Chapter 2
Bond Graph Modelling Overview

2.1 Introduction

The bond graph physical modelling analogy provides a powerful approach to modelling engineering systems in which the power exchange mechanism is important, as is the case in mechatronics. In this chapter we give an overview of the bond graph modelling technique. The intention is not to cover bond graph theory in detail, for there are many good references that do this well, e.g. [1–3]. The purpose is to introduce the reader to the basic concepts and methods that will be used to develop a general, systematic, object-oriented modelling approach in Chap. 3.

2.2 Word Models

Many engineering systems consist of components, e.g. electric motors, gears, shafts, transistors etc. (Fig. 1.3). Simulation models of such components can be represented as objects in the computer memory and depicted on the screen by their *word model*, i.e. a word description chosen to describe the component (Fig. 2.1).

The component name is useful for reference to the model. But, what is more important, the word model represents also how the component is connected to other components. When we look at a component, the internals of its design are usually hidden (e.g. by its housing). What are seen are the locations where it is connected to other components. These places—such as are electrical terminals, output shafts, fixing places, hydraulic ports, and boundary surfaces across which heat transfer takes place—are termed ports. In Fig. 2.1 the ports are shown by short lines.

When the component is connected and the system is energized from a suitable power source, there is a flow of power through these ports. Also, some ports serve to monitor or control the component. Thus, the ports serve as places where power or information exchange takes place. This is explained in Sect. 2.3.

V. Damić and J. Montgomery, *Mechatronics by Bond Graphs*,
DOI 10.1007/978-3-662-49004-4_2

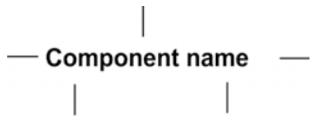


Fig. 2.1 Component word model

A component represented by its word description (its "name") and its ports is taken as the most fundamental representation of a component model and is termed the *word model*. The word model is used as the starting point of component model development.

2.3 Ports, Bonds, and Power Variables

Ports, as noted in Sect. 2.2, are places where interactions between components take place. These interactions can be looked on as power or information transfer. Thus, two types of ports are defined.

Ports characterised by power flow into or out of a component are termed *power ports*. Such ports are depicted by a *half arrow* (Fig. 2.2). The half arrow pointing to the component describes *power inflow*. It is assumed that at such a port there is positive power transfer into the component. Similarly, a half arrow pointing away from the component depicts *power outflow* from the component and the corresponding power transfer is then taken as negative.

Another type of port is characterised by negligible power transfer, but high information content. These are termed *control ports* and are depicted by a *full arrow*. The arrow pointing to the component denotes transfer of information into the component (*control input*). Similarly the port arrow pointing away from the component denotes information extracted from the component (*control output*).

The word model, i.e. the component represented by the name and the ports, is taken as the lowest level of component abstraction (Fig. 2.2). Components interact with other components through their ports. These interactions are looked on as power or information transfer between components and are depicted by lines connecting corresponding component ports (Fig. 2.3a). The lines that connect power ports are termed *bond lines*, or *bonds* for short. A bond line joins a *power*

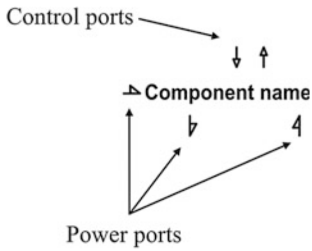


Fig. 2.2 Component word model with the ports defined

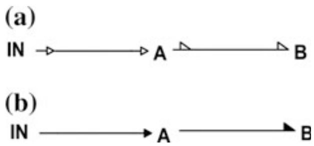


Fig. 2.3 Connecting components: **a** connecting ports by bond lines, **b** line and connected ports represented by bond line only

outflow port of one component and a *power inflow* port of the other and clearly shows the assumed direction of power transfer between components. Similarly, lines connecting control *output ports* and control *input ports* are termed *active* or *control* bonds. These lines show the direction of information transfer between components. When a bond line is drawn, ports and connecting lines appear as a single line with a half or full arrow at one of its ends (Fig. 2.3b). In the bond graph literature emphasis is put on the bonds, with ports playing a minor role. In our approach just the opposite point of view is taken: Ports, the places where inter-component actions take place, receive the emphasis.

Power or information exchange between component ports can be quite complex. It generally depends on the processes taking place in the components. In the simplest case the process in the component as seen at a power port can be described by a pair of *power variables*, the *effort* and *flow* variables. Their product is the power through the port (Sect. 1.3). Connecting such ports by a bond simply implies that effort and flow variables of interconnected ports are equal. Similarly, information at component control ports can be described by a single *control variable* (signal). Connecting an output port of a component to an input port of the other just means that these two control variables are equal.

In general, the situation is not this simple. Thus, the revolute joint illustrated schematically in Fig. 2.4a may be used to connect robot links or, a door in a door-frame. The joint can be represented by a word model (Fig. 2.4b), with ports representing the parts of the joint provided for the connection.

The function of the joint is to enable rotation of the connected bodies about the joint axis. To describe the interactions at the joint connection properly, pairs of effort and flow *vectors* are used. The effort vector can be represented by three rectangular components of the forces and torques, and likewise the flow vector by the rectangular components of linear and angular velocities. The meaning of these

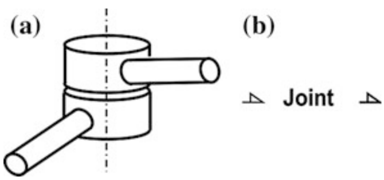


Fig. 2.4 Revolute joint: **a** scheme of joint, **b** word model representation

variables can be explained by defining a detailed model of the joint and the bodies in question. Hence the connection of a body to the joint can be represented by a bond, which denotes again that the efforts and flows of connected parts are equal. This time the power variables are not simple one-dimensional variables, but vector quantities.

Complex interactions at the ports can also be represented using multidimensional bond notation known as *multibonds* [4, 5]. We do not use this approach here, but instead treat the component ports as *compounded*. This means that the component ports are not simply objects, but define the structure of the mathematical quantities that describe the processes taking place inside the component. The bond lines simply define which port is connected to which, and hence which mathematical quantities should be equal. To define the structure of the ports the component model is developed in more detail.

2.4 Component Model Development

The detailed model of a component represented by the word model (Fig. 2.5, top-left) can be described in a document framed by a rectangle (Fig. 2.5, on the right). We call it a document because it will be represented on the computer screen in a document window and saved in a file (Chap. 3). The document title uses the name of the component that it models. The document contains ports represented by short strips placed just outside of the frame rectangle. These document ports correspond to the ports of the component: Every component port has a document port. The component `Comp A` in Fig. 2.5 has three ports: a power-in port, a power-out port, and a control-out port. Thus, there are exactly three document ports of the same type. The document ports are depicted in the positions around frame rectangle that corresponds to the position of the component ports around the component text (name). This way it is easy to see which port corresponds to which.

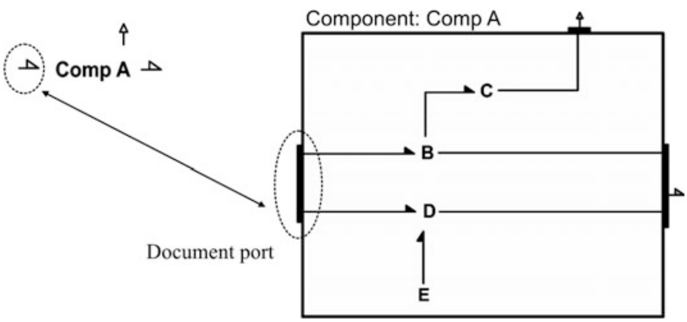


Fig. 2.5 Concept of component model

To develop the model, the component is analysed to identify the components of which it consists. Each is represented by its word model. Thus, in Fig. 2.5 there are four such components, named B, C, D, and E. Next we determine how these components are interconnected. Some are connected only internally, and this is represented by bonds connecting their respective ports, e.g. component B to component C. Some of the components are connected to the outside. In this case the respective component ports should be connected by the bonds to the document port strips, e.g. component B ports should be connected to the left and the right document ports.

Thus, document ports serve for the internal connection of the contained components. The document strip allows more than one bond to be connected to the port. We also assume that these bond connections are ordered, e.g. from top to bottom and from left to right. Hence, a component port (Fig. 2.5, left top) is represented by an array of internal component connections (Fig. 2.5 on the right).

When all the word models of the contained components are connected, we obtain a diagrammatical representation of the component model structure termed the *bond graph*. To complete the model it is necessary to continue developing the models of all contained components, which are represented by their word models, e.g. components B, C, etc. (Fig. 2.5).

The important question is how and when we end this process of systematic model decomposition. Formally, this happens when we get to a component that is fundamental, i.e. it doesn't contain simpler components. This is the problem of the level of abstraction we use when developing a model.

Normally we start model development from the *system level* (Sect. 1.2). At that level we define the model as a bond graph of the components. This is the lowest level of problem abstraction and component word models at this level usually correspond to the real-world components. In the next step we describe a model of the component by identifying the basic physical effects in the component, ignoring other, less important effects. Thus, the electrical resistor or mechanical spring can be described by an elementary model, such as Ohms law or the linear spring force-extension relationship. But we can also include the inductivity effect of the resistor or inertial effects in the spring. Hence, even in such simple cases we can use either simple or compounded component models. In other more complex devices such as robot arms, we identify real components that constitute such an arm, e.g. links, joints, base, etc. But even then we reach a stage at which we decide on the level of detail to be included in the underlying model. Physical processes are usually distributed over the component space, not restricted to small regions only. In such cases distributed models are usually discretized and can be represented by bond graphs of the components.

The bond graph modelling analogy enables the representation of models of basic physical processes taking place in engineering systems in the form of elementary components (Sect. 1.3). These components are described in more detail in Sect. 2.5. In addition, signal processing can also be described by several elementary operations (Sect. 2.6). Thus, starting from the system level, it is possible to develop the model gradually by applying the component decomposition technique. At every level of decomposition the components can be represented as elementary, or by a word model that is developed further. The resulting model thus can have one or more levels of decomposition. This depends on the system under study and the accepted level of abstraction of the problem being solved.

2.5 Modelling Basic Physical Processes

2.5.1 *Elementary Components*

The notion of elementary components has already been introduced in Sects. 1.2 and 1.3. These have a simple structure and serve as the building blocks of complex component models. In the bond graph method such components represent basic physical processes. Sometimes such components can be used as simplified representations of real components, such as bodies, springs, resistors, coils, or transformers.

There are, altogether, nine such components that represent underlying physical processes in a unique way. These are

- Inertial (`I`), Capacitive (`C`) and Resistive (`R`) components
- Sources of efforts (`SE`) and of flows (`SF`)
- Transformers (`TF`) and Gyrators (`GY`)
- Effort (`1`) and flow (`0`) junctions

The standard symbols used for the components are given in the parentheses.

In this way multi-domain physical processes, typical of mechatronics and other engineering systems, can be modelled in a unified and consistent way. A review of all the elementary components is given in Fig. 2.6. Components are described by their constitutive relations in terms of variables and physical parameters.

The components can have one or more power ports. The processes seen at these ports are described by pairs of power variables: *effort* *e* and *flow* *f*. In addition,

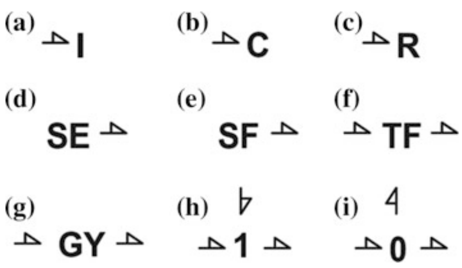


Fig. 2.6 Elementary components: **a** inertial, **b** capacitive, **c** resistive, **d** source effort, **e** source flow, **f** transformer, **g** gyrator, **h** effort junction, **i** flow junction

certain components have internal state variables. The next sub-sections give a detailed description of each component (Sects. 2.5.2–2.5.7). In Sect. 2.5.8 the controlled elementary components are described, i.e. common components with added control ports. At the control ports a control variable c is defined that is used for supplying information to, or extracting information from, the component.

2.5.2 The Inertial Components

The *inertial component* is identified by the symbol `I` and has at least one power port (Fig. 2.6a). This component is used to describe the inertia of a body in translation or rotation, or the inductivity of an electrical coil.

The port variables are effort e and flow f. In addition, there is an energy variable, *generalised momentum* p, defined by the relationship

$$e = \dot{p} \tag{2.1}$$

The generalised momentum can be viewed as the accumulation of effort in the component,

$$p = p_0 + \int_0^t e\, dt \tag{2.2}$$

The constitutive relation of the process reads

$$p = I \cdot f \tag{2.3}$$

where I is a parameter. The constitutive relation also can be non-linear, of the form

$$p = \Phi(f, par) \tag{2.4}$$

or, alternatively,

$$f = \Phi^{-1}(p, par) \tag{2.5}$$

where Φ is a suitable non-linear function and par denotes the parameters.

If the component has n ports, the constitutive relation at the ith port generally has the form

$$p_i = \Phi_i(f_j, par), \ (i, j = 1, \ldots, n) \tag{2.6}$$

or, alternatively,

$$f_i = \Phi_i^{-1}\left(p_j, par\right), (i,j = 1, \ldots, n) \tag{2.7}$$

where Φ_i are suitable multivariate functions.

A process represented by an inertial component is characterised by the accumulation of power flow into the component in form of energy

$$E = E_0 + \int_0^t e \cdot f\, dt$$

Using (2.1) we get

$$E(p) = E(p_0) + \int_0^t f\, dp \tag{2.8}$$

2.5.3 The Capacitive Components

The *capacitive* component is identified by the symbol C and has at least one power port (Fig. 2.6b). This component is used to model mechanical springs, electrical capacitors, and similar processes.

The port variables are effort e and flow f. In addition, there is an energy variable, *generalised displacement* q, defined by relation

$$f = \dot{q} \tag{2.9}$$

Thus, generalised displacement can be viewed as the accumulation of the *flow* in the component,

$$q = q_0 + \int_0^t f\, dt \tag{2.10}$$

The constitutive relation of the process reads

$$q = C \cdot e \tag{2.11}$$

where C is a parameter. The constitutive relation also can be nonlinear, i.e.

$$q = \Phi(e, par) \tag{2.12}$$

or, alternatively,

$$e = \Phi^{-1}(q, par) \tag{2.13}$$

where Φ is a suitable non-linear function and *par* denotes parameters.

If the component has n ports, the constitutive relation at the ith port generally is of the form

$$q_i = \Phi_i\left(e_j, par\right), \quad (i,j = 1, \ldots, n) \tag{2.14}$$

or, alternatively,

$$e_i = \Phi_i\left(q_j, par\right), \quad (i,j = 1, \ldots, n) \tag{2.15}$$

and Φ_i are suitable multivariate functions.

A process represented by a capacitive component is characterised by the accumulation of power flow into the component in form of energy

$$E = E_0 + \int_0^t e \cdot f \, dt$$

or by (2.9)

$$E(q) = E(q_0) + \int_0^t e \, dq \tag{2.16}$$

2.5.4 The Resistive Components

The *resistive component* is identified by the symbol R and, like the inertial and capacitive components, has at least one port (Fig. 2.6c). This component models friction in mechanical systems, or electrical resistors.

The port variables are effort e and flow f. The component constitutive relation is given by

$$e = R \cdot f \tag{2.17}$$

where R is a parameter. The constitutive relation can also be non-linear

$$e = \Phi(f, par) \tag{2.18}$$

or, alternatively,

$$f = \Phi^{-1}(e, par) \tag{2.19}$$

where Φ is a suitable non-linear function and *par* denotes parameters.

If the component has *n* ports, the constitutive relation at the *i*th port generally has the form

$$e_i = \Phi_i(f_j, par), \quad (i, j = 1, \ldots, n) \tag{2.20}$$

or,

$$f_i = \Phi_i^{-1}(e_j, par), \quad (i, j = 1, \ldots, n) \tag{2.21}$$

and Φ_i are suitable multivariate functions.

2.5.5 The Sources

Sources are components that represent power generation (or power sinks) such as voltage and current sources, certain types of forces (e.g. gravity), volume flow sources (such as pumps) etc. In these sources efforts or flows are almost independent of the other power variable. It is possible to define two types of source components: *source efforts*, designated by `SE`; and *source flows*, designated by `SF` (Fig. 2.6d, e). These are, basically, single port components. Denoting the port effort by *e* and port flow by *f*, the corresponding constitutive relations are given by the following relationships depending on the source type.

2.5.5.1 Source Efforts SE

$$e = E_0 \tag{2.22}$$

or, more generally,

$$e = \Phi(t, par) \tag{2.23}$$

2.5.5.2 Source Flows SF

$$f = F_0 \tag{2.24}$$

or, more generally,

$$f = \Phi(t, par) \tag{2.25}$$

In the relationships above, E_0, F_0, and *par* are suitable parameters, and Φ is a function of time t.

2.5.6 The Transformers and Gyrators

The *transformer* TF and the *gyrator* GY are two important components that represent transformations of the power variables between their ports (Fig. 2.6f, g). Both have two ports; power is directed into the component at one port, and out of the component at the other. Thus, power is assumed to flow through the component.

An important characteristic of these components is the conservation of power flow, i.e. power inflow is equal to power outflow. If we denote the corresponding power ports effort-flow variables by e_i and f_i (i = 0,1), this fact can be expressed by the relationship

$$e_0 f_0 = e_1 f_1 \tag{2.26}$$

2.5.6.1 Transformer TF

The transformer models the levers, gears, electrical transformers, and similar devices. In robotics and multi-body mechanics, transformers are extensively used for the transformation of power variables between body frames.

In the transformer there is a linear relationship between the same types of port variables, i.e. effort to effort and flow to flow. Denoting the transformation ratio by m, we have

$$\left.\begin{aligned} e_1 &= m \cdot e_0 \\ f_0 &= m \cdot f_1 \end{aligned}\right\} \tag{2.27}$$

These relationships satisfy the power conservation relationship given by (2.26). It is sufficient to define one of these relationships; the other follows due to the power conservation requirement. There is some ambiguity in how to define the transformation ratio because the power conservation relation is also satisfied by the inverse equations

$$\left.\begin{aligned} e_0 &= k \cdot e_1 \\ f_1 &= k \cdot f_0 \end{aligned}\right\} \tag{2.28}$$

The transformation ratio k in the last pair of the equations is just the reciprocal of ratio m in the former equations, i.e. $k = 1/m$. The form to use is left to the discretion of the modeller.

2.5.6.2 Gyrators GY

The gyrator is similar to the transformer, but relates the different types of ports variables, i.e. the efforts to flows. Denoting the gyrator ratios by m and k, the corresponding equations are

$$\left.\begin{aligned} e_0 &= m \cdot f_1 \\ e_1 &= m \cdot f_0 \end{aligned}\right\} \tag{2.29}$$

and alternatively,

$$\left.\begin{aligned} f_0 &= k \cdot e_1 \\ f_1 &= k \cdot e_0 \end{aligned}\right\} \tag{2.30}$$

The gyrators have their roots in the gyration effects well known from mechanics. Their use is essential in rigid-body dynamics. The gyrator is a more fundamental component than the transformer [1]. Two connected gyrators are equivalent to a transformer. A gyrator and an inertial component are equivalent to a capacitive element. Similarly, a source effort connected to a gyrator is equivalent to a source flow. Using such combinations makes it possible to reduce the set of elementary components necessary for physical modelling. We do not follow this approach here; there is little to be gained by using a smaller number of elementary components, as the resulting model would be more complicated and more abstract than necessary.

2.5.7 *The Effort and Flow Junctions*

Physical processes interact in such a way that there are restrictions on the possible values that efforts and flows can attain. Many physical laws express such constraints. In mechanics, forces and moments—including inertial effects—are governed by the momentum and the moment-of-momentum laws. In electricity, there is the Kirchhoff voltage law, and there are similar laws in other fields. Similar constraints on flows in rigid body mechanics are governed by the kinematical relative velocity laws, by the law of continuity of fluid flow in fluid mechanics, the Kirchhoff current law in electricity, etc. To satisfy such laws elementary components defined previously are connected to the junctions that impose constraints on efforts or flows. Such junctions are known as *effort* and *flow junctions* (Fig. 2.6h, i).

2.5.7.1 Effort Junctions

The *effort junction* is a multi-port component into which power flows in or out. The traditional symbol for this junction is `1`. This junction also is called a *common flow* junction because the flows at all junction ports are the same, i.e.

$$f_0 = f_1 = \cdots = f_{n-1} \tag{2.31}$$

where n is the number of ports at the junction. There is no power accumulation within the junction; thus the sum of the power inflows and outflows equals zero,

$$\pm e_0 f_0 \pm e_1 f_1 \cdots \pm e_{n-1} f_{n-1} = 0 \tag{2.32}$$

In this equation the plus sign is used for the ports pointing towards the junction (positive power) and the minus sign for ports pointing away from the junction (negative power). Using (2.31) we get equation of effort balance at the junction

$$\pm e_0 \pm e_1 \cdots \pm e_{n-1} = 0 \tag{2.33}$$

2.5.7.2 Flow Junctions

The *flow junction* is similar to the effort junction, with the roles of efforts and flows exchanged. The flow junction is a multi-port component traditionally denoted by the symbol `0`. This junction is also known as a *common effort* junction, as the efforts at all ports are the same, i.e.

$$e_0 = e_1 = \cdots = e_{n-1} \tag{2.34}$$

There also holds the conservation of power of flows through the junction (2.32). Thus, by (2.34) we get an equation of balance of flows at the junction

$$\pm f_0 \pm f_1 \cdots \pm f_{n-1} = 0 \tag{2.35}$$

2.5.8 Controlled Components

The component constitutive relations introduced so far depend on port and internal variables only (and *time* which is the global variable). In many instances it is also necessary to permit dependence on some external variables. This is the case when modelling controlled hydraulic restrictions in valves, variable resistors; capacitors, sources and other controlled components in electronics; and coordinate transformations in multi-body mechanics. For this purpose bond graphs use so called modulated components—modulated source efforts `MSE` and sources flows `MSF`, modulated transformers MTF and gyrators MGY. Some authors introduce other modulated components as well. We do not introduce such special components, but allow components to have control ports in addition to power ports.

Figure 2.7 shows components with added control ports. The most elementary components in Fig. 2.6 can have can have an input port. The only components that cannot have control input ports are effort and flow junctions. The components with control input ports are called *controlled* and their constitutive relations (see Sects. 2.5.2–2.5.6) depend also on the corresponding control variables. The transformers and gyrators must satisfy also the power conservation requirement. But this is not a problem because it is satisfied not only by constant transformer and gyrator ratios, but also by the ratios that dependent on a control variable c. Thus, e.g. the corresponding constitutive relations for controlled transformer and gyrators can have the same forms as given by (2.27)–(2.30), but with variable transformer and gyration ratios, e.g.

$$\left.\begin{aligned} e_1 &= m(c) \cdot e_0 \\ f_0 &= m(c) \cdot f_1 \end{aligned}\right\} \tag{2.36}$$

and,

$$\left.\begin{aligned} e_0 &= m(c) \cdot f_1 \\ e_1 &= m(c) \cdot f_0 \end{aligned}\right\} \tag{2.37}$$

respectively.

In addition we may define one specific component called the *switch*, denoted by Sw (Fig. 2.7b). This component has one power port and one control input port. The constitutive relation for the component is

$$\left.\begin{aligned} e &= 0, \quad c > 0 \\ f &= 0, \quad c \leq 0 \end{aligned}\right\} \tag{2.38}$$

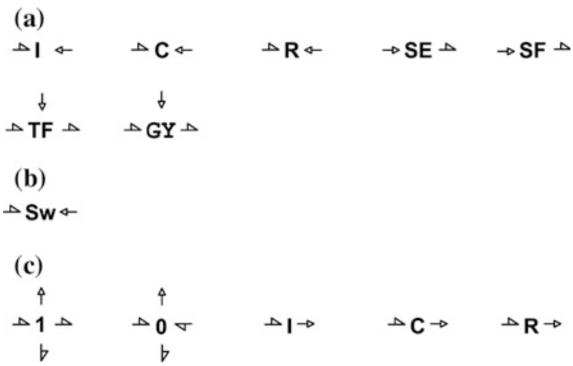


Fig. 2.7 Components with control ports: **a** inputs, **b** switch component, **c** outputs

where e and f are the power port effort and flow variables of, and c is the control variable. This component can be viewed as a controlled source that imposes zero effort or zero flow condition, depending on the sign of the control variable. This component models hard stops and clearances in machines, switches and relays in electronics, and possibly other discontinuous processes. The component can be generalised to allow effort or flow expressions, such as in sources (2.22)–(2.25) and in systems with more complex switching logic than in (2.38).

Finally, control output ports are used to access the component variables that cannot be accessed other way (Fig. 2.7c). Control output ports are commonly used for extraction of information on junction variables (efforts or flows). We also use such ports for access to the internal variables of inertial and capacitive components (momenta and displacements) and for extraction of information from other components, too.

2.6 Block Diagram Components

2.6.1 Introduction

Processes inside a system can be represented, either partially or completely, by signals (see Sect. 1.4). Thus, to complete the arsenal of components for modelling mechatronic systems, we define components that describe the input-output operations (Fig. 2.8). These are in effect the word model components of Fig. 2.2, which has only control ports. They may serve to define the basic block diagram operations in the system, e.g. they can be used to define control laws of mechatronic devices, the processing inside the system, or post processing of the simulation results.

The signals in a system can be broadly classified as *continuous-time* and *discrete-times* ones. The first type describes processes that are defined at every instant of time over some interval. These are commonly termed *analog* processes and usually represent different physical quantities, e.g. voltages, velocities, etc. The basic input-output components used to model continuous-time processes are described in Sect. 2.6.2.

In discrete-time processes the relevant quantities are defined only at discrete points in time. Such processes can be generated by sampling the continuous-time signal at the discrete times (Fig. 2.9). If the sampling frequency f_s is constant then the signal is *uniformly sampled* with sampled time index k given by $t_k = kT_s, (k = 1, 2, \ldots)$, where $T_s = 1/f_s$ is the *sampling interval*. This is typically the case in microprocessor-controlled systems where sampling is achieved by *analog to digital converters* (ADC). The basic components used to model discrete-time processes are discussed in Sect. 2.6.3.

Fig. 2.8 Block-diagram of a component

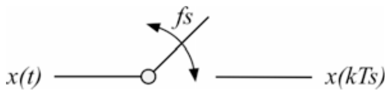


Fig. 2.9 Ideal sampling of a continuous-time signal

2.6.2 Continuous-Time Components

Some of fundamental input-output components are shown in Fig. 2.10. These components are well-known from control theory and will be reviewed briefly.

2.6.2.1 Input Components

The input components (Fig. 2.10a) generates control input action. These components can have only single control *output* port. This component generates the output in the general form

$$c_{out} = \Phi(t, par) \tag{2.39}$$

where Φ is a suitable function of time and the parameters.

These components typically are used to generate step inputs, sinusoidal inputs, pulse trains, and other input functions.

2.6.2.2 Output Components

The *output components* (Fig. 2.10b) display output signals. They can have one or more *input* ports. Typically such components are used for collecting signal for displaying as x-t and x-y plots. These components can be also represented by graphical symbols which resemble x-y plotters. Such a component is called `Display`.

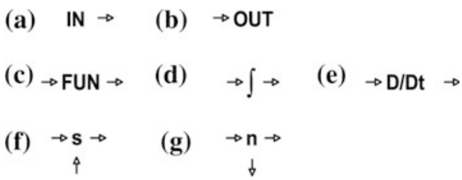


Fig. 2.10 Basic block diagram components: **a** input, **b** output, **c** function, **d** integrator, **e** differentiator, **f** summator, **g** node

2.6.2.3 Function Component

The *function components* (Fig. 2.10c) generates output as linear or non-linear functions of its inputs. The component can have one or more control input ports and single output port. The output generated by the function generally can have the form

$$c_{out} = \Phi(c_0, c_1, \ldots, c_{n-1}, par) \tag{2.40}$$

where c_0, c_1,…, c_{n-1} are the inputs, and *par* are the parameters.

Such a function can be used to represent linear gains, multiplications of the inputs, or other non-linear operation on the inputs. Often instead of generic symbol `FUN` more specific words can be used, which better describe the function, such as `k` for *k*-gains, `Limiter` for functions that limits the output, etc.

2.6.2.4 Integrator

As its name implies, this component evaluates the time integral of its input (Fig. 2.10d), i.e.

$$c_{out} = c_{out}(0) + \int_0^t c_{in}(t)dt \tag{2.41}$$

Obviously, this is a single input–single output component. Important parameter of the function is the initial value of the output.

2.6.2.5 Differentiator

In parallel with the integrator we introduce the *differentiator* (Fig. 2.10e), as component which generates the time derivative of its input, i.e.

$$c_{out} = \frac{dc_{in}}{dt} \tag{2.42}$$

However, it is not a proper input-output component, because time derivative of a time function is not well defined operation. This is reason why such an operation is not usually met among block diagram components. Here it is included because, as will be show later, the system model that we use are in the form of semi-implicit differential-algebraic equations (DAEs), which are solved using a method based on differentiation (so called *backward differentiation formula* BDF). Therefore, such a function is legitimate, but will be solved differently than the other input-output functions.

The differentiator can be used to model D component of PID controllers, and also if time-derivatives are explicitly needed.

2.6.2.6 Summator

The *summator* (Fig. 2.10f) gives the sum of its inputs, with optional positive or negative signs, i.e.

$$c_{out} = \pm c_0 \pm c_1 \cdots \pm c_{n-1} \tag{2.43}$$

At every input port there is associated a *plus* or *minus* sign, which indicates whether the corresponding input is added or subtracted when evaluating the output. Often instead of `s` (for summation), more common symbols such as Σ, ⊕ , or ⊗ are used.

2.6.2.7 Node

The *node* (Fig. 2.10g) serves for branching signals. This component has a single input and one or more outputs. Usually, instead of symbol `n` (for node), a large dot • is used.

2.6.3 Discrete-Time Components

Many continuous-time input-output components have their discrete-time counterparts. This is the case with the *Input*, *Output*, *Function*, *Summator* and *Node* components. However, there are components that are specific to discrete-time processes, in particular digital ones. These are *Analog to Digital Converters* (ADC), *Digital to Analog Converters* (DAC), *Clocks* and *Memory* (*delay*) components (Fig. 2.11).

The `A/D` component in Fig. 2.11a describes the quantization of the input signal, which can be described by the relationship.[1]

$$c_{out} = q \cdot round(c_{in}/q) \tag{2.44}$$

[1]This component corresponds to Quantizer block in Matlab-Simulink.

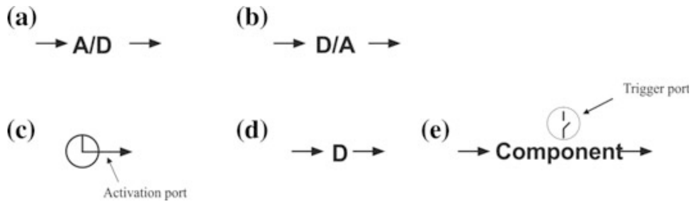


Fig. 2.11 Basic discrete components and ports: **a** analog to digital converter, **b** digital to analog converter, **c** clock with activation port, **d** delay, **e** triggered component

where c_{in} is the *analog input* signal and c_{out} is the *quantized output* signal, and q is the *quantization interval* (Fig. 2.12). Function *round* converts the ratio of input signal and the quantization interval to the lower integer value.

A *clock* component (Fig. 2.11c) can be used to synchronize the discrete operations in the system. It can be created in the form of a unit integrator, i.e.

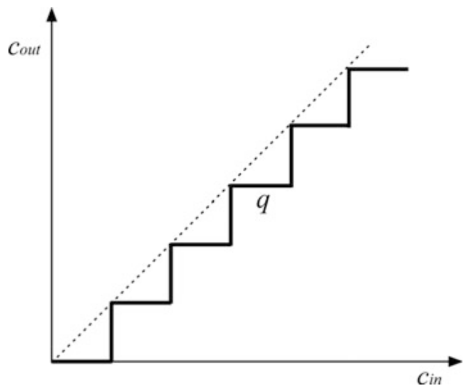
$$\int_0^t 1dt$$

which generates the outputs every sampling interval T_s (Sect. 2.6.3). Note that the clock has a special port called the *activation port* which activates the part of the system following the point of connection of the clock.

As an illustration of the application of the last two components consider the model of the A/D conversion in Fig. 2.13. The input to the A/D conversion is the analog signal, and the output is a digital number. If the range of the input is $c_{inH} - c_{inL}$ than the linear gain of M-bit converter is given by [6]

$$K = \frac{2^M - 1}{c_{inH} - c_{inL}} \tag{2.45}$$

Fig. 2.12 Quantization of the input signal



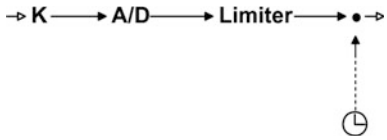


Fig. 2.13 Model of A/D converter

Thus, if the input range of 24 bit converter is ±2.5 V the linear gain is 3355443 1/V. This gain is represented in Fig. 2.13 by continuous-time function `K`. The `A/D` component converts the scaled output to a discrete value using a suitable quantization interval. Ideally, the quantization interval should be based on a number of possible binary values, i.e. $q = 1$ bit. However, due to noise such low resolution cannot be achieved. The minimum change in input voltage required to guarantee a change in the discrete output value is called the *least significant bit* (LSB). Thus in the above example if the LSB is 5 μV (ideally it is $1/K \approx 0.3$ μV) the corresponding quantization interval q = 5 μV • 3355443 $V^{-1} \approx 17$ bits. The `Limiter` limits the converter output to a maximum and minimum binary values that the converter can generate. Finally, there is a clock connected to the corresponding node. The clock defines the sampling rate of the converter and the starting and ending activation times.

The `D/A` component in Fig. 2.11b converts a discrete-time signal to the corresponding continuous-time one. There are various possible ways how to do this. The most popular way to generate the signal values over the next sampling interval is to hold the value of the signal at the constant value. Because the constant is zero-order polynomial this method is known as *zero-order hold*. Sometimes a linear interpolation is used (*first-order hold*), but it is generally accepted that zero-order hold is satisfying and is widely used. Note that there is a scaling involved between the digital input and analog output, and a similar gain function should be applied as in Fig. 2.13.

An important component that is often used in digital systems is the *delay* or `D` component (Fig. 2.11d). This component stores the current input value into a corresponding memory location. Because this value is available at the next sampling interval it is described by the relationship

$$\left.\begin{aligned} c_{out}[k] &= c_{in}[k-1], \quad k > 1 \\ c_{out}[1] &= c_{in}[0] = c_0 \end{aligned}\right\} \tag{2.46}$$

Therefore, the output of the component at the current sampling instance is equal to the value of the input at the previous instance. This function is, therefore, a *unit delay function*. In order to be defined at the first sample instance, i.e. when $k = 1$, it is necessary to define the initial value of the input, $c_{in}[0]$. The `D` function is analogous to the `integrator` in continuous-time processes (2.41).

The discrete components discussed so far are fundamental to modelling discrete (digital) processes. It is possible of course to define additional more specific components as well. We will conclude the discussion of discrete component by discussion of one special port introduced and shown in Fig. 2.11e. This is the *trigger port* and serves to activate or deactivate processing inside a discrete block to which it is added if an external signal connected to the port satisfies suitable conditions.[2] Such a component is termed a *triggered component*. Using an external signal it is possible to control when processes in a component are starting and ending. The conditions under which it happens are defined by the trigger port. Thus, it is possible to define that processes starts when a rising input signal goes through zero, or when a dropping signal crosses some value, or when some other condition is satisfied.

2.7 Modelling Simple Engineering Systems

The approach outlined in the previous sections can be used for the systematic computer-aided model development of engineering problems. We apply this approach to two simple, well-known problems, one from mechanical engineering and the other from electrical engineering. The technique is compared with the common bond graph modelling technique as given e.g. in [1]. We also consider a more complicated practical example from mechanical engineering (the See-saw problem).

2.7.1 Simple Body Spring Damper System

The first example models a single-degree-of-freedom mechanical vibration system (Fig. 2.14a). It consists of a body of mass m that translates along a floor, and is connected to a wall by a spring of stiffness k, and by a damper with a linear friction velocity constant b. An external force F acts on the body. We neglect the Coulomb friction between the block and the floor for simplicity, as well as the weight of the body. In Fig. 2.14b the system is decomposed into its basic components. This is the *free-body diagram* well known from engineering mechanics. This decomposition clarifies the power flow direction assignment of the component ports.

The bond graph model of the system is shown at the top of Fig. 2.15. The system consists of three components: `Spring`, `Damper`, and `Body` represented by the corresponding word models. The `Wall` (and floor) constitutes a component belonging to the system environment and is represented by the word model. The power ports of `Spring` and `Damper` and corresponding ports in `Wall` and `Body` are connected by the bond lines. The body is acted upon also by an external force, represented by a `SE` (Source Effort) elementary component, and is connected to the

[2]Its function is similar to Triggered Subsystems in Matlab-Simulink.

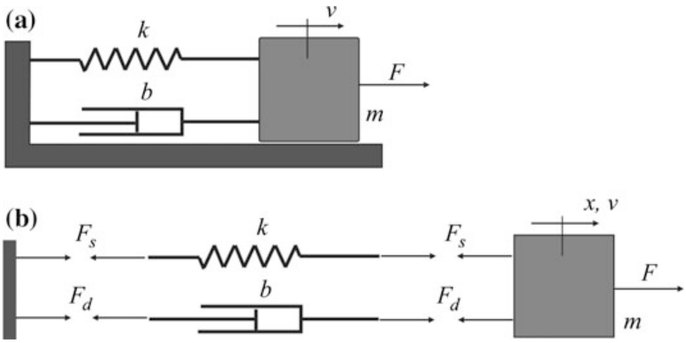
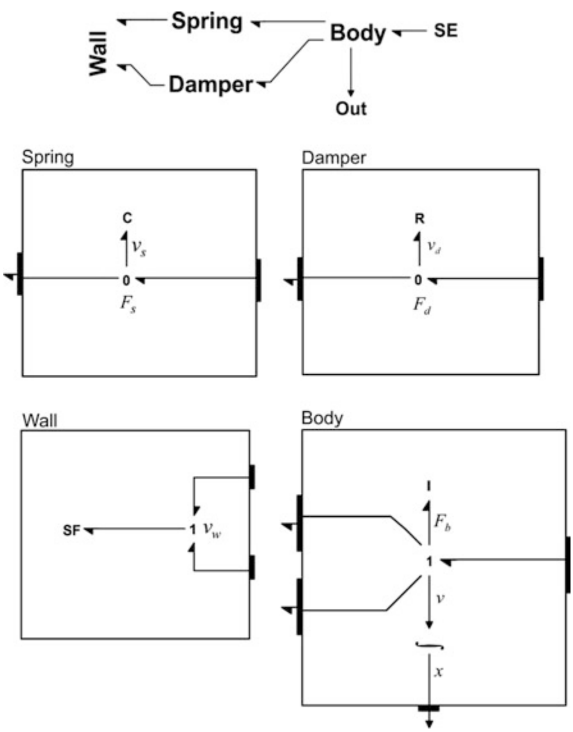


Fig. 2.14 Body spring damper system: **a** schematic representation, **b** free body diagram

Fig. 2.15 Bond graph model of problem in Fig. 2.14



corresponding `Body` port by a bond. There is also a control-out port on the `Body` which is used to extract information on the body position. This port is connected to the `Out`(put) port for display of body position during the simulation.

The model at this level of abstraction has a structure that closely corresponds to the scheme of the system in Fig. 2.14a. The direction of power flow in the model is taken from the `SE` through the body, then through the spring and damper, and finally to the wall.

This corresponds to the physical situation in Fig. 2.14b. If the sense of the external force and the body velocity are as shown, the power at the external force port is positive; i.e., it is directed *into* the body. Assuming that the spring and damper resist the movement of the body—i.e., the sense of their forces is opposite to that of the body velocity—the powers at the corresponding ports are negative, and the power port arrows are directed *out* of the body. At the spring and damper ports, power again is positive, flowing into these components. Because of the direction of the body's force, according to *Newton's Third Law* their forces act in the opposite sense. A similar conclusion can be drawn regarding the wall side ports. Hence, by joining mechanical ports Newton's Third law is satisfied. Thus, to construct the bond graph model it is not necessary to draw the free-body diagram at all.

Next we develop the component models (Fig. 2.15). The force generated by the spring depends on the relative displacement (extension) of the spring. Thus, the model of the spring can be represented by a flow junction with three ports, two for connecting internally to the spring end ports and the third for the connection of the capacitive element that models the elasticity of the spring (Fig. 2.15 `Spring`). The junction variable is the force F_s in the spring; and the extension of the spring x_s is the generalized displacement of the capacitive element, with spring stiffness k taken as the element parameter. Thus, the constitutive relations for the capacitive element are (*see* (2.9) and (2.11))

$$v_s = \dot{x}_s \tag{2.47}$$

and

$$F_s = k \cdot x_s \tag{2.48}$$

where v_s is the relative velocity of the spring ends.

The damper has a similar model, with the resistive element used to model mechanical dissipation in the damper (Fig. 2.15 `Damper`). The junction variable F_d represents the force developed by the damper, v_d is the relative velocity of the damper ends and the velocity constant b is a parameter of the element. Assuming a linear constitutive relation for the resistive element we have (see (2.17))

$$F_d = b \cdot v_d \tag{2.49}$$

The third component of the system is the body of mass m (Fig. 2.15 `Body`). This component uses an effort junction to describe the balance of forces applied to the body including the inertial force of the body. This junction has four power ports: three for internal connections to the body ports and fourth for the connection of an inertial element. Denoting the body velocity taken as the junction variable by v (see Fig. 2.14a) and the inertial force of the body of mass m taken as parameter by F_b, the constitutive relations of the inertial element are (see (2.1) and (2.3))

$$F_b = \dot{p}_b \tag{2.50}$$

and

$$p_b = m \cdot v \tag{2.51}$$

To calculate the body position, a control output port is added to the junction, and the junction variable is fed to an integrator that outputs the body position x. The corresponding equation can be written as

$$\dot{x} = v \tag{2.52}$$

The next component is simply the source effort element `SE`, which generates the driving force on the body

$$F = \Phi(t) \tag{2.53}$$

The spring and damper are connected to the fixed wall. The model of the wall is given in Fig. 2.15 `Wall`. The component uses an effort junction with three ports, which describes the force balance at the wall. Two ports serve for the internal connection to the wall ports, where the spring and damper are connected, and the third is for connecting to the source flow, which imposes a zero wall velocity condition. The junction velocity is v_w. Thus, the relation for the source flow is

$$v_w = 0 \tag{2.54}$$

To complete the mathematical model of the system the equations of the effort and flow junctions are added. Corresponding variables can be found by following the bonds connected to junction ports until some elementary component is found that completes the bond. Thus, for the body effort junction in Fig. 2.15 `Body`, the port effort variables are the spring force F_s, damper force F_d, inertia force F_b and driving force F, respectively. The equation of the effort balance at the junction thus reads

$$-F_s - F_d - F_b + F = 0 \tag{2.55}$$

If we denote by F_w the total force at the wall, the equation of effort balance at the wall junction reads (Fig. 2.15 `Wall`)

$$F_s + F_d - F_w = 0 \tag{2.56}$$

A similar equation can be written for the flow junctions. This time the summation is on the flows. Thus, we have (Fig. 2.15 `Spring`)

$$-v_w - v_s + v = 0 \tag{2.57}$$

and (Fig. 2.15 `Damper`)

$$-v_w - v_d + v = 0 \tag{2.58}$$

We, therefore, may describe the motion of the system by eight equations of elements that describe the physical processes in the system, i.e. (2.47)–(2.53), and four equations which involve the junctions (2.54)–(2.58). There are, altogether, twelve differential and algebraic equations that have to be satisfied by twelve variables: F_s, v_s, x_s, F_d, v_d, F_b, v, p_b, F, F_w, v_w and x. Although we have arrived at a relatively large number of equations of motion for this simple problem, the equations are very simple, having on average only 27/12 = 2.25 variables per equation.

The structure of the matrix of these equations is very sparse; this simplifies the solution process. We can simplify these equations further. Direct processing can be used to eliminate some, or all, of the algebraic variables (i.e., variables that are not differentiated). We can also simplify the bond graph first, and then write the corresponding equations. We consider the second approach in more detail, as it leads to the sort of bond graphs usually found in the literature.

We can simplify the model by substituting every component at the top of Fig. 2.15 by its corresponding model, given at the bottom part of the same figure. The resulting bond graph is shown in Fig. 2.16a.

The source flow on the left imposes zero wall velocity; thus, we can remove the effort junction and the source flow, as well as the two bonds connecting to the flow junctions. We also remove the corresponding ports at the junctions. This yields a bond graph represented by Fig. 2.16b. We should also eliminate these flow junctions, for they are trivial, having only one power input port and one power output port. Thus, the `C` and `R` element ports can be connected directly to the effort junction ports on the right. This results in the bond graph of Fig. 2.16c.

The model in Fig. 2.16c is much simpler than that in Fig. 2.15. The resulting equations now consist of

$$\left.\begin{aligned}
&v = \dot{x}_s \\
&F_s = k \cdot x_s \\
&F_d = b \cdot v \\
&F_b = \dot{p}_b \\
&p_b = m \cdot v \\
&F = \Phi(t) \\
&-F_s - F_d - F_b + F = 0 \\
&\dot{x} = v
\end{aligned}\right\} \tag{2.59}$$

We have reduced the system to eight equations with eight variables F_s, x_s, F_d, F_b, v, p_b, F, and x. This was achieved, however, by eliminating some variables that can be of interest, e.g. total force transmitted to the wall. This bond graph can be

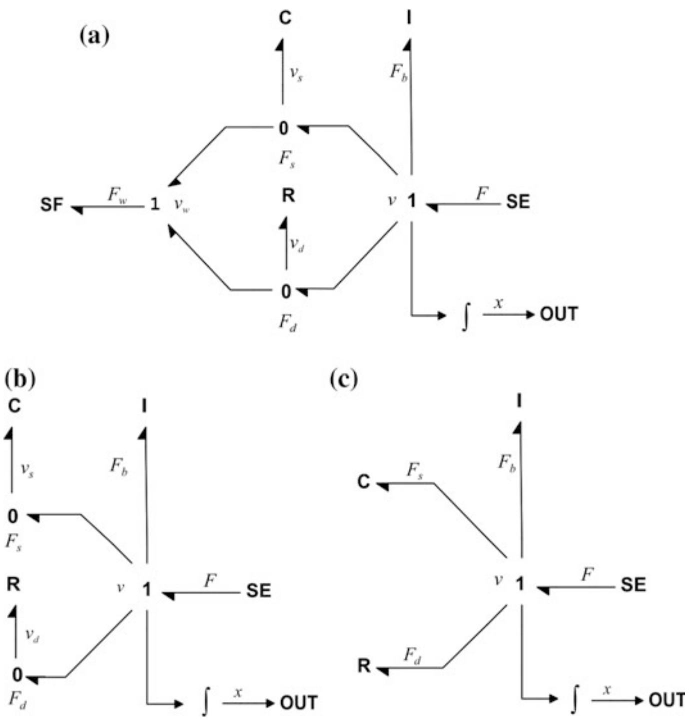


Fig. 2.16 Simplification of the bond graph of Fig. 2.15

developed directly from Fig. 2.16a by the application of classical methods of bond graph modelling, as explained in [1].

Using this form of bond graph model, the equations of motion of the system can be developed in an even simpler form than that given above. It should be noted, however, that this is not true in general for engineering systems of practical interest. We address this matter in more detail in Sects. 2.9 and 2.10. The reduced model is, on the other hand, much more abstract. This makes it more difficult to understand and interpret: Unlike the component model of Fig. 2.15, there is no topological similarity to the system represented in Fig. 2.14a. A change in any part of the model affects the complete model. On the other hand, in the model of Fig. 2.15 we can change some of the components, leaving the others unchanged. Such a model can be refined much more easily, thereby retaining the overall topological similarity to the physical model.

2.7.2 *The Simple Electrical Circuit*

The second example considers the electrical Resistor Inductor Capacitor circuit (RLC circuit) shown in Fig. 2.17. The circuit consists of a series connection of a

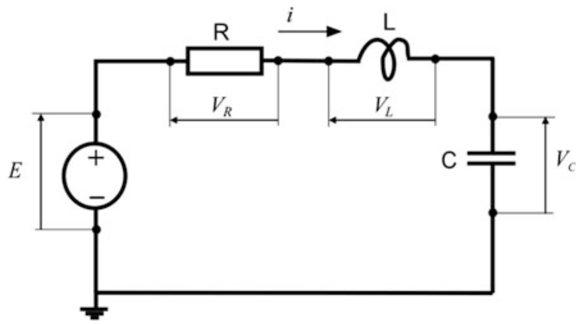


Fig. 2.17 Simple electrical circuit

voltage source generating an electromotive force (e.m.f.) `VS`, a resistor `R`, an inductor `L`, and a capacitor `C`. The polarities of the voltage drop across the electrical components are also shown, as well as the assumed direction of the current flow. We can develop a bond graph model using an approach similar to the mechanical system analysed previously.

We can represent the electrical components by suitable word models. But instead of the component names, we use the common electrical symbols. Standard component names—such as resistor, capacitor, and the like—may be retained internally for compatibility with the usual word model representation. The resulting bond graph is shown at the top of Fig. 2.18.

The source voltage supplies electrical power to other parts of the circuit. The power port corresponding to the positive terminal is taken to be directed outward, and the other port inward. Power from the source flows through the resistor, inductor, and capacitor until the node component is reached where a part of the power flow branches to the ground component. (Later it is shown that there is no power flow to the ground.) The other part returns back to the voltage source. The model has a very similar appearance to the electrical scheme in Fig. 2.17. What is different is the presence of power ports showing the assumed direction of power flow in the circuit. Thus, the correspondence of the bond graph model and the electrical scheme is really very close. Note that for the output component a component graphically resembling x-y plotter (`Display`) is used.

Component models for the voltage source `VS`, resistor `R`, inductor `L`, and the capacitor `C` are shown in the lower part of the Fig. 2.18. The components have two ports used for connecting to other components. Their models are represented by three ports effort junctions `1`. Two of these are used for internal connection to the component's ports, and the third is used for connection of the elementary components that describe the physical processes in the components. Power flow is chosen to flow into these elementary components. Thus, the effort of the elementary component port represents the voltage difference across the component. We model the components by the idealized linear elements.

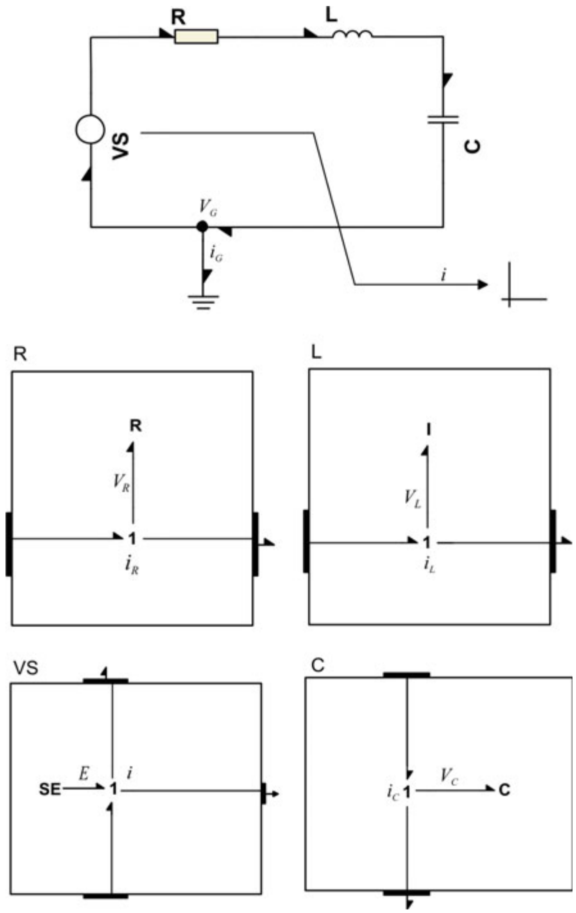


Fig. 2.18 Bond graph model of circuit of Fig. 2.17

In the case of the resistor, the junction variable is the current i_R flowing through the component, and the voltage drop is V_R. Thus, the constitutive relation for the resistor is given by (2.17)

$$V_R = R \cdot i_R \tag{2.60}$$

with R the resistance parameter.

Similarly for the inductor, the junction variable is the current i_L through the inductor and the voltage drop is V_L. If we denote the flux linkage of the coil by p_L, the constitutive relations read (see (2.1) and (2.3))

$$V_L = \dot{p}_L \tag{2.61}$$

and

$$p_L = L \cdot i_L \tag{2.62}$$

where L is the inductance parameter of the inductor.

For the voltage source the joint variable is the current i through the source terminals (ports). The voltage E generated is described by the source effort element. The constitutive relation reads

$$E = \Phi(t) \tag{2.63}$$

Finally, for the capacitor, the junction variable is the current i_C through the component and the voltage drop is V_C. If we denote the capacitor charge by q_C, the constitutive relations can be written as (see (2.9) and (2.11))

$$i_C = \dot{q}_C \tag{2.64}$$

and

$$V_C = q_C / C \tag{2.65}$$

where C is the capacitance.

The node component is simply another representation of the flow junction, and the ground is just the ground potential source effort. The node variable is the ground potential v_G. We take the ground potential as zero, hence the ground component constitutive relation reads

$$V_G = 0 \tag{2.66}$$

If we start from any of the component effort junctions and follow the bonds connected internally to the port, then out of the component to the next component port, and again into the component, we find that all effort junctions are interconnected. We, thus, can treat all these junctions as a single junction, the result being that all junction variables are, in essence, the same variable, i.e.

$$i \equiv i_R \equiv i_L \equiv i_C \tag{2.67}$$

Counting only ports connected to other components, the balance of efforts reads as follows

$$V_G + E - V_R - V_L - V_C - V_G = 0 \tag{2.68}$$

After cancellation of the ground potential v_G, we get

$$E - V_R - V_L - V_C = 0 \tag{2.69}$$

This is the Kirchhoff voltage law for the circuit.

Finally, if we denote the current drawn by the ground by i_G, the balance of flows at the node reads

$$-i - i_G + i = 0 \tag{2.70}$$

Again, due to the cancellation of current i, we obtain

$$i_G = 0 \tag{2.71}$$

This lengthy derivation shows that we arrive at the circuit equation by using the constitutive equations for elementary components and symbolically simplifying the junction equations. Thus, the mathematical model of the circuit consists of nine differential-algebraic equations (2.60)–(2.66), (2.69), and (2.71) with nine variables E, V_R, i, V_L, p_L, V_C, q_C, V_G and i_G. Equations (2.66) and (2.71) are trivial and could be eliminated from the system.

As in the previous example, we can simplify the bond graph instead of the equations. We first substitute the bond graph model of the components from the lower part of Fig. 2.18 into the system bond graph at the top. Also the node is changed to the flow junction and the ground to source effort. We thus obtain bond graph shown in Fig. 2.19a.

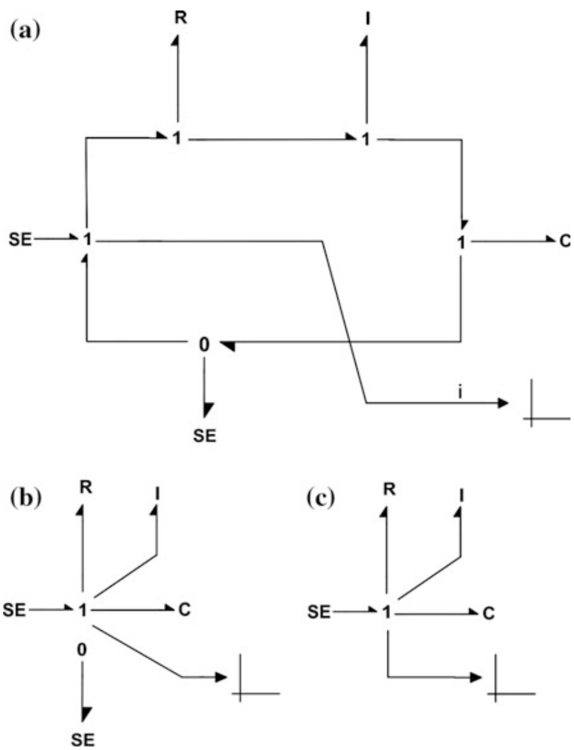


Fig. 2.19 Simplification of model of Fig. 2.18

We again see that the four effort junctions are interconnected and can be condensed into a single junction, retaining only the ports connected to other components. In addition, we see that this junction is connected to the same flow junction by two ports of opposite power-flow sense. Hence, such ports can be disconnected, and then removed. The corresponding ports of the flow junction must be removed, too. This yields the bond graph shown in Fig. 2.19b. We now have a flow junction connected only to a ground source effort. These can be removed, too. This results in the final simplified system bond graph (Fig. 2.19c). The last bond graph can be described by the same equations as before, but without (2.66) and (2.71).

The above procedure shows that, instead of the simplification of junction (2.68) and (2.70), we could directly arrive at (2.69) and (2.71) by noting that interconnected effort junctions are connected to the *same* flow junction. Corresponding junction ports then could be treated as internal, and not taken into account when writing the junction equations.

Comparing bond graphs in Figs. 2.18 and 2.19c, we draw similar conclusions as in the previous example: The model in Fig. 2.18 is much easier to interpret and upgrade. Even people who are not too familiar with bond graphs could understand such a model. On the other hand, it retains the advantages that bond graphs enjoy over other modelling methods.

2.7.3 A See-Saw Problem

As a third problem, we develop a model of a simple see-saw often found in children's playgrounds (Fig. 2.20). On a much larger scale, this same problem is known as the swing boat at the fair ground. The system consists of a platform that can rotate around a horizontal pin O, fixed in a frame, and having a body at each end, e.g. a boy and a girl sitting on the see-saw seats. If one of the bodies is pushed down, and then released, the system will begin to oscillate around its equilibrium position.

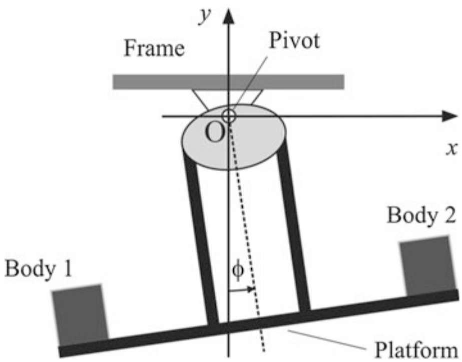


Fig. 2.20 See-saw problem

This problem is more complicated than those presented previously, as it consists of three interconnected bodies moving in the vertical plane. The model could be developed easily by treating the system as a physical pendulum. Instead, we consider it as a multi-body system and demonstrate how the model can be developed systematically by decomposition.

We start by defining the overall model structure (Fig. 2.21). The word models `Body 1` and `Body 2` represent the bodies on the platform. They each have a port that corresponds to the location where the body acts on the platform. The component `Platform` has four ports, two of which correspond to the places where the bodies act, a third for connecting to the `Pivot` component, and the last for the input of information on the rotation angle. The `Pivot` permits only rotation of the platform about a horizontal pin fixed in the `Frame`.

We assume that the bodies are firmly placed on the platform and move with it. Hence, we join the ports of the bodies to the corresponding platform ports by bonds. We assume the power flow sense from `Body 1` and `Body 2` through the `Platform` and `Pivot` to the `Frame`. Further, the information on the force at the pivot is of interest. Thus, we take the rectangular components of force F_x and F_y on the `Pivot` and feed them to the `Display`. Similarly, we extract information on the rotation angle *Phi* and feed it to the node. This information branches further to the `Platform` and to `Display.` Note, `Display` is output component in form of x-y plotter (see Sect. 2.6.2).

We proceed by developing models of the components. This requires defining the interactions taking place between them. Motion of the system is described in global co-ordinate frame *Oxy*, with origin *O* at the point of rotation of the platform in the vertical plane, axis *y* directed upward, and *x* to the right (Fig. 2.20).

The `Body1` and `Body2` models are shown in Fig. 2.22 `Body 1` and `Body 2`. Separate effort junctions are used for the summation of the *x* and *y* components of forces acting on the bodies. The junction variables are the *x* and *y* components of the velocities of the bodies. The junctions are connected internally to their respective ports. The order of connection going from the left to the right is the *x* component first, then the *y* component,. This order of connection is also used for the other ports. Hence, `Body1` and `Body2` port variables are pairs of effort flow vectors $\mathbf{F}_1,\mathbf{v}_1$ and $\mathbf{F}_2,\mathbf{v}_2$, respectively.

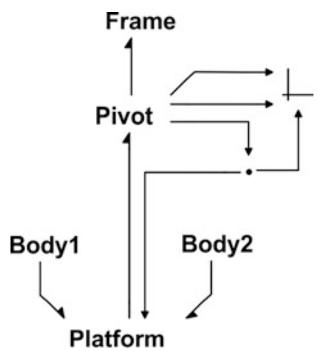


Fig. 2.21 Overall structure of the see-saw

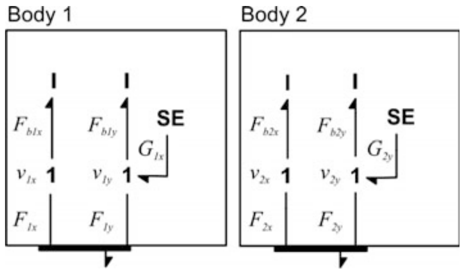


Fig. 2.22 Dynamics of bodies attached to the see-saw platform

The inertial effects of the bodies in the *x*- and *y*-directions are represented by the inertial elements **I** connected to the corresponding effort junctions, with power flow directed into the inertial elements. The weights of the bodies, acting in the *y* direction only, are represented by source efforts connected to the y component junctions, with power flow directed into the junctions. The equations of motion of the bodies can be obtained directly from the bond graphs of Fig. 2.22. The masses of the bodies are m_1 and m_2, and g is the gravitational acceleration. The relevant variables are also shown in the figure.

Body 1:

$$\left.\begin{aligned} \dot{p}_{b1x} &= F_{b1x} \\ \dot{p}_{b1y} &= F_{b1y} \\ p_{b1x} &= m_1 \cdot v_{1x} \\ p_{b1y} &= m_1 \cdot v_{1y} \\ G_{1y} &= -m_1 \cdot g \\ -F_{1x} - F_{b1x} &= 0 \\ -F_{1y} - F_{b1y} + G_{1y} &= 0 \end{aligned}\right\} \tag{2.72}$$

Body 2:

$$\left.\begin{aligned} \dot{p}_{b2x} &= F_{b2x} \\ \dot{p}_{b2y} &= F_{b2y} \\ p_{b2x} &= m_2 \cdot v_{2x} \\ p_{b2y} &= m_2 \cdot v_{2y} \\ G_{2y} &= -m_2 \cdot g \\ -F_{2x} - F_{b2x} &= 0 \\ -F_{2y} - F_{b2y} + G_{2y} &= 0 \end{aligned}\right\} \tag{2.73}$$

The Frame simply fixes the pivot, about which the platform rotates, against translation and rotation (Fig. 2.23). The equations are:

Frame:

$$\left.\begin{aligned} v_{Px} &= 0 \\ v_{Py} &= 0 \\ \omega_P &= 0 \end{aligned}\right\} \tag{2.74}$$

The Pivot allows rotation only about the pin (Fig. 2.24). Two flow junctions are inserted to extract information on pin force components. Rotation is assumed frictionless, but friction can be added if required, e.g. by a resistive component R used instead of the source effort. The relative angular velocity of the platform is denoted by ω_r.The signal taken from the effort junction is integrated to get the platform rotation angle ϕ. The governing equations are again very simple:

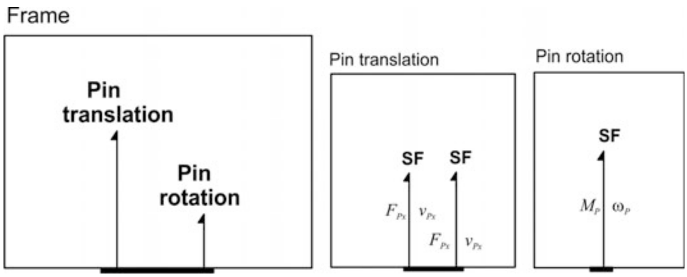


Fig. 2.23 Model of the see-saw frame

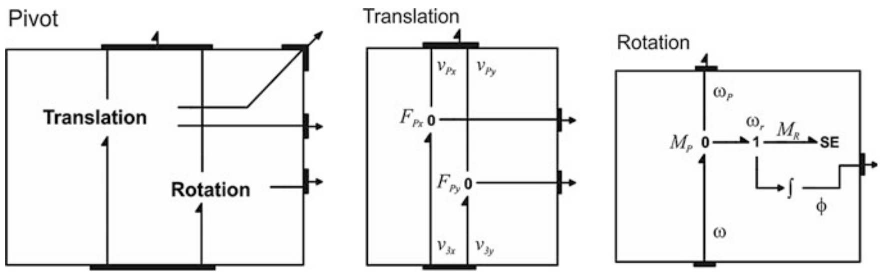


Fig. 2.24 Model of the see-saw pivot

Pivot:

$$\left.\begin{aligned} v_{3x} - v_{Px} &= 0 \\ v_{3y} - v_{Py} &= 0 \\ \omega - \omega_P - \omega_r &= 0 \\ M_P - M_R &= 0 \\ M_R &= 0 \\ \dot{\phi} &= \omega_r \end{aligned}\right\} \tag{2.75}$$

The platform acts as a transformer of the velocities of the attached bodies. Simultaneously, the transformation of the reaction forces of the bodies also takes place. To develop the bond graph model of the platform we analyse the plane motion of the platform in the global co-ordinate frame Oxy (Fig. 2.25).

The position and orientation of the platform is defined by the body frame $C\bar{x}\bar{y}$ with the origin at its mass centre. The position vector of the origin C is described by a column vector of its global co-ordinates, i.e.

$$\mathbf{r}_C = \begin{pmatrix} x_C \\ y_C \end{pmatrix} \tag{2.76}$$

Orientation of the body is defined by the rotation matrix (see e.g. [7])

$$\mathbf{R} = \begin{pmatrix} \cos\,\varphi & -\sin\,\varphi \\ \sin\,\varphi & \cos\,\varphi \end{pmatrix} \tag{2.77}$$

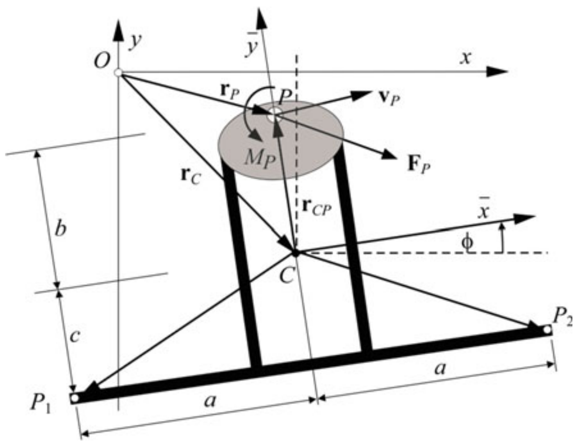


Fig. 2.25 See-saw platform plane motion

The vector of the position of a point P in the body with respect to the origin of the body frame can be expressed in the body frame by a vector of its coordinates

$$\bar{\mathbf{r}}_{CP} = \begin{pmatrix} \bar{x}_{CP} \\ \bar{y}_{CP} \end{pmatrix} \tag{2.78}$$

The position of the same point P with respect to the global frame is defined by the vector of its global co-ordinates

$$\mathbf{r}_P = \begin{pmatrix} x_P \\ y_P \end{pmatrix} \tag{2.79}$$

The relationship between these vectors is given by

$$\mathbf{r}_P = \mathbf{r}_C + \mathbf{r}_{CP} \tag{2.80}$$

Note that vector $\mathbf{r}_{CP}$ is the relative vector expressed in the global frame, i.e.

$$\mathbf{r}_{CP} = \begin{pmatrix} x_{CP} \\ y_{CP} \end{pmatrix} \tag{2.81}$$

The relationship between the vectors of (2.78) and (2.81) is given by the co-ordinate transformation

$$\mathbf{r}_{CP} = \mathbf{R}\bar{\mathbf{r}}_{CP} \tag{2.82}$$

Substituting the rotation matrix of (2.77) and evaluating the resulting expression yields

$$\begin{pmatrix} x_{CP} \\ y_{CP} \end{pmatrix} = \begin{pmatrix} \bar{x}_{CP} \cos\varphi - \bar{y}_{CP} \sin\varphi \\ \bar{x}_{CP} \sin\varphi + \bar{y}_{CP} \cos\varphi \end{pmatrix} \tag{2.83}$$

The velocity of a point P in the body can be found by taking the time derivative of (2.80), i.e.

$$\mathbf{v}_P = \mathbf{v}_C + \mathbf{v}_{CP} \tag{2.84}$$

which relates the velocity of the point P to the velocity of the origin C of the body frame and to the relative velocity of the point P with respect to the point C. These velocity vectors are expressed by their components in the global frame as

$$\mathbf{v}_P = \begin{pmatrix} v_{Px} \\ v_{Py} \end{pmatrix}, \quad \mathbf{v}_C = \begin{pmatrix} v_{Cx} \\ v_{Cy} \end{pmatrix}, \quad \mathbf{v}_{CP} = \begin{pmatrix} v_{CPx} \\ v_{CPy} \end{pmatrix} \tag{2.85}$$

and are defined by

$$\mathbf{v}_P = \frac{d\mathbf{r}_P}{dt}, \quad \mathbf{v}_C = \frac{d\mathbf{r}_C}{dt}, \quad \mathbf{v}_{CP} = \frac{d\mathbf{r}_{CP}}{dt} \tag{2.86}$$

Taking the time derivative of (2.82), and noting that $\bar{\mathbf{r}}_{CP}$ is a constant vector, we arrive at the expression for the relative velocity of point P:

$$\mathbf{v}_{CP} = \frac{d\mathbf{R}}{dt}\bar{\mathbf{r}}_{CP} \tag{2.87}$$

The time derivative of the rotation matrix $\mathbf{R}$ in (2.77) yields

$$\frac{dR}{dt} = \begin{pmatrix} -\sin\varphi & -\cos\varphi \\ \cos\varphi & -\sin\varphi \end{pmatrix} \cdot \frac{d\varphi}{dt} \tag{2.88}$$

The time derivative of the body rotation angle is the body angular velocity

$$\omega = \frac{d\varphi}{dt} \tag{2.89}$$

Thus, substitution of (2.88) and (2.89) into (2.87) yields

$$\mathbf{v}_{CP} = \mathbf{T}\omega \tag{2.90}$$

where $\mathbf{T}$ is the transformation matrix, given by

$$T = \begin{pmatrix} -\bar{x}_{CP}\sin\varphi - \bar{y}_{CP}\cos\varphi \\ \bar{x}_{CP}\cos\varphi - \bar{y}_{CP}\sin\varphi \end{pmatrix} \tag{2.91}$$

Compared with (2.83), this matrix also can be expressed as

$$\mathbf{T} = \begin{pmatrix} -y_{CP} \\ x_{CP} \end{pmatrix} \tag{2.92}$$

Equations (2.84) and (2.89)–(2.91) are the basic relations describing the kinematics of rigid body motion in a plane. Next, we consider the kinetic relationships relating the forces and moments applied to the platform.

A force $\mathbf{F}$ applied to the platform can be described by a vector of its rectangular components in the global frame, i.e.

$$\mathbf{F} = \begin{pmatrix} F_x \\ F_y \end{pmatrix} \tag{2.93}$$

The power delivered at point P is given by $\mathbf{v}_P^T\mathbf{F}$, where the superscript T denotes matrix transposition. From (2.84) and (2.90) we get

$$\mathbf{v}_P^T\mathbf{F} = \mathbf{v}_C^T\mathbf{F} + \mathbf{T}^T\mathbf{F}\omega \tag{2.94}$$

Evaluating the leading part of the second term on the right of (2.94) yields

$$\mathbf{T}^T\mathbf{F} = -y_{CP}F_x + x_{CP}F_y \tag{2.95}$$

We recognise this as the moment M_C of the force at P about point C, thus

$$M_C = \mathbf{T}^T\mathbf{F} \tag{2.96}$$

By substituting in (2.94), we finally arrive at the equations of power transfer across the body

$$\mathbf{v}_P^T\mathbf{F} = \mathbf{v}_C^T\mathbf{F} + M_C\omega \tag{2.97}$$

This equation can be read as a statement of *force equivalents*, well known from Engineering Mechanics (see e.g. [8]). That is, a force applied at a point P is equivalent to the same force applied at a different point C plus the moment of force about C. If at point P a torque also acts, its moment M_P should be added, too. Equations (2.96) and (2.97), jointly with (2.84) and (2.89)–(2.91), constitute the fundamental equations of rigid-body motion in a plane. To complete the dynamical equations we need to add the inertias of translation and rotation. These equations clearly show how to represent the dynamics of the platform (see Fig. 2.26 `Platform`).

At every point of application of a force (platform ports) we introduce a component `0` corresponding to summation of the velocities, as given by (2.84). These components contain two flow junctions. The corresponding junction variables are the x and y components of the force at the port (Fig. 2.26 `0`). The effort junction `1` is used to represent the angular velocity of the body, and the component `CM` describing the motion of the mass centre (Fig. 2.26 `Platform` and `CM`). We connect the junctions 0 to the angular velocity junction 1 by the components `LinRot`, and to the mass centre motion component `CM`.[3]

The `LinRot` components represent the *linear to rotation* transformations given by (2.90) and (2.96). The components consist of two transformers, which implement the transformations by matrix given by (2.91), and an effort junction that sum up the moments according to (2.96) (see Fig. 2.21 `LinRot`). The necessary

[3]Because of space limitation, only one of the **`0`** and **`LinRot`** components are shown. The others have a similar structure.

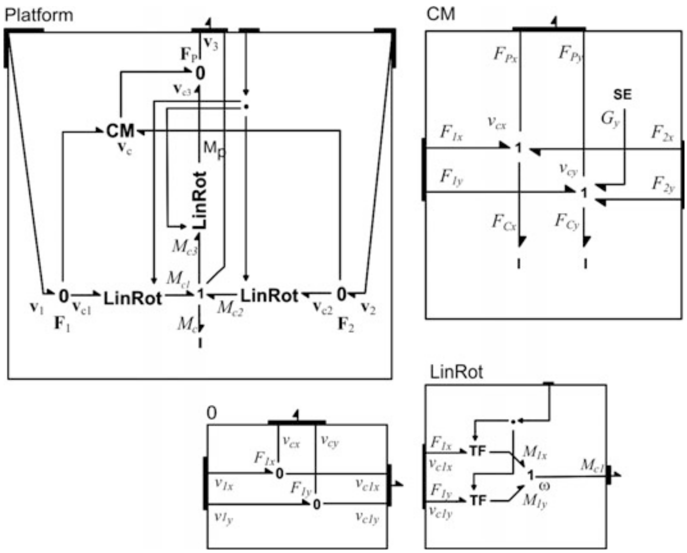


Fig. 2.26 Model of the platform

information on the angle of rotation of the see-saw platform is taken from the input port. In addition to these force effects, any moment at a port is transmitted directly to the rotation effort junction `1`. An inertial element added to the junction represents the rotational inertia of the platform with respect to mass center. The body translation inertia with mass center is represented by component `CM` which consists of two effort junctions that add inertial elements corresponding to the x and y motion (Fig. 2.26 `CM`). The platform gravity is also added there.

The mathematical model of the platform can be written directly from the Fig. 2.26. Respective variables are given in the figure and parameters a, b, and c are dimensions shown in Fig. 2.25; m is the platform mass, and I_C is its mass moment of inertia about its mass centre.

The equations read:

Platform—left side:

$$\left.\begin{aligned}
&v_{1x} - v_{C1x} - v_{Cx} = 0\\
&v_{1y} - v_{C1y} - v_{Cy} = 0\\
&v_{C1x} = (a \cdot \sin\varphi + c \cdot \cos\varphi) \cdot \omega\\
&v_{C1y} = (-a \cdot \cos\varphi + c \cdot \sin\varphi) \cdot \omega\\
&M_{1x} = (a \cdot \sin\varphi + c \cdot \cos\varphi) \cdot F_{1x}\\
&M_{1y} = (-a \cdot \cos\varphi + c \cdot \sin\varphi) \cdot F_{1y}\\
&-M_{C1} + M_{1x} + M_{1y} = 0
\end{aligned}\right\} \tag{2.98}$$

Platform—right side:

$$\left.\begin{aligned}
&v_{2x} - v_{C2x} - v_{Cx} = 0\\
&v_{2y} - v_{C2y} - v_{Cy} = 0\\
&v_{C2x} = (-a \cdot \sin\varphi + c \cdot \cos\varphi) \cdot \omega\\
&v_{C2y} = (a \cdot \cos\varphi + c \cdot \sin\varphi) \cdot \omega\\
&M_{2x} = (-a \cdot \sin\ + c \cdot \cos\varphi) \cdot F_{2x}\\
&M_{2y} = (a \cdot \cos\varphi + c \cdot \sin\varphi) \cdot F_{2y}\\
&-M_{C2} + M_{2x} + M_{2y} = 0
\end{aligned}\right\} \tag{2.99}$$

Platform—upper side:

$$\left.\begin{aligned}
&-v_{3x} + v_{C3x} + v_{Cx} = 0\\
&-v_{3y} + v_{C3y} + v_{Cy} = 0\\
&v_{C3x} = -(b \cdot \cos\varphi) \cdot \omega\\
&v_{C3y} = -(b \cdot \sin\varphi) \cdot \omega\\
&M_{3x} = -(b \cdot \cos\varphi) \cdot F_{Px}\\
&M_{3y} = -(b \cdot \sin\varphi) \cdot F_{Py}\\
&M_{C3} - M_{3x} - M_{3y} = 0
\end{aligned}\right\} \tag{2.100}$$

Platform—mass centre motion:

$$\left.\begin{aligned}
&\dot{p}_{Cx} = F_{Cx}\\
&\dot{p}_{Cy} = F_{Cy}\\
&p_{Cx} = m \cdot v_{Cx}\\
&p_{Cy} = m \cdot v_{Cy}\\
&G_y = -mg\\
&F_{1x} + F_{2x} - F_{Px} - F_{Cx} = 0\\
&F_{1y} + F_{2y} - F_{Py} - F_{Cy} + G_y = 0
\end{aligned}\right\} \tag{2.101}$$

Platform—rotation:

$$\left.\begin{aligned}
&\dot{K}_C = M_C\\
&K_C = I_C \cdot \omega\\
&M_{C1} + M_{C2} - M_{C3} - M_P - M_C = 0
\end{aligned}\right\} \tag{2.102}$$

The model consists of fifty-four very simple equations. No substitutions or other simplifications have been made, as we wished to develop the model strictly by describing every elementary component in terms of its variables and parameters.

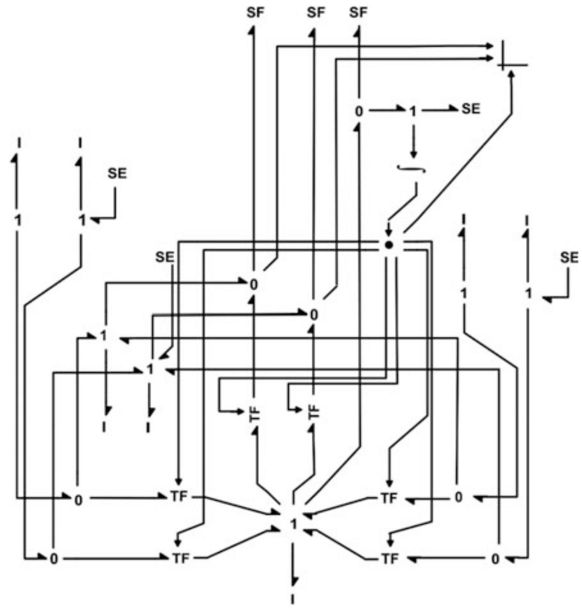


Fig. 2.27 Single level model of the see-saw

This procedure, based on the systematic decomposition, results in multi-level models. Such models can be developed and changed more easily, if necessary, than conventional "flat" models. Some of the components can also be reused in other models. For example, the `Platform` component can be used for problems dealing with the plane motion of rigid bodies. For comparison, a flat model corresponding to the model developed above is shown in Fig. 2.27. Such a model, however, is not easy to follow, particularly for people unfamiliar with bond graphs: There are many bonds, and it is not easy even to draw them correctly! It is thus more susceptible to errors and more difficult to change.

2.8 Causality of Bond Graphs

2.8.1 The Concept of Causality

The concept of *causality*, or *cause-effect relationships*, was introduced in the bond graph method to define the *computational* structure of the resulting mathematical equations at bond-graph level. Thus, the physical and computational structure of the model is defined in parallel during the modelling stage. It should be stressed that physical laws do not imply any causal preference: There is no physical reason to treat forces as the cause, and velocities of the body motions as effects; or voltages as

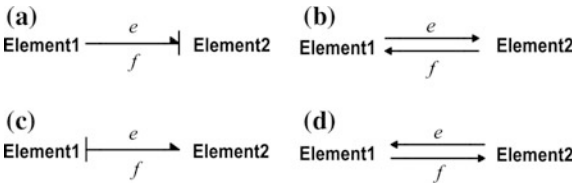


Fig. 2.28 Causality assignment: **a** and **c** possible stroke attachments, **b** and **d** meaning of the attachments

the cause, and currents in circuits as effects. The assignment of causality can be looked on as a convenient—but not an essential—part of the modelling task. Further it is arguable that it is convenient at all, in particular when using the object-oriented paradigm in simulation model building. We nevertheless briefly describe causality and its consequences in bond graphs because of their close connection with bond graph theory (see e.g. [1, 2]).

Causality means that, at every port of an elementary component, one of the power variables is the input (cause) and the other is the output (effect). Because bond lines in bond graphs connect the ports, the same variable is the input variable at one port and the output variable at the other connected port. Causal relationships between connected port variables are depicted in the bond graph literature by *causal strokes*. These are short lines drawn at one bond end (port) perpendicular to the bond (Fig. 2.28). This stroke denotes that the effort at the port is the input to the element and the flow variable is the output (Fig. 2.28a, b). At the other port just the opposite relation is valid; that is, the flow variable is the input and the effort variable is the output. Causal stroke assignment is independent of the power flow direction (Fig. 2.28a, c).

2.8.2 Causalities of Elementary Components

The causality assignment defines the input-output relationship of the elementary component constitutive relations. Possible types of causalities of elementary component ports are summarised in Fig. 2.29.

Source effort ports (Fig. 2.29a) can have only one possible type of causality, i.e. the effort is always the output, because flow at the input port is not defined. Similarly, at source flow ports the output is the flow, because the effort is not defined (2.29b). Thus, sources have *fixed causalities*.

The inertial component can have one of two possible causalities. If effort at the port is the input and the flow is output (Fig. 2.29c), the constitutive relations are then given by (2.2) and (2.5)

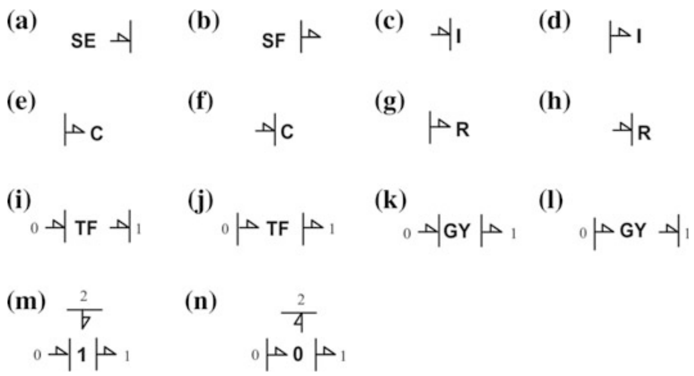


Fig. 2.29 Causalities of elementary components

$$\left.\begin{aligned} p &= p_0 + \int_0^t e dt \\ f &= \Phi^{-1}(p, par) \end{aligned}\right\} \tag{2.103}$$

Such causality is known as *integrating causality* because *integration* is used to calculate the output flow.

The other possibility is that the flow is the input and effort is the output (Fig. 2.29d). In this case evaluation proceeds by (2.4) and (2.1), i.e.

$$\left.\begin{aligned} p &= \Phi(f, par) \\ e &= \dot{p} \end{aligned}\right\} \tag{2.104}$$

This type of causality is known as *differentiation causality* because *differentiation* is used to calculate the output.

Analogous causal forms exist for capacitive ports. If we take the flow variable as input and the effort variable as output (Fig. 2.29e), by (2.10) and (2.13) we have

$$\left.\begin{aligned} q &= q_0 + \int_0^t f dt \\ e &= \Phi^{-1}(q, par) \end{aligned}\right\} \tag{2.105}$$

In this case we have integrating causality. On the other hand, if effort is the input and flow is the output (Fig. 2.24f), the calculation proceeds by (2.12) and (2.9), i.e.

$$\left.\begin{array}{l} q = \Phi(e, par) \\ f = \dot{q} \end{array}\right\} \tag{2.106}$$

This yields differentiation causality.

Of these two possible causalities, integrating causality is *preferred* because integration is more easily implemented than differentiation. This is because integration works on the past values, whereas differentiation involves prediction.

For the resistor there are also two possible causalities. If the flow is the input and effort is the output (Fig. 2.29g), evaluation of the output is done using (2.18), i.e.

$$e = \Phi(f, par) \tag{2.107}$$

On other hand, if the effort is input (Fig. 2.29h) and the flow is output, calculation is implemented by (2.19), i.e.

$$f = \Phi^{-1}(e, par) \tag{2.108}$$

The first one is sometimes called *resistive causality*, and the other *conductive causality*. Preference of one over the other depends on which form is better defined, as some non-linear constitutive relationships are not invertible.

Transformers can also have two possible types of causality. If the effort at one port is the input, then at the other port the effort has to be the output; the same applies to the flows. For causality as expressed in Fig. 2.29i, the constitutive relations are given by (2.27), i.e.

$$\left.\begin{array}{l} e_1 = m \cdot e_0 \\ f_0 = m \cdot f_1 \end{array}\right\} \tag{2.109}$$

On other hand, if causality is as in Fig. 2.29j, the constitutive relations are given by (2.28), i.e.

$$\left.\begin{array}{l} e_0 = k \cdot e_1 \\ f_1 = k \cdot f_0 \end{array}\right\} \tag{2.110}$$

Two possible causalities for gyrators are shown in Fig. 2.29k, l, respectively. Inputs at the gyrator ports can represent either the efforts or the flows. For the case in which inputs are *efforts*, output flows are given by (2.30), i.e.

$$\left.\begin{array}{l} f_0 = k \cdot e_1 \\ f_1 = k \cdot e_0 \end{array}\right\} \tag{2.111}$$

Similarly, if the inputs are *flows*, the output efforts are given by (2.29), i.e.

$$\left.\begin{aligned} e_0 &= m \cdot f_1 \\ e_1 &= m \cdot f_0 \end{aligned}\right\} \tag{2.112}$$

Effort junctions represent the balance of efforts at the junction ports. Hence, one effort can only be the output at one port; all others must be inputs. For the effort junction in Fig. 2.29m, the effort at port 2 is taken as the output and all others are inputs. Thus, output effort e_2 is given by

$$e_2 = -e_0 + e_1$$

A similar statement holds for the flow junctions: Flow can only be the output at one port; all other flows must be inputs. For the flow junction in Fig. 2.24n, the output flow f_1 is given by

$$f_1 = f_0 - f_2$$

In the expression for output effort or flow, the sign of all input efforts or flows should be positive if the sense of the power flow is opposite to the sense of the output power flow. Otherwise, the sign is negative.

2.8.3 The Procedure for Assigning Causality

The causalities of junction, transformer, and gyrator ports are interrelated and thus imply constraints on the causalities of connected elements. The causalities of the complete bond graph can be assigned in a systematic way. The usual procedure is known generally as the *sequential causal assignment procedure* (SCAP) [1]. This procedure is summarised as follows:

1. Choose a source effort or source flow and assign causality to it. Extend the causality assignment, if possible, to the connected effort and flow junctions, the transformers, and the gyrators. Proceed in a like fashion until the causality of all sources has been assigned.
2. Choose an inertial or a capacitive element and assign to it the preferred (integrating) causality. Extend the causality assignment as in 1. Proceed until the causality of all such elements has been assigned. Otherwise, if the causality assignment of the all bonds is not achieved, go to the next step.
3. Assign causality to an unassigned resistor using any acceptable causality. Extend the assignment to the connected effort and flow junctions, transformers, and gyrators. Proceed until the causality of all resistors has been assigned. Otherwise, if the causality of all bonds is not already assigned, go to the next step.

4. Assign causality to any remaining bond. Extend the causality assignment to effort and flow junctions, transformers, and gyrators. Proceed until the causality of all bonds is assigned.

The bond graph to which causality has been assigned usually is termed a *causal* bond graph. Otherwise, it is termed an *acausal* bond graph.

The procedure can be illustrated on the simple body-spring-damper problem of Sect. 2.7.1. Other more complex examples can be found in, for example, [1]. Here we use the simplified bond graph of Fig. 2.16c, which is repeated in Fig. 2.30a.

We start with the source effort SE (step 1 of SCAP) and assign its causality as shown in Fig. 2.30b. We cannot extend the causality assignment immediately to the effort junction, as the connected port is an effort input port. There are no more sources, thus we proceed with step 2 of SCAP. We can choose to assign causality to either the inertial or the capacitive element. Let us choose to assign integration causality to the inertial element I (Fig. 2.30c). We now can extend the causality assignment to the effort junction, because the port connected to the inertial port is an effort output port, and all other junction ports must be effort input ports (Fig. 2.30d). This completes the causality assignment of the bond graph.

We have obtained integration causality of the capacitive element C, as well. If we start at step 2 by choosing the capacitive element instead of the inertial element, we would have to assign the causality of the inertial element before we could proceed to the effort junction. The first procedure is somewhat shorter.

The causal assignment of Fig. 2.30 defines the order of evaluation of the equations. This is shown by the block diagram of Fig. 2.31.

We start with the SE first. Next, we calculate the output flow of the inertial element. This is the input to the effort junction and the output of its all other ports. It

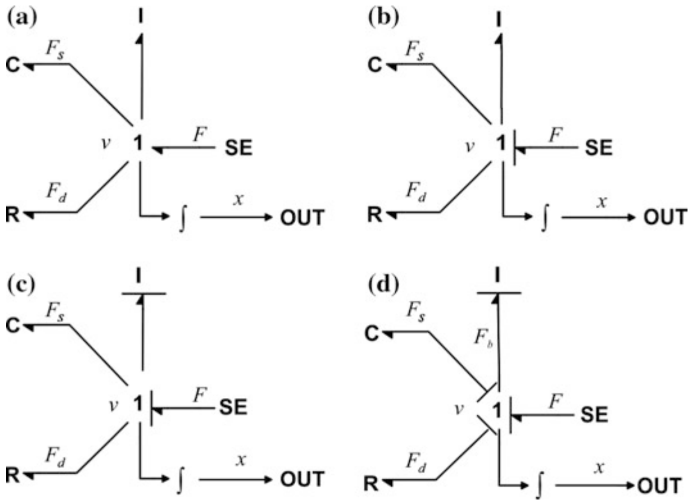


Fig. 2.30 Illustration of the causality assignment procedure

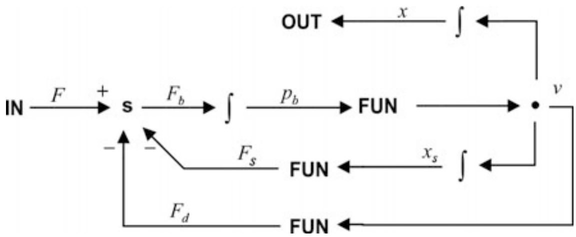


Fig. 2.31 Computational order of the bond graph Fig. 2.30d

is the input to the capacitor and the resistor used to calculate their outputs. These, together with the source effort output, are used to calculate the output of the effort junction, hence the inertial element input. Independently, it is used as input to the integrator to calculate body position.

2.9 The Formulation of the System Equations

The bond graph of a system completely defines its mathematical model. In Sect. 2.7 it was shown that the model could be generated directly from the bond graph by describing the elementary components, including junctions, in terms of their constitutive relations. This way of representing mathematical models is known as the *descriptor* form [9] and is widely used in electrical circuits. This is a non-minimum form because the equations are not expressed using the minimal number of variables. Some variables could be eliminated, e.g. by substituting into the equations of flow and effort junctions. This approach is in effect used in modified nodal analysis (MNA) of electrical circuits [10, 11]. This also is the case with certain approaches used in multi-body dynamics [7]. In Sect. 2.7 it was shown that the matrix of the equations is typically very sparse, and this can be used to advantage in their solution.

The descriptor form of equation formulation leads to the model in the form of systems of differential-algebraic equations (DAEs). The success or failure of the descriptor formalism depends to a large extent on the possibility of solving equations in DAE form efficiently and reliably. Solving such equations has a relatively long history and started with the famous DIFSUB routine of CW Gear [12] for stiff systems. The work reported in this book also has its roots in software that solves DAEs in a way that is based on the DIFSUB routine. From that time significant advances have been achieved in the theory of DAEs and their application [13, 14]. Today this is a viable approach to solving simulation models. We return to this again in Chap. 5.

Another common approach is to formulate the system in *state space* form. This technique uses a minimal set of independent variables to formulate the governing equations. It has its roots in the generalised coordinate methods of Analytical Mechanics [15], but it also is used widely, and is perhaps better known, from

Control Theory. The theory of state-space equations has been a topic of research for a long time and is well understood. This approach is followed not only in bond graph theory, but is also used in many continuous system simulation languages (Sect. 1.7).

The usual approach in continuous system simulation languages is to create a system of sorted equations that is solved sequentially. Such systems can be solved relatively easily. Unfortunately, in many engineering problems of practical interest it is not easy to put the equations in such a form.

The sequential causal assignment procedure of Sect. 2.8.3 was really designed as an aid to the generation of mathematical model equations in sorted form. From that the equations can be reduced to the state space form. The bond graphs with completed causality assignment can be put in such a form if inertial and capacitive elements have integrating causality, and if there are no algebraic loops [1, 16]. We illustrate this with the body-spring-damper system represented by the causal bond graph of Fig. 2.30d (or, equivalently, by the block diagram of Fig. 2.31). More elaborate examples can be found elsewhere [1].

We start with the source effort (sixth of (2.59)), following the order of causal assignment of Sect. 2.8,

$$F = \Phi(t) \tag{2.113}$$

The output of the inertial element **I** is given by (see (2.103) and fifth (2.59))

$$v = p_b/m \tag{2.114}$$

The variable v (by the effort junction) is used as the input to the capacitor C, resistor R, and the integrator. The order of evaluation of these elements is immaterial. From the first (2.105), written in derivative form, or first (2.59), we get

$$\dot{x}_s = v \tag{2.115}$$

Output of the capacitor is given by the equation (see (2.105) and second (2.59))

$$F_s = k \cdot x_s \tag{2.116}$$

Output of the resistor (see (2.107) or the third (2.59)) is

$$F_d = b \cdot v \tag{2.117}$$

Hence, all the inputs to the summator are found and we can calculate its output as

$$F_b = F - F_s - F_d \tag{2.118}$$

The output of the summator is the input to the inertial element. Thus, from the first equation of (2.103), written in differential form, or fourth (2.59), we get

$$\dot{p}_b = F_b \tag{2.119}$$

To these equations we add the output of the integrator written as (the last (2.59))

$$\dot{x} = v \tag{2.120}$$

This completes the generation of the system of sorted equations.

The equations above consist of differential equations (2.115), (2.119) and (2.120), and algebraic equations (2.113), (2.114), (2.116), (2.117), and (2.118). Hence, it is a differential/algebraic system of equations (DAE), but of a special structure. We classify all variables in these equations as being either differentiated or participating in algebraic operations only. The first are called *differentiated variables*, i.e. x_s, x and $\mathrm{p_b}$. The others are *algebraic* variables; in the equations above these are F, v, F_s, F_d, F_b. All algebraic variables above can be expressed as functions of the differentiated variables and time. We see that the variables F, v, and F_s are already in this form (see (2.113), (2.114) and (2.116)). Eliminating v from (2.117) and (2.114) we get

$$F_d = \frac{b}{m} \cdot p_b \tag{2.121}$$

Finally, substituting from (2.113), (2.116), and (2.121) into (2.118) we obtain

$$F_b = \Phi(t) - k \cdot x_s - \frac{b}{m} \cdot p_b \tag{2.122}$$

We now substitute these expressions into the differential equations (2.115), (2.119), and (2.120). We, thus, obtain

$$\dot{x}_s = p_b/m \tag{2.123}$$

$$\dot{p}_b = \Phi(t) - k \cdot x_m - \frac{b}{m} \cdot p_b \tag{2.124}$$

$$\dot{x} = p_b/m \tag{2.125}$$

Note that (2.123) and (2.125) has the same form, but generally different initial conditions, because the first refers to the spring extension, and the second to the body position.

Equations (2.123)–(2.125) represent the model in the *state-space form*. Variables x_s, p_b and x constitute a minimal set of independent variables that completely define the state of the system. Solving these equations with suitable initial conditions, the all other variables can be found from (2.113), (2.114), (2.116)–(2.118).

In general, if all capacitive and inertial ports have integrating causality, then the corresponding differentiated variables, i.e. generalised moments and displacements of Sects. 2.5.2 and 2.5.3 can be looked upon as independent variables where accumulation of past histories of the efforts and flows take place. Such variables completely determine the future state of the system and usually are called *state variables*. All other variables can then be determined if the state of the system is known. If, in addition, there are no algebraic loops—that is, there are no implicit algebraic equations between variables—then all other variables can be eliminated from the governing equations. Thus, if all state variables are represented by a vector $\mathbf{p}$ and all external inputs (represented by the sources) by a vector $\mathbf{u}$, then a change of system state can be described by a vector equation

$$\dot{\mathbf{p}} = \mathbf{\Phi}(\mathbf{p}, \mathbf{u}, t) \tag{2.126}$$

where $\mathbf{\Phi}$ is a suitable vector-function of the state, inputs, and eventually time. This is an ordinary differential equation that can be solved given the *initial state* of the system. Such an equation is termed the *state-space equation* of the system.

2.10 The Causality Conflicts and Their Resolution

The sequential causal assignment procedure (SCAP) of Sect. 2.8.3, in many cases of practical interest, leads to a causally augmented graph that cannot be described by equations in state space form [1, 17, 18, 19]. We illustrate this using the examples of Sect. 2.7.

We first analyse the electrical circuit of Fig. 2.18, but with the resistor replaced by a diode (Fig. 2.32a).

If we model the diode as a non-linear resistor, we get the equivalent causally augmented bond graph shown in Fig. 2.32b (see Fig. 2.19c). The problem here is that the diode is a non-linear element normally described in conductive form, i.e. the diode current is a function of the voltage across the diode. It is thus in conflict with the assigned causality that implies resistive causality. In order to resolve conflicts caused by non-linear elements, the relaxed causal assignment procedure was proposed in [20] and its modification in [21]. This procedure requires that, at step 2 of the SCAP (Sect. 2.8.3), propagation of causalities over junctions may not violate the fixed causality of non-linear elements. Thus, applying the causal assignment procedure again results in the augmented bond graph of Fig. 2.32c. The conflict caused by the fixed causality of the diode disappears, but a *causal conflict* appears at the effort junction because there is more than one output. Thus, the equation corresponding to the effort junction constitutes an algebraic constraint that the variables have to satisfy. The procedure permits casual conflicts at effort or flow junctions as an indication that the mathematical model is of the differential-algebraic equations (DAE) form, rather than of the state space form.

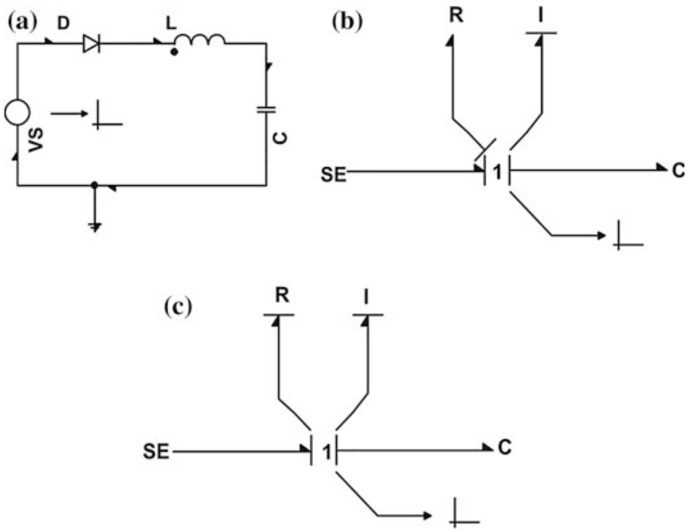


Fig. 2.32 Causality conflict in the electric circuit with a diode

In the see-saw problem (Sect. 2.7.3) there also is a causality problem. The see-saw is a single-degree-of-freedom mechanical system. The motions of the bodies depend on the motion (rotation) of the platform, which is represented in Fig. 2.26 by the `LinRot` transformers. To show this we apply the SCAP to the bond graph of Fig. 2.27. The resulting causally augmented bond graph is shown in Fig. 2.33. There is only one inertial element with integrating causality. All others have differential causalities.

This bond graph is rather complicated, so numbers are used to indicate the order of the causality assignments. Selection of the preferred (integrating) causality for one inertial element, e.g. the platform rotation, implies derivative causalities for all other inertial elements. Hence, there is only one state variable and all other generalized variables are non-state variables. The model again is a system of differential-algebraic equations.

There have been attempts to resolve causality conflicts by, for example, adding 'parasitic' compliances or inertias [22, 23]. This is not an acceptable approach, however, because, in the first instance, it is not clear how to do this without adversely changing the model behaviour. On the other hand, such modified models are not much easier to solve numerically than the corresponding DAE models because they are very stiff.

The causality assignment defines the model's computational scheme based only on the model's structure. In cases in which the model changes sufficiently, such a priori schemes can lead easily to a loss of efficiency and even failure of the equation solving routines. This is the case with models that have discontinuities.

Discontinuities are present in engineering systems in various forms, e.g. switches in electrical circuits, hard stops, clearances, and dry friction. For example,

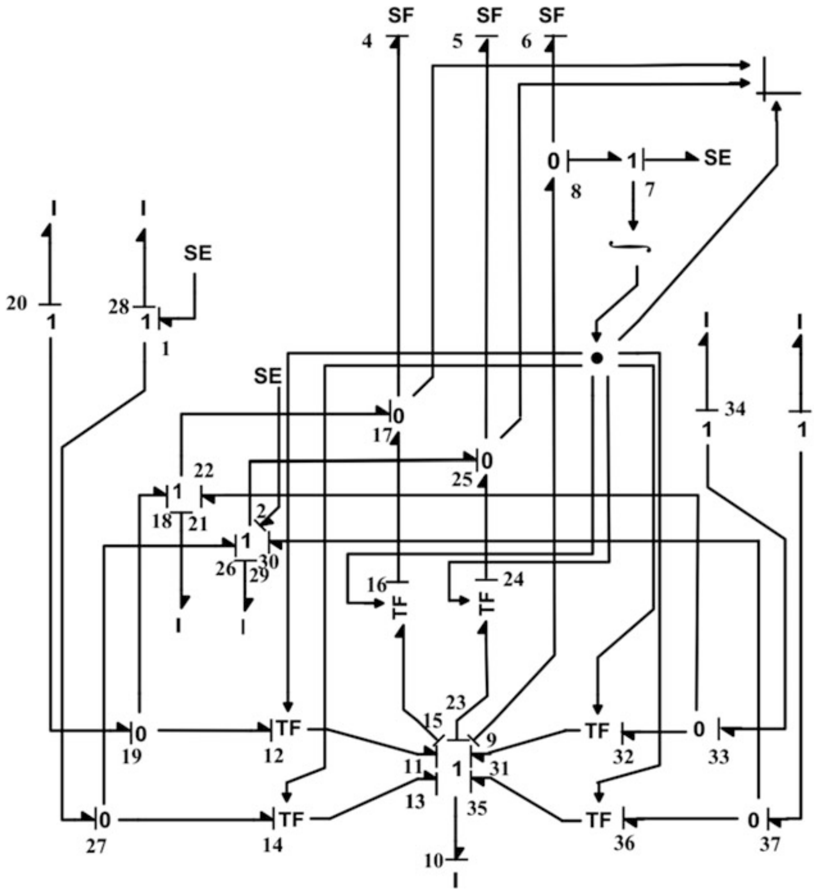


Fig. 2.33 Causal bond graph of Fig. 2.27

the diode represented in the circuit of Fig. 2.34a is modelled as a switch in Fig. 2.34b.

If the diode is forward-biased (conducting), then the switch behaves as a source effort implying a zero voltage drop across the diode (Fig. 2.34c). When, on the other hand, the diode is reverse-biased, the switch behaves as a flow source of small reverse saturated current (Fig. 2.34d). The model structure and causalities are apparently different for these two states. In the conducting regime the system has two state variables, while in the non-conducting regime it has only one.

There have been various attempts to solve causality problems with switches [24–28]. Overall, these procedures are not completely satisfactory in the general case. This is particularly true if the discontinuities are not confined to switch elements, but appear in the element constitutive relations, too.

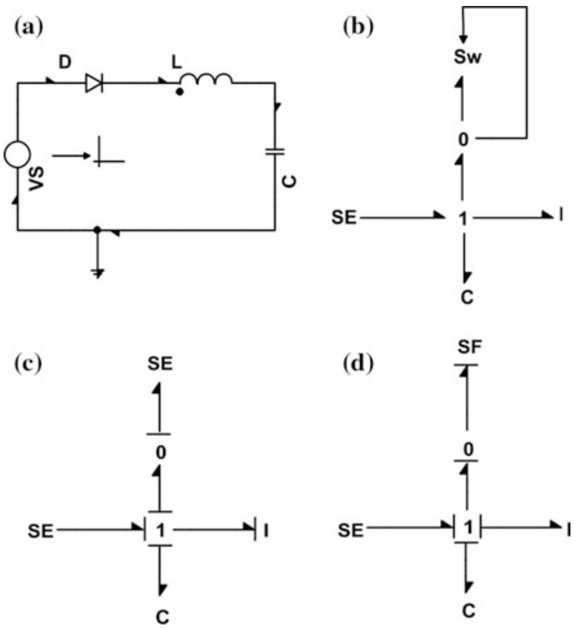


Fig. 2.34 Change of causality pattern in the circuit with a switch

The concept of causality is generally not suitable for use in an automated object-oriented modelling environment. It is not only too restrictive with respect to the forms of models that can be used, but also puts restrictions on the design and usability of models libraries. A component that has one causality pattern in one system can have a quite different one when inserted in another system. We disregard causality issues when developing models of general engineering and mechatronic systems. Modelling is treated as a separate task from model simulation. The models will be generated in the form of DAE systems and solved as such.

References

1. Karnopp DC, Margolis DL, Rosenberg RC (2000) System dynamics: modeling and simulation of mechatronic systems, 3rd edn. Wiley, New York
2. Thoma J, Bousmsma BO (2000) Modelling and simulation in thermal and chemical engineering, a bond graph approach. Springer, Heidelberg
3. Borutzky W (2010) Bond graph methodology: development and analysis of multidisciplinary dynamic system models. Springer, Berlin
4. Breedveld PC (1982) Proposition for an unambiguous vector bond graph notation. J Dyn Syst Measure Control 104:267–270
5. Fahrenthold EP, Wargo JD (1991) Vector and tensor based bond graphs for physical systems modeling. J Franklin Inst 328:833–853
6. Analog-to digital converter. http://en.wikipedia.org/w/index.php. Accessed 22 Apr 2014

7. Haug EJ (1989) Computer-aided kinematics and dynamics of mechanical systems, Vol I: basic methods. Allyn and Bacon, Needham Heights
8. Beer FP, Johnson ER (1990) Vector mechanics for engineers, 2nd SI edn. McGraw-Hill Book Co., Singapore
9. Newcomb RW (1981) Semistate description of nonlinear and time variable circuits. IEEE Trans Circuit Syst CAS-26:62–71
10. Märtz R, Tischendorf K (1997) Recent results in solving index-2 differential-algebraic equation in circuit simulation. SIAM J Sci Comput 18:135–159
11. Vladimirescu A (1994) The spice book. Wiley, New York
12. Gear CW (1971) Numerical initial-value problems in ordinary differential equations. Prentice Hall, Englewood Cliffs
13. Brenan KE, Cambell SL, Petzold LR (1996) Numerical solution of initial-value problems in differential-algebraic equations, classics in applied mathematics. SIAM, Philadelphia
14. Hairer E, Wanner G (1996) Solving ordinary differential equations II, stiff and differential-algebraic problems, 2nd Revisited edn. Springer, Heidelberg
15. Goldstein H (1981) Classical mechanics, 2nd edn. Addison-Wesley Publishing Co., Reading
16. Rosenberg RC (1971) State-space formulation of bond graph models of multiport systems. Trans ASME J Dyn Syst Measure Control 93:35–40
17. Van Dijk J, Breedveld PC (1991) Simulation of system models containing zero-order causal paths—I. Classification of zero-order causal paths. J Franklin Inst 328:959–979
18. Van Dijk J, Breedveld PC (1991) Simulation of system models containing zero-order causal paths—II. Numerical implications of class 1 zero-order causal paths. J Franklin Inst 328:981–1004
19. Gawthrop P, Smith L (1996) Metamodelling: bond graphs and dynamic systems. Prentice Hall, Hemel
20. Joseph BJ, Martens HR (1974) The method of relaxed causality in the bond graph analysis of nonlinear systems. Trans ASME J Dyn Syst Measure Control 96:95–99
21. Van Dijk J, Breedveld PC (1995) Relaxed causality a bond graph oriented perspective on DAE-modelling. In: Cellier FE, Granda JJ (eds) International conference on bond graph modeling and simulation. Las Vegas, Nevada, pp 225–231
22. Margolis D, Karnopp D (1979) Analysis and simulation of planar mechanisms using bond graphs. ASME J Mech Des 101:187–191
23. Zeid A, Chung CH (1992) Bond graph modelling of multibody systems: a library of three-dimensional joints. J Franklin Inst 329:605–636
24. Borutzky W (1995) Representing discontinuities by sinks of fixed causality. In: Cellier FE, Granda JJ (eds) International conference on bond graph modeling and simulation. Las Vegas, Nevada, pp 65–72
25. Cellier FE, Otter M, Elmqvist H (1995) Bond graph modeling of variable structure systems. In: Cellier FE, Granda JJ (eds) 1995 International Conference on Bond Graph Modeling and Simulation. Las Vegas, Nevada, pp 49–55
26. Lorentz IF, Haffaf H (1995) Combination of discontinuities in bond graphs. In: Cellier FE, Granda JJ (eds) International conference on bond graph modeling and simulation. Las Vegas, Nevada, pp 56–64
27. Mosterman PJ, Biswas G (1998) A theory of discontinuities in physical systems models. J Franklin Inst 335B:401–439
28. Soderman U, Stromberg JE (1995) Switched bond graphs: towards systematic composition of computational models. In: Cellier FE, Granda JJ (eds) International conference on bond graph modeling and simulation. Las Vegas, Nevada, pp 73–79

http://www.springer.com/978-3-662-49002-0

Mechatronics by Bond Graphs
An Object-Oriented Approach to Modelling and Simulation
Damić, V.; Montgomery, J.
2015, XIX, 510 p. 439 illus. in color. With online files/update., Hardcover
ISBN: 978-3-662-49002-0