, J., Coelingh, H. J., & de Vries, T. J. A. (2000–2002). *Computer support for mechatronic control system design, robotics and autonomous systems* (Vol. 30 (3), pp. 249–260, PII: SO921-8890(99)00090-1).

Weustink, P. B. T., de Vries, T. J. A., & Breedveld, P. C. (1998). Object oriented modelling and simulation of mechatronic systems with 20-sim 3.0. In J. Adolfson & J. Karlsén (Eds.). *Mechatronics 98*. Amsterdam: Elsevier.

Wheeler, H. A., & Dettinger, D. (1949). *Wheeler Monograph 9* (p. 7).

Job van Amerongen studied Electrical Engineering at Delft University of Technology where he obtained an M. Sc. degree in 1971 and a Ph. D. degree in 1982. From 1971 to 1973 he did his military service as an officer in the Royal Netherlands Navy. From 1973 to 1987 he was assistant and associate professor at the Control Laboratory of the Faculty of Electrical Engineering of Delft University of Technology, where he worked on applications of modern control theory, especially model reference adaptive control, in ship control systems and electrical power production systems.

Since 1987, he is professor in Control Engineering in the Faculty of Electrical Engineering at the University of Twente. His current research interests are applications of modern control theory, especially intelligent control, in mechatronic systems. As head of the control laboratory he is also involved in the research in modelling and simulation of dynamical systems and in embedded control systems. He is scientific director of the Drebbel Institute for Mechatronics, of the University of Twente. He is member of the IFAC Technical Committee on Mechatronics and the IFAC TC on Marine Systems. From 1994 to 1998, he was dean of the Faculty of Electrical Engineering. In the first half year of 1999, he was on leave at the University of Newcastle in Australia.

He is (co-) author of many papers on adaptive and intelligent control systems, mechatronics and automatic steering of ships, co-author of a book on adaptive control systems and author of three courses on systems and control of the Dutch Open University (http://www. ce. utwente. nl/amn).

Peter Breedveld is an associate professor with tenure at the University of Twente, The Netherlands, where he received a B. Sc. in 1976, an M. Sc. in 1979 and a Ph. D. in 1984. He has been a visiting professor at the University of Texas at Austin in 1985 and at the Massachusetts Institute of Technology in 1992–1993 teaching integrated physical system modelling and dynamic systems and control. He is or has been an industrial consultant. He initiated the development of the modelling and simulation tool that is now commercially available under the name 20-sim. In 1990, he received a Ford Research grant (the first in continental Europe outside of Germany) for his work in the area of physical system modelling and the design of computer aids for this purpose.

He is an associate editor of the '*Journal of the Franklin Institute*', *SCS* '*Simulation*' and '*Mathematical and Computer Modelling of Dynamical Systems*'. His scientific interests are: integrated modelling, control and design of physical systems; graphical model representations (bond graphs); generalised thermodynamics; computer-aided modelling, simulation, analysis and design; dynamics of spatial mechanisms; mechatronics; generalised networks; numerical methods; applied fluid mechanics; applied electromagnetism; qualitative physics; surface acoustic waves in piezo-electric sensors and actuators.