A software application for visualizing and understanding hydraulic and pneumatic networks

Tony Wong, Pascal Bigras
Department of Automated Manufacturing Engineering
École de technologie supérieure
University of Québec

Daniel Cervera
Department of Mechanical Engineering Technologies
Collège de Valleyfield

Corresponding author:
Tony Wong
Department of automated manufacturing engineering
École de technologie supérieure
University of Québec
1100 Notre-Dame West,
Montréal (Québec)
Canada, H3C 1K3
Tel: (514) 396-8930
Fax: (514) 396-8595
email: tony.wong@etsmtl.ca

Postal address of co-authors:

Pascal Bigras
Dept. of automated manufacturing engineering
École de technologie supérieure
University of Québec
1100, rue Notre-Dame ouest
Montréal (Québec)
Canada, H3C 1K3
email: pascal.bigras@etsmtl.ca

Daniel Cervera
Dept. of mechanical engineering technologies
Collège de Valleyfield
169, rue Champlain
Salaberry-de-Valleyfield (Québec)
Canada, J6T 1X6
email: daniel.cervera@colval.qc.ca

A software application for visualizing and understanding hydraulic and pneumatic networks

Abstract

Hydraulic and pneumatic networks are highly nonlinear and difficult to analyze. This paper presents a software application designed to help students visualize and understand fluid systems' dynamic behaviors. The application uses a combined bond graph and singular perturbation approach for system equation formulation. A standard iterative and adaptive integrator provides online numerical solutions to the system equations. Coupled to the integrator's output are a graphical animation subsystem and an instrumentation subsystem. The animation subsystem is responsible for rendering movable components on screen, at every simulation time-step, creating the illusion of continuous movement. The instrumentation subsystem collects and displays numerical data in numerical and graphical forms. An interesting contribution of this fluid system analyzer is its "user-in-the-loop" feature. This feature allows students to become active participants by enabling them to interact with network components while a simulation run is in progress.

Keywords: Modeling, simulation, nonlinear phenomena, hydraulic pneumatic network, bond graphs.

1. Introduction

Hydraulic and pneumatic networks are highly nonlinear and difficult to analyze. These systems exhibit nonlinear behavior because of restricted flows, finite displacements and non-negligible static and dynamic frictions. Pneumatic network nonlinearity is even stronger because of air compressibility. It is always a challenge to explain the effects of these nonlinear phenomena on the overall network behavior. This paper presents a software application designed to help students understand these complex phenomena without resorting to sophisticated finite element analysis.

In order to achieve this goal the proposed application translates the hydraulic and pneumatic networks into state-space form. The networks under study are obtained from a graphical editor that acts as the main user interface of the software application. The state-space translation step involves the use of a bond graph causality assignment technique to determine the proper input-output relationships of the fluid systems. Most hydraulic and pneumatic components do not have fixed input-output assignments and they have to be determined according to network topology. An adaptive numerical integrator then solves the resulting system equations. At constant time interval, the computed outputs are forwarded to the software application's animation and instrumentation subsystems. The animation subsystem is responsible for online graphical rendering of dynamic variables while the instrumentation subsystem is responsible for data collection and display.

To further enhance the visualization and understanding process, the software application also includes a special subsystem that permits in-simulation interactions. This important feature allows students to change the states of most fluid system components while a simulation is in progress. It is thus easy for instructors to design what-if scenarios and help

students to test their hypothesis and assumptions. Therefore, the students are not passive spectators but active participants in the learning process.

2. Problem identification

It has been noticed that students having learning difficulties can not explain or resolve apparently simple problematic situations. Even when the solution to these problems does not involve any calculation but simple qualitative description of the observed phenomenon. These students were unable to establish a correct relationship between: i) the situation confronting them; ii) the phenomenon observed; iii) the concepts implied. This inability to formulate a correct tripartite relationship is the result of misconception [1].

In Cervera [2] and in Youssef [3], the idea of misconception was explored in the field of fluid mechanics. Their studies involved groups of technical college students and first-year university engineering students. It was shown that most students exhibit confused reasoning when asked to explain simple observations. Below is an extract of some answers taken from Cervera's student survey [2]. Authors' commentary is shown in bold characters enclosed within parenthesis.

" Pressure is itself a material entity (**false**). A pump produces it (**false**). We can circulate, manipulate the pressure or use it to move a cylinder (**false**). Pressure is a function of the fluid's flow rate (**partially true**). The smaller the pipe the faster is the flow and the pressure is higher (**partially true**). Thus, a cylinder's strength depends on the size of the pipe (**false**). "

The students perceive the “pressure” not as an explanatory concept but as a real physical object. For them, pressure as a physical object, can be moved around within the system. And it is this physical object that acts on the cylinder’s piston and thus making it moves. Also, they consider a proportional relationship between the pressure and flow rate. By a similar deduction process, they conclude that the force produced by a cylinder is a function of the size of the connecting pipe.

2.1 Example of nonlinearities

We believe that one of the factors contributing to these misconceptions is the nonlinear relationship governing the pressure and the flow rate in fluid systems [4]. Students tend to relate quantities in linear or proportional terms. Their faulty representation is often the result of the general application of an idea, which states that linear behaviors can approximate nonlinear behaviors when the variations are small. As we will show, this is not always true in fluid systems even in very simple cases.

Consider a simple hydraulic setting with a single restriction and the fluid is assumed to be non compressible. This setting is illustrated in Figure 1.

[Take in Figure 1]

We shall neglect the effects of gravity and assume the velocity of the fluid at point 1 as negligible compared to that at point 2. Using Bernoulli’s equation, we have according to [5]

$$P_1 = P_2 + \tfrac{1}{2}\rho V_2^2 \qquad (1)$$

where P_i is the pressure at point i, V_i is the velocity of the fluid and ρ its density. If we consider that the opening surface at point 2 is A then the relation between the flow rate Q and the pressure difference at points 1 and 2 can be written as

$$Q = A\sqrt{\tfrac{2}{\rho}}\sqrt{P_1 - P_2}\,, \tag{2}$$

where $V = Q / A$. Evidently, this flow rate is not a linear function of the pressure. Moreover, simple linear approximation does not hold when the pressure difference is near zero. Consequently, most students will have difficulty conceiving the correct representation of this phenomenon.

We now turn the one dimensional fluid flow of Figure 1 into a pneumatic setting. The pneumatic setting is highly nonlinear because gas is compressible. We shall again neglect the effects of gravity. Again according to [5] and [6], the differential form of Bernoulli's equation gives

$$\frac{dP}{\rho} + \frac{1}{2}d(V^2) = 0\,, \tag{3}$$

where ρ is a function of the pressure P. If we assume an isentropic process for the fluid flow and apply the ideal gas hypothesis then we have

$$\rho = P^{\frac{1}{k}}C^{-\frac{1}{k}}\,, \tag{4}$$

$$\rho = \frac{P}{RT}\,, \tag{5}$$

where R is the gas constant, k is the specific heat coefficient ratio, T the temperature and C a constant. By replacing equation (4) into equation (3) and integrating between point 1 and point 2, we get

$$C^{\frac{1}{k}}\frac{k}{k-1}\left(P_2^{(k-1)/k} - P_1^{(k-1)/k}\right) + \frac{1}{2}\left(V_2^2 - V_1^2\right) = 0\,. \tag{6}$$

Again, we assume the velocity of the fluid at point 1 as negligible compared to that of point 2. Knowing that $V = Q / A$, the flow rate at point 2 can easily derived by combining equations (4), (5) and (6). The flow rate at point 2 is described by

$$Q_2 = A\sqrt{RT_1}\sqrt{\frac{2k}{k-1}}\sqrt{1-\left(\frac{P_2}{P_1}\right)^{(k-1)/k}} \tag{7}$$

which is a highly nonlinear function of P_1 and P_2. To further complicate the situation, the flow rate is different at point 1 and point 2 since gas is compressible. Most students will at first be confused by this phenomenon. To bypass this difficulty, we can instead use the mass flow $\dot{m}$. Because of mass conservation, mass flow is invariant along a pipe. The mass flow is related to fluid flow by

$$\dot{m} = \rho_2 Q_2 . \tag{8}$$

By combining equations (4), (5), (7) and (8),we can derive the following mass flow expression

$$\dot{m} = \frac{AP_1}{\sqrt{RT_1}}\sqrt{\frac{2k}{k-1}}\sqrt{\left(\frac{P_2}{P_1}\right)^{2/k} - \left(\frac{P_2}{P_1}\right)^{(k+1)/k}} . \tag{9}$$

In practical situations, we need to take into account an additional effect in the mass flow expression. It is the so-called diffusion phenomenon where the fluid velocity at point 2 cannot exceed the speed of sound [6]. When the fluid velocity at point 2 equals to that of sound, lowering the pressure at the restriction's output will not cause the mass flow to increase. With this added phenomenon, we have to rewrite equation (9) as

$$\dot{m} = \begin{cases} \dfrac{AP_1}{\sqrt{RT_1}}\sqrt{\dfrac{2k}{k-1}}\sqrt{\left(\dfrac{P_2}{P_1}\right)^{2/k} - \left(\dfrac{P_2}{P_1}\right)^{(k+1)/k}} & \text{if} \quad \dfrac{P_2}{P_1} < r_c, \\ \sqrt{k\left(\frac{2}{k+1}\right)^{(k+1)/(k-1)}} & \text{if} \quad \dfrac{P_2}{P_1} \geq r_c, \end{cases} \tag{10}$$

where

$$r_c = \left(\frac{2}{k+1} \right)^{k/(k-1)} . \quad (10)$$

Because of these cascading considerations, the final mass flow expression has the form of a piecewise function. Even in such a simple setting, fluid systems exhibit nonlinearity. In a more practical setting where basic fluid components are involved, the relationships must include other nonlinear phenomena. For example, fluid accumulation in a cylinder is a function of volume variations caused by rod displacement. This nonlinear relationship results in the coupling of the fluid part and the mechanical part of the system. Furthermore, the mechanical part also possesses nonlinear characteristics such as finite piston length, static and dynamic frictions. All these nonlinear characteristics and phenomena contribute to the overall complexity of most simple fluid systems.

3. Fluid system modeling

In order to obtain an effective software application for hydraulic and pneumatic network visualization and understanding, it is necessary to employ a systematic and efficient approach for its design and implementation. By modeling all hydraulic, pneumatic and mechanical components as multi-port devices, one can apply the bond graph theoretic approach for network topological recognition [7], [8]. The bond graph approach is a general technique that results in system equation formulation. It is well suited for hydraulic, pneumatic and general mechanical systems because of its so-called causality analysis capability [9].

In this design approach, all components are modeled as multi-port devices. There are two variables associated with each device port. They represent the effort-quantity (pressure, force, etc.) and the flow-quantity (flow, speed, etc.) of the device. However device port's

directional information is not known a priori. That is, input-output relationship of the device ports is dependent on the network's topology. By using bond graph's causality analysis technique, one can determine input-output relationship of every connected port online.

The goal of bond graphs is to represent a dynamic system by means of basic multi-port devices and their interconnections by bonds. These basic multi-ports exchange and modulate power through their bonds. There exist two power variables and two corresponding energy variables on each connected bond. They are: i) effort variable $e(t)$ – pressure (force); ii) flow variable $i(t)$ – flow rate (velocity); iii) momentum $\phi(t)$ – pressure momentum; iv) displacement variable $q(t)$ – volume. The effort variable is the time derivative of some momentum and conversely, the momentum is the time integral of an effort,

$$e(t) = d\phi(t)/dt, \quad \phi(t) = \phi_0 + \int e(t)dt. \tag{12a}$$

The same relationship applies to the flow variable and the displacement variable,

$$i(t) = dq(t)/dt, \quad q(t) = q_0 + \int i(t)dt. \tag{12b}$$

Graphically a bond is a directed line connected to two ports sharing the bond variables. The direction of the line shows the power flow between ports and is called the power half arrow. The direction of the power flow is chosen by convention and does not need to reflect the true polarity of the flow. Figure 2 illustrates the power flow between two multi-ports m_1 and m_2.

[Take in Figure 2]

We define a bond graph as $BG = <G, I>$. G is a directed and labeled graph and I a function identifying the multi-port type. We represent the graph G as a triplet $G = (M, B, \lambda_m)$, where

$M = \{m_1, \ldots, m_{|M|}\}$ is the set of multi-ports, $B = \{b_1, \ldots, b_{|B|}\}$ is the set of bonds and $|M|$, $|B|$ denotes the cardinality of the set M and B. For each bond $b \in B$, there is an ordered couple (m_i, m_j) joined by b which defines the power direction of the bond. While λ_m is the set of bonds incident to a multi-port $m \in M$. Also, we define $\lambda^+ : B \to M$, a function that returns the starting multi-port of a bond. Similarly, we define $\lambda^- : B \to M$ as a function returning the ending multi-port of a bond. We thus have

$$\lambda_m = \left\{ b \mid b \in B, \lambda^+(b) = m \right\} \cup \left\{ b \mid b \in B, \lambda^-(b) = m \right\}. \tag{13}$$

Finally, the identifying function I of the bond graph is simply $I : M \to T$, where $T \in$ {SF, SF, R, L, TF, GY, 0-junction, 1-junction} are the basic multi-port element labels.

[Take in Table 1]

Table 1 shows the set of basic multi-ports. The fourth column presents the constitutive laws of the basic multi-ports. A constitutive law determines the relationships of the associated variables for a given multi-port. The second column shows the usual power flow convention of the multi-ports. The third column presents the mandatory, constrained, preferred and indifferent computational causality of the multi-ports. The small stroke, called the causal stroke, at one end of a bond indicates the direction of travel of the effort variable information. The reaction to the effort information is the presence of a flow variable traveling in the opposite direction. Thus, the causal stroke represents the flow causality at one end of a bond. The opposite end of a bond must have complementary causality. This constitutes the fundamental causal constraint of a bond graph.

The 1-port sources S_e and S_f represent the interaction of a system with its environment. In fluid systems, they may represent pressure and flow sources. The 1-port resistive element *R*, capacitor element *C* and Inertia element *L* act as power dissipation and storage elements. The 2-port transformer and gyrator are power continuous elements (no power storage and no power dissipation). The *n*-port junction elements are also power continuous. The constitutive laws of a 0-junction are analogous to the Kirchoff's current law. The constitutive laws of a 1-junction are analogous to the Kirchoff's voltage law.

In bond graphs, the inputs and outputs are characterized by the effort causality and flow causality. Thus, causality assignment is a process by which the bond variables $e(t)$ and $i(t)$ are partitioned into input-output sets. There exist four types of causal constraints in bond graphs. *Mandatory causality* – The constitutive laws allow only one of the two port variables to be the output. Sources S_e and S_f have mandatory causality. *Preferred causality* – For the storage elements *C* and *L*, there can be time derivative causality or time integral causality. The preferred causality here refers to the integral causality of these elements. *Constrained causality* – For *TF*, *GY*, 0- and 1-junction there are relations between the causality of the different ports of the element. The relations are causal constraints because the causality of a particular port imposes the causality of the other ports. *Indifferent causality* – Indifferent causality means there is no causal constraints. The linear resistor element *R* exhibits indifferent causality since both power variables $e(t)$, $i(t)$ can be made member of the input and output sets.

Most traditional causality assignment procedures use a local constraint propagation scheme to label bond causality. From some starting point, usually one of the source elements, bond causality is assigned sequentially, according to the multi-port connected to the bond,

until all element ports are labeled. These causality assignment procedures must also satisfy the four causality types and the fundamental causal constraint.

The Sequential Causality Assignment procedure or SCAP is an example of causal labeling by local propagation [9]. The following pseudocode describes the causality assignment procedure.

```
Procedure SCAP : input(BG), output(BG)

P ← M
while (| P | > 0) {
    x ← m, where I(m) = SE ∨ I(m) = SF
    if ( {x} == ∅)
        x ← m, where I(m) = C ∨ I(m) = L
    if ( {x} == ∅)
        x ← m, where I(m) = R
    P = P – { m }
    assign(x, λx)
    If conflict(BG) abort
    If complete(BG) return BG
}
```

The assign(x, λ_x) procedure selects the appropriate mandatory, preferred or constrained causality (shown in Table 1) for multi-port x. It also attempts to assign causality to transformer, gyrator and n-port junction elements that are connected to x by using the set of bonds λ_x. After the assign procedure, a check is made to ensure that no causal conflict exists in the bond graph. Otherwise, the procedure will abort. Normally a causal conflict indicates the presence of topological loops (differential/algebraic loops), dependent storage elements or simply design errors. Such degenerated networks produce a set of differential algebraic equations and its solution is often time consuming. More details on causality assignment techniques can be found in [9], [10].

3.1 State-space formulation

For each causality, the numerical model of the *i*-th component is a nonlinear ODE system of form

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{h}_i(\mathbf{x}_i) + \mathbf{b}_i(\mathbf{x}_i, \mathbf{u}_i) \\ \mathbf{y}_i &= \mathbf{g}_i(\mathbf{x}_i) + \mathbf{d}_i(\mathbf{x}_i, \mathbf{u}_i) \end{aligned}, \tag{14}$$

where $\mathbf{x}_i$ is the state vector, $\mathbf{y}_i$ is the outputs vector and $\mathbf{u}_i$ is the inputs of the *i*-th component. Using the partitioning scheme given in [11] (Rosenberg 1971), it is possible to obtain a global state-space representation of the system. Figure 3 depicts this partitionning scheme graphically.

[Take in Figure 3]

In this scheme the set of transformer, gyrator and *n*-port junction elements are grouped together forming a so-called junction structure. The S_e and S_f elements are inputs to the junction structure. While the *C*, *L* and *R* elements have input-output relationships with the junction structure. Using the computational causality assigned to each element port, it is easy to determine the input bond variables and the output bond variables. We then write the set of equations representing the partitioned bond graph. For the resistive elements, we have

$$\mathbf{v} = \mathbf{R(w)}, \tag{15}$$

where **w**, **v** are the input and output variable vectors and **R** is the impedance functions vector. Similarly for the storage elements we have

$$\mathbf{z} = \mathbf{S(x)}, \tag{16}$$

where **S** is a function vector and **x** is called the energy variable vector or simply state vector. The set of equations governing the junction structure is

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \mathbf{J}_3 \\ \mathbf{J}_4 & \mathbf{J}_5 & \mathbf{J}_6 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{v} \\ \mathbf{u} \end{bmatrix}, \tag{17}$$

where $\mathbf{u}$ is the source vector and $\mathbf{J}_i$ are matrices representing the constraints imposed by the junction structures to the set of element ports. By substituting (15) and (16) into (17) we obtain

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{J}_1\mathbf{S}(\mathbf{x}) + \mathbf{J}_2\mathbf{R}(\mathbf{w}) + \mathbf{J}_3\mathbf{u}, \\ \mathbf{w} &= \mathbf{J}_4\mathbf{S}(\mathbf{x}) + \mathbf{J}_5\mathbf{R}(\mathbf{w}) + \mathbf{J}_6\mathbf{u}. \end{aligned} \tag{18}$$

Thus, without loss of generality, the system model comprising all network components is

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{H}(\mathbf{x}) + \mathbf{B}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} &= \mathbf{G}(\mathbf{x}) + \mathbf{D}(\mathbf{x},\mathbf{u}) \end{aligned}, \tag{19}$$

where $\mathbf{x}$ is the state vector, $\mathbf{y}$ is the outputs vector and $\mathbf{u}$ is the inputs vector of the system. The input-output coupling between network components can be modeled as

$$\mathbf{u} = \mathbf{C}\mathbf{y}\,. \tag{20}$$

Finally, the global model of all interconnected components is obtained by substituting equation (20) into equation (19). Since there is no algebraic loop, the model can then be written as follows:

$$\dot{\mathbf{x}} = \mathbf{H}(\mathbf{x}), \tag{21a}$$

$$\mathbf{y} = \mathbf{G}(\mathbf{x}). \tag{21b}$$

We can consider the system equations as an assembly of two coupled subsystems: one depicting the slow-varying part and the other depicting the fast-varying part of the system. In physical terms, the slow-varying part represents the mechanical interactions and the elasticity of the pneumatic fluid. Using similar reasoning, the fast-varying part represents the elasticity

of the hydraulic fluid. By applying the singular perturbation approach [12], the global model can then be rewritten as:

$$\dot{\mathbf{x}}_s = \mathbf{H}_s(\mathbf{x}_s, \mathbf{x}_f), \tag{22a}$$

$$\varepsilon\dot{\mathbf{x}}_f = \mathbf{H}_f(\mathbf{x}_s, \mathbf{x}_f), \tag{22b}$$

$$\mathbf{y} = \mathbf{G}(\mathbf{x}_s, \mathbf{x}_f). \tag{22c}$$

where $\mathbf{x}_s$ and $\mathbf{x}_f$ are the vectors of the state variables associated to the slow and fast subsystems and ε is a small positive parameter associated to the time constant of the fast subsystem. Since the parameter ε is close to zero, a good approximation solution of the slow subsystem can be obtained by posing $\varepsilon = 0$. The global model is then given by

$$\dot{\mathbf{x}}_s = \mathbf{H}_s(\mathbf{x}_s, \mathbf{x}_f), \tag{23a}$$

$$\mathbf{0} = \mathbf{H}_f(\mathbf{x}_s, \mathbf{x}_f), \tag{23b}$$

$$\mathbf{y} = \mathbf{G}(\mathbf{x}_s, \mathbf{x}_f). \tag{23c}$$

In this model, the set of nonlinear differential equations of the fast varying part is transformed into a set of nonlinear algebraic equations [12]. This result greatly facilitates the construction of the numerical integrator that can operate in quasi real-time. An adaptive integrator, part of the software application subsystems, is used to ensure the accuracy of the solution at each time-step. Note that the global model includes a static part (23b) and a dynamic part (23a). The integrator with adaptive time-steps is based on multiple evaluations of the model. At each model evaluation, the static part must be solved. This approach allows us to have a larger time-step since only the slow subsystem must be integrated. However, because the latter is iterative in nature, it is not possible to guarantee hard real-time performance.

4 Design and implementation

In the proposed software application, the process of construction, simulation and data display is a set of simple tasks. In the construction task, hydraulic, pneumatic and mechanical components are selected from the component palettes. Using hydraulic (pneumatic) links and connecting them to component ports, one can create component interconnections to form a network. Each network component contains a numerical model whose parameters are adjustable via the user interface. A student can accept the default values, use specific settings given by instructor or specify values directly from manufacturer's data sheets. For the simulation task, numerical simulation is performed in either continuous mode or step-by-step mode. In continuous mode, system equations solution and graphical animation updates are continuously executed every time-step. In step-by-step mode, the student is responsible for time-step increment (by depressing a key or a mouse click) so that it is possible to slow down or speed up the simulation process. Finally, the data display task consists of data collection using measurement components. Data are displayed synchronously with the simulator output in both numerical and graphical forms. Data collected are persistent until the next simulation run. The measurement components can also perform formatted storage operations. This enables extensive post-simulation analysis using external data analysis tools.

The above-described tasks are best realized by the use of object-oriented technologies. In the object-oriented approach, a software application is the result of a collection of cooperating objects [13], [14]. We obtain the objects for the software application by analyzing its specification along a responsibility assignment point of view [15]. Thus, all

objects must have nontrivial responsibilities assigned to them. Otherwise they are discarded from the design.

4.1 In-simulation interactions

As shown in Fig. 4, there is a logical interface between the integrator output and the graphical animation subsystem. The latter subsystem is responsible for updating color changes and the rendering of moving parts on behalf of the network components. The graphical animation subsystem also collaborates closely with network components and the network editor to allow in-simulation interactions. An in-simulation interaction permits a student to change or modify a component's state while a simulation run is in progress. Thus, the student can stop or start motors, select different distributor pistons, modify cylinder's longitudinal position and so on.

[Take in Figure 4]

In order to allow in-simulation interactions, the application establishes a so-called mechanical chain to determine the effect of changes on other network components. This involves the recalculation of network states and their propagation to all connected components by a graph traversal. To implement in-simulation interactions every network component that possesses user-interactivity must register itself with an object called In-simulation Observer. During a simulation run, all mouse and keyboard events (or simply user events) are filtered and routed to the In-simulation Observer. The Observer passes along the user events to the registered components. Consequently, it is the network components that

are responsible for user event processing. Figure 5 shows the object diagram of the in-simulation interaction subsystem. It is actually an implementation of the classical object-oriented Observer pattern [15].

[Take in Figure 5]

A network component is programmed to recognize only meaningful events. For example, dragging the body of a hydraulic pump during a simulation run has no meaning. However, dragging its displacement control will signal the pump to update its displacement value. After a successful event processing by a registered component, the In-simulation Observer will inform the Integrator, via the Editor, to propagate state changes to other connected components. The simulation then restarts at the same point in time using the newly recalculated state variables.

4.2 Instrumentation

It is worth noting that instrumentation (i.e. adding measurement points to the system) does not increase system complexity. Since instrumentation merely associates a set of system variables to the set of output variables, there is no extra overhead in processing instrumentation measurement points. In the software application, the grapher object is responsible for online plotting of instrumented network variables. Finally, instrumentation is preprocessed in the causality analysis phase so that no negative impact can affect the overall simulation time.

The Animator object uses a color-coded scheme to represent state changes in transient regime and transient to steady state transitions. Usually, mechanical displacements (linear translations and rotational movement) can be expressed easily by graphical animations. However, dynamic entities such as pressure differential, flow resistance and temperature variations are much more difficult to express. A color-coded scheme is suitable for representing these dynamic variables. In this scheme a set of colors is mapped to a range of variable values. While a simulation run is in progress, the student can notice component internal changes by observing its color variations. For a more quantitative evaluation, the student can also connect measurement components to the network. These measurement components display or plot numerical data at each time-step.

5 Applications

The fluid system construction makes use of a set of component palettes. Some of these component palettes are shown in Fig. 6.

[Take in Figure 6]

By using the “drag and drop” technique, the student can lay out components on the workspace. In order to create a useful hydraulic or pneumatic system, one has to connect together the appropriate component ports. This task is shown in Figure 7.

[Take in Figure 7]

A typical two-speed hydraulic network provides different forward and backward cylinder speeds. The user can control cylinder's forward and backward motion by activating the distributor pistons. This typical network is shown in Fig. 8.

[Take in Figure 8]

Also in Figure 8 is a grapher object, which shows the cylinder position as a function of simulation time-steps.

The simulator also allows hydraulic and pneumatic system coupling at mechanical contact points. This capability permits the study of different system dynamics and their interplay. In Figure 9, the mechanical contact point is the interface between the pneumatic and hydraulic cylinders. This hybrid network shows an interesting application where a pneumatic cylinder, driven by its compressor group, is the prime moving force.

[Take in Figure 9]

The hydraulic cylinder, which is coupled to the pneumatic one, acts as a retaining force on behalf of the load. Note that this hybrid configuration offers different retaining force for forward and backward cylinder motions since there is two flow controls installed on the hydraulic cylinder. Again, the student can control forward or backward motion by activating the distributor pistons or by dragging one of the cylinder's pistons.

As indicated earlier every hydraulic, pneumatic and mechanical component possesses a numerical model that is fully adjustable. Figure 10 shows a typical component dialog box for

parameter input. In most cases, non-ideal component characteristics are possible. For example, Figure 10 depicts an input dialog box for a double acting cylinder. The student can select from several joint leak degrees (negligible, low, medium and high) to obtain non-ideal characteristics caused by aging and wears of the component.

[Take in Figure 10]

6 Conclusions

This paper presented the motivation and the design of a software application for visualizing and understanding hydraulic and pneumatic networks. The intent is to help students analyze fluid system behaviors. This application is suitable for small-to-medium scale fluid system analysis. It uses a combined bond graph and singular perturbation approach for system equation formulation. A standard adaptive integrator solves the system equations that are divided into a slow-varying part and a fast-varying part. The logical interface of the integrator output and the graphical animation and instrumentation subsystems are also detailed in the paper. The graphical animation subsystem is responsible for updating color changes and rendering of moving parts on behalf of network components. An important feature of this computer application is its ability to permit in-simulation interactions. It allows the students to manipulate network components while a simulation run is in progress. This feature should facilitate hydraulic and pneumatic network understanding by putting the students in the simulation loop. Finally, this software application is in use in number of universities and technical colleges throughout Québec. It has been awarded the Québec Education Minister's Award for "best educational software" in the year 2000.

References

[1] D. Cervera, Élaboration d'un environnement d'expérimentation en simulation, incluant un cadre théorique pour l'apprentissage de l'énergie des fluides, PhD dissertation, faculté des sciences de l'éducation, Université de Montréal, 1998.

[2] D. Cervera and A. Métioui, Énergie des fluides: analyse conceptuelle et représentation des élèves, technical report (PAREA), *Collège de Valleyfield*, Québec, 1993.

[3] A. Youssef, A. Métioui, P. Bigras and D. Cervera, Les représentations des étudiants du collégial professionnel à l'égard des principes de fonctionnement des circuits hydrauliques, technical report, École de technologie supérieure, University of Québec, 1991.

[4] T. Wong, P. Bigras, and D. Cervera, A software tool for understanding nonlinear phenomena in hydraulic and pneumatic systems, Proceedings of the 2002 American Society of Engineering Education Conference, session 2220, paper 514, 2002.

[5] R. R. Munson, D. F. Young, T. H. Okiishi, Fundamentals of Fluid Mechanics, John Wiley & Sons, New York, 1998.

[6] B. W. Andersen, The Analysis and Design of Pneumatic Systems, John Wiley & Sons, New York, 1967.

[7] A. Mukherjee and R. Karmakar, Modeling and Simulation of Engineering Systems Through Bondgraphs, CRC Press LLC, Boca Raton, 2000.

[8] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, System Dynamics: A unified Approach, John Wiley & Sons, New York, 1990.

[9] J. D. Dijk, On the role of bond graph causality in modelling mechatronic systems, PhD dissertation, University of Twente, CIP-Gegevens, 1994.

[10] T. Wong, P. Bigras, K. Khayati, Causality assignment using multi-objective evolutionary algorithms, IEEE International Conference on Systems, Man and Cybernetics, Vol: 4 , 2002, pp. 36 –41.

[11] R. C. Rosenberg, State-Space Formulation for Bond Graph Models of Multi-Port Systems, Transactions of the ASME, Journal of Dynamic Systems Measurement and Control, 93(2), pp.123-125, 1971.

[12] H. K. Khalil, Nonlinear Systems, Macmillan, New York, 1992.

[13] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, Object-Oriented Modeling and Design, Prentice Hall, New Jersey, 1991.

[14] R. Allen, A formal approach to software architecture. PhD dissertation, Carnegie Mellon University, 1997.

[15] A. Shalloway, and R. Trott, Design Patterns Explained: A New Perspective on Object-Oriented Design, Addison-Wesley, New York, 2001.

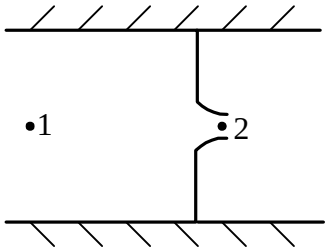


Figure 1. One dimensional fluid flow with single restriction.

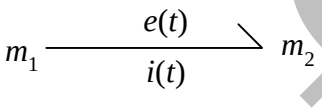


Figure 2. A bond connecting two multi-ports m_1 and m_2. The half arrow indicates that power flows from m_1 to m_2.

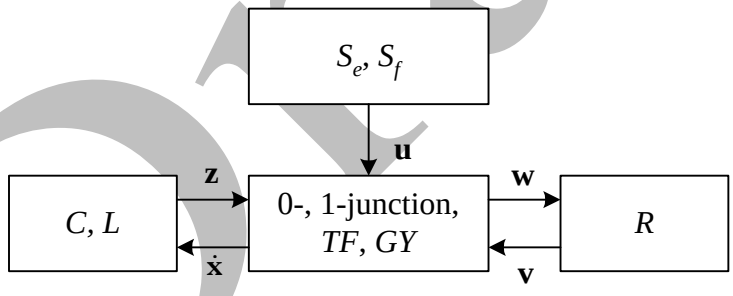


Figure 3. Partition of a bond graph based on the input-output sets of multi-port elements.

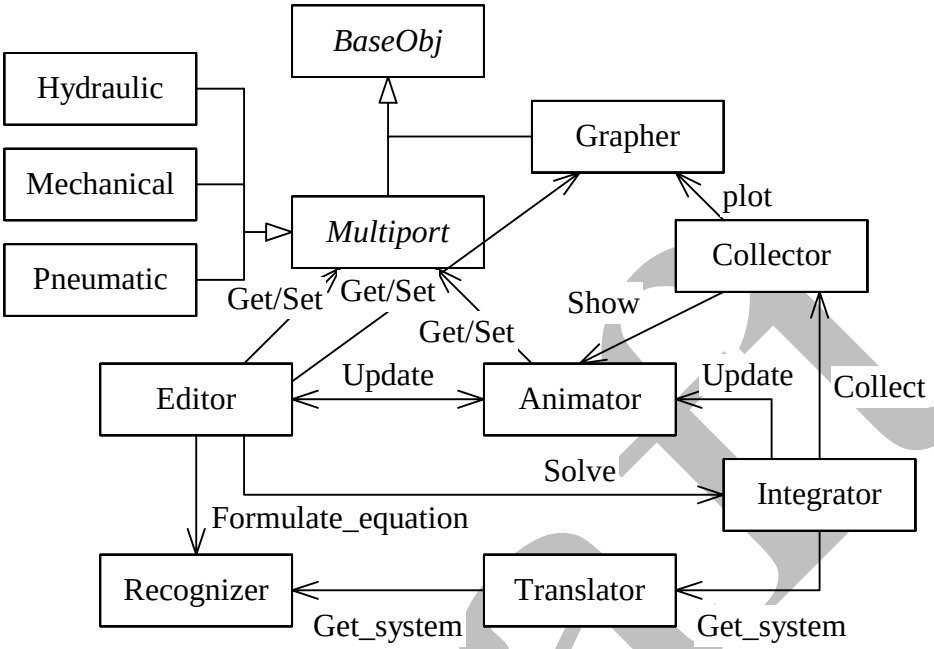


Figure 4. Simplified application objects relationships.

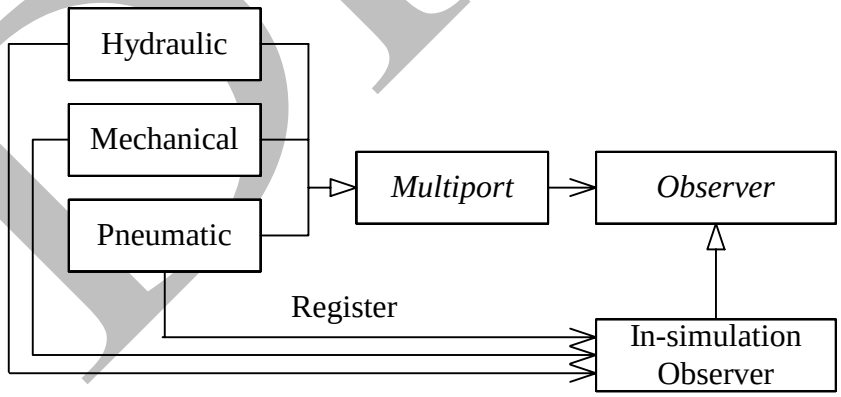


Figure 5. In-simulation interactions modeling using the Observer pattern.

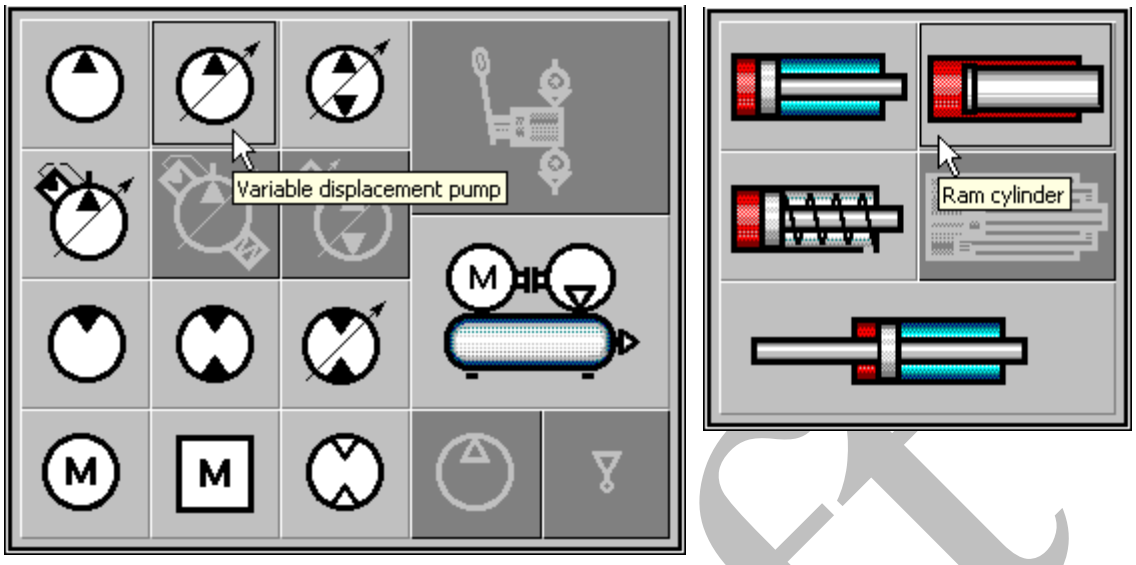


Figure 6. Two component palettes of the simulation package.

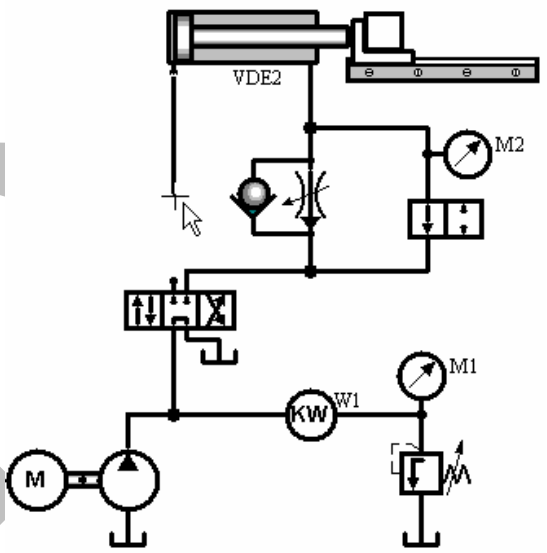


Figure 7. Components are interconnected by the use of links.

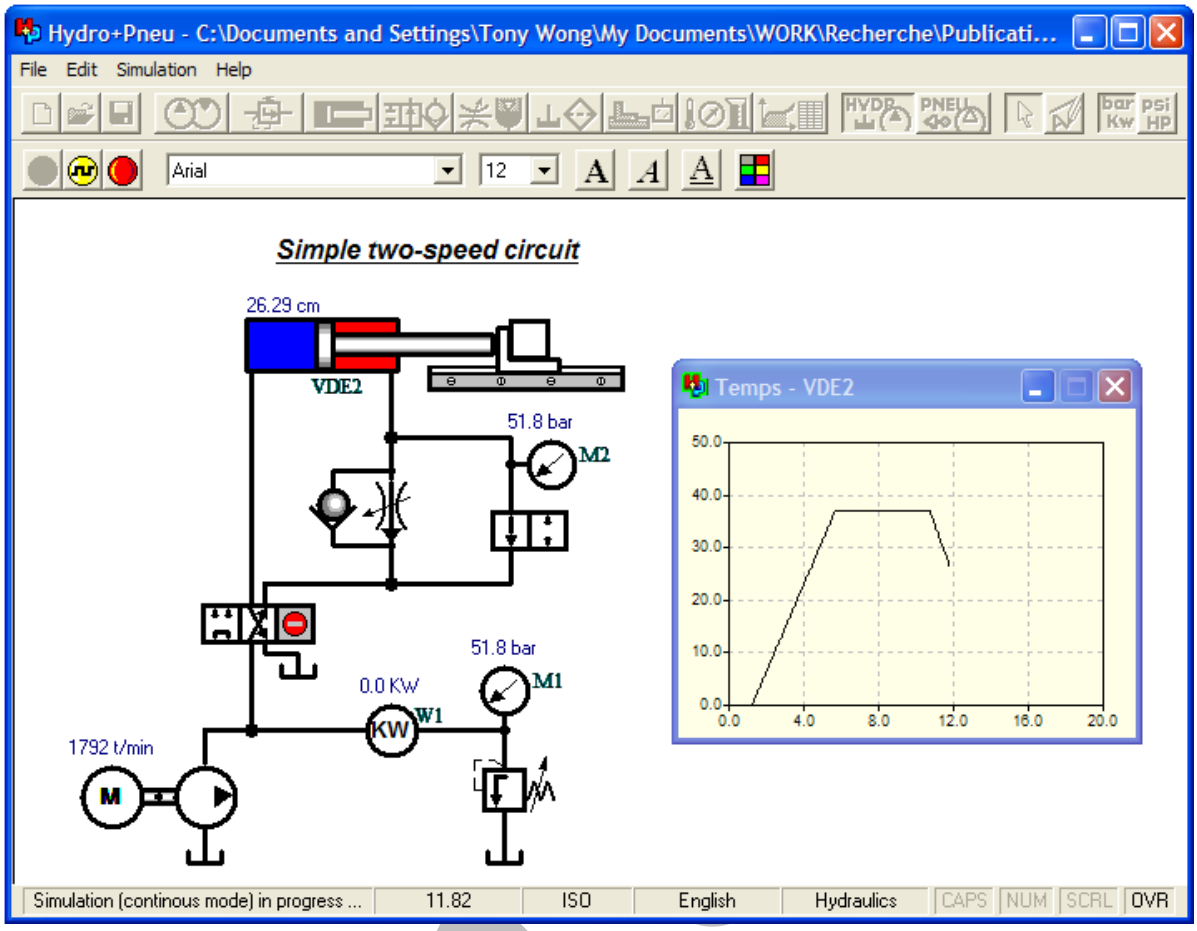


Figure. 8. Typical two-speed hydraulic system with instrumentation and grapher output.

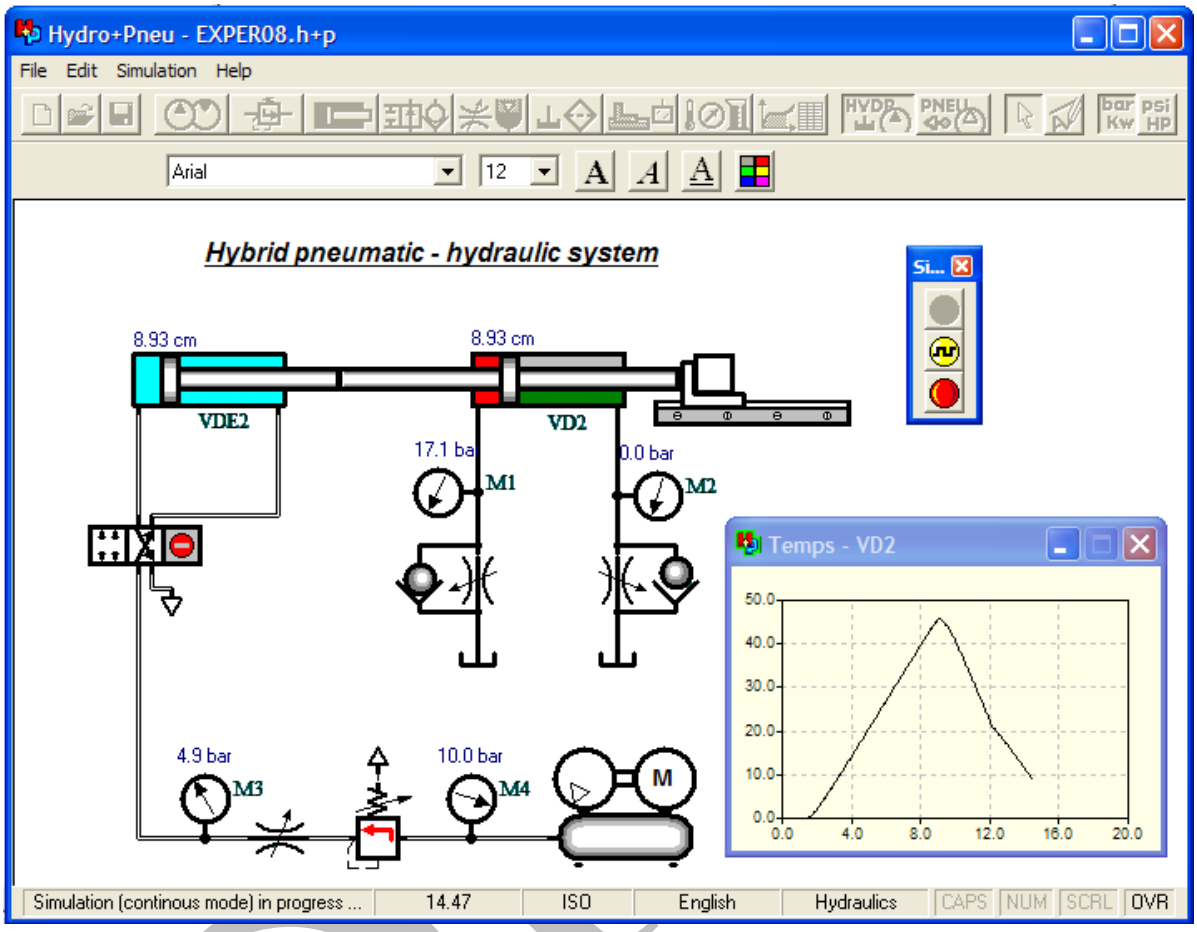


Figure 9. Hybrid hydraulic – pneumatic system with compressor group.

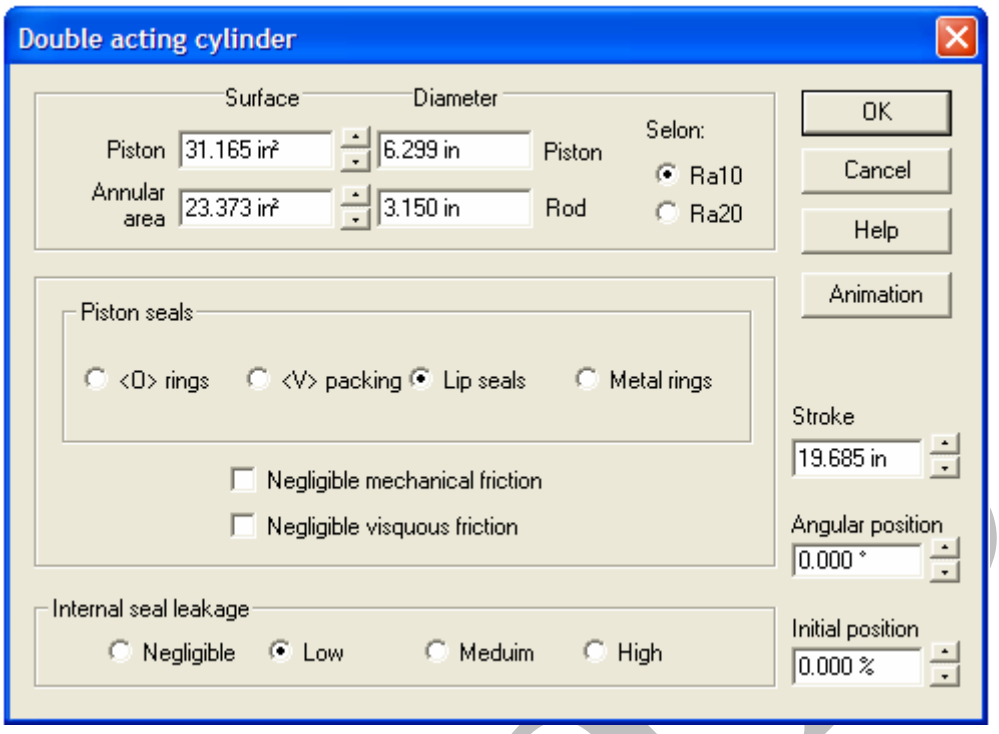
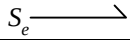
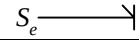
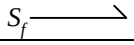
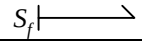
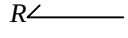
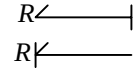
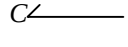
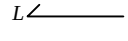
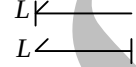
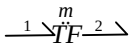
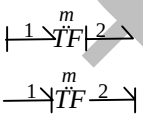
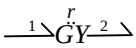
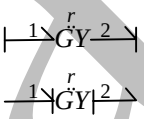
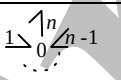
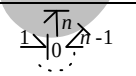
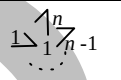
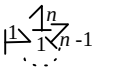


Fig. 10. Numerical model parameters for a double acting cylinder.

Table 1 Bond graph's Basic Multi-ports Elements

Multi-port Name	Powerflow	Computational Causality	Constitutive laws
Effort source			$e(t) = E(t)$
Flow source			$i(t) = I(t)$
Resistor			$e(t) = f_R i(t),$ $i(t) = f_R^{-1} e(t).$
Capacitor			$e(t) = f_C^{-1} \int i(t)dt,$ $i(t) = f_C\, de(t)/dt.$
Inertia			$i(t) = f_L^{-1} \int e(t)dt,$ $e(t) = f_L\, di(t)/dt.$
Transformer			$e_1 = m\, e_2,$ $i_2 = m\, i_1.$ $i_1 = m^{-1} i_2,$ $e_2 = m^{-1} e_1.$
Gyrator			$e_1 = r\, i_2,$ $e_2 = r\, i_1.$ $i_1 = r^{-1} e_2,$ $i_2 = r^{-1} e_1.$
0-junction			$e_1 = \ldots = e_{n-1} = e_n,$ $i_1 + \cdots + i_{n-1} + i_n = 0.$
1-junction			$i_1 = \ldots = i_{n-1} = i_n,$ $e_1 + \cdots + e_{n-1} + e_n = 0.$