MT360 - méthodes numériques & modèles déterministes

Chapitre 4 : modélisation des systèmes multiphysiques

L'approche Bond Graphs

Laurent Lefèvre

laurent.lefevre@lcis.grenoble-inp.fr

Laboratoire de Conception et d'Intégration des Systèmes (LCIS)
Grenoble Institute of Technology (Grenoble INP - Esisar)

Avril, 2025

Models classification for dynamical systems

- **No global methodology**
- **No model built independently of some problem/context**

Examples of classification

- Probalistic / **deterministic**
- Event driven / continuous (sampled) time **evolution dynamics**
- Microscopic / **macroscopic**
- Black box / grey box / **knowledge-based**

A model should be made as simple as possible ... but not simpler!

Dynamical systems?

What is a system?

- an entity separable from the rest of the universe (**environment**)
- physical or conceptual **boundary** with the environment
- composed of interacting parts (**subsystems** or **components**)
- **more than the sum** of components behaviours

What means dynamical?

- the variation of the behaviour as a **function of time** is important
- time **transients effects** are important
- static or **steady-state analysis is not sufficient** to deal with the problem (e.g. non minimum phase systems, unstabilities, etc.)

Multi-physics macroscopic systems network theory (thermodynamics)

- **macroscopic** intensive/extensive variables

$$(P, V), (T, S), (v, p), (U, q), \ldots$$

- **First principle:** the system energy (among others)

$$H(t) := H(V(t), S(t), p(t), q(t), \ldots)$$

is a first order homogeneous state function of the extensive variables:

$$\frac{dH}{dt} := -P\frac{dV}{dt} + T\frac{dS}{dt} + v\frac{dp}{dt} + U\frac{dq}{dt} + \ldots$$

- **Second principle:** positive-entropy production principle (irreversibility)

$$\dot{S}(t) \geq 0$$

- **Circuit-like** approach inspired from electro-mechanical systems

Thermodynamics and mechatronics

Physical interactions

- **power exchanges** between subsystems and the environment (system topology)
- subsystems (components) are **supplying**, **dissipating** or **storing** energy in various forms (energy domains)
- within the system, any interaction is described using **power conjugated variables**

Signal/information exchanges

- do not imply power exchange, neither bidirectional interaction
- are necessary to represent mechatronic / control systems
- may cause loss of self-consistance

Reticulation process

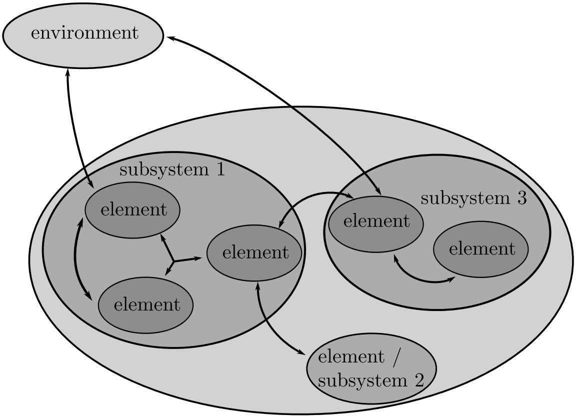


Figure: Recursive modelling process

State space systems and control

State-determined system

- set of ODEs in the **state variables**
- set of algebraic equations that relate system variables
- **causal** system with the whole history summarized in the state value
- the future is determined by the current state value and the future inputs!

Uses of dynamical models

- Analysis
- Identitification
- Synthesis/design
- Control

Contents

1. 1: the electro-mechanical analogy
2. 2: Bond graphs
3. 3: Bond graph algorithms

References: see *chamilo.grenoble-inp.fr*, MT360

- **System Dynamics: Modeling, Simulation and Control of Mechatronics Systems**, Dean C. Karnopp, Donald L. Margolis and Ronald C. Rosenberg, Wiley, Hoboken, 2012 (5th edition)

  $\rightarrow$ Chapters 1 to 5

- **Port-Hamiltonian Systems Theory: An Introductory Overview**, Arjan van der Schaft and Dimitri Jeltsema, Foundations and Trends in Systems and Control, Vol. 1, N° 2-3, pp. 173-378, 2014

  $\rightarrow$ Chapters 1, 2, 4, 5, 7, 10 and 15

- **Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations**, Ernst Hairer, Christian Lubich and Gerhard Wanner, Springer, Berlin, 2006 (2nd edition)

  $\rightarrow$ Chapters 1, 2 and 6

4.1: The electro-mechanical analogy

Outline

1. Stored energy and state variables
2. Co-energy variables
3. System topology and power balance equation
4. Storage constitutive equations
5. Dissipation constitutive equations
6. Source constitutive equations
7. Conclusion

Mechanical and electrical circuits (1)

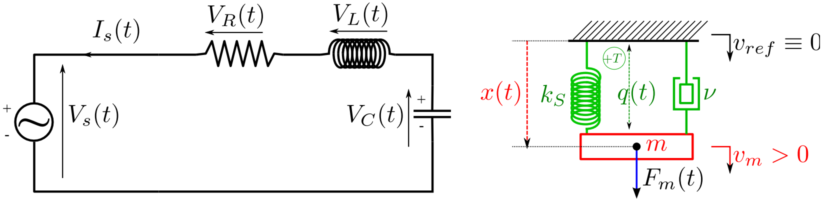


Figure: serial RLC circuit and equivalent Mass-Spring system

Stored energy and state variables

$$H(q,p) := \frac{q^2(t)}{2C} + \frac{p^2(t)}{2L}$$

Extensive energy state variables:

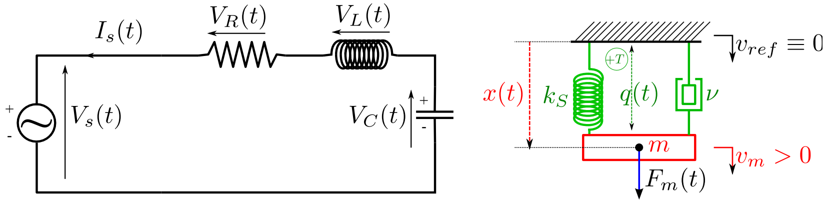
$$q(t) := Q(t)$$
$$p(t) := \phi(t)$$

$$H(q,p) := \frac{q^2(t)}{2\frac{1}{k_S}} + \frac{p^2(t)}{2m}$$

Extensive energy state variables:

$$q(t) := q(t)$$
$$p(t) := m\dot{x}(t)$$

Mechanical and electrical circuits (2)



Co-energy variables

$$\frac{dH}{dt} := \frac{\partial H}{\partial q}\dot{q} + \frac{\partial H}{\partial p}\dot{p} = e_q(t)f_q(t) + e_p(t)f_p(t)$$

with the efforts/flows:

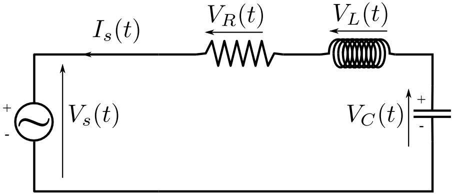
$$e_q := \frac{\partial H}{\partial q} = \frac{Q}{C} = V_C \quad f_q := \dot{Q} = I_C$$
$$e_p := \frac{\partial H}{\partial p} = \frac{\phi}{L} = I_L \quad f_p := \dot{\phi} = V_L$$

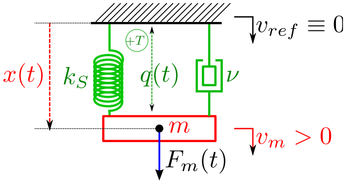
$$\frac{dH}{dt} := \frac{\partial H}{\partial q}\dot{q} + \frac{\partial H}{\partial p}\dot{p} = e_q(t)f_q(t) + e_p(t)f_p(t)$$

with the efforts/flows:

$$e_q := \frac{\partial H}{\partial q} = k_S q = F_S \quad f_q := \dot{q}$$
$$e_p := \frac{\partial H}{\partial p} = \frac{p}{m} = \dot{x} \quad f_p := \dot{p} = F_m$$

Mechanical and electrical circuits (3)





System topology

Kirchoff's laws

$$
\begin{aligned}
I_s &= I_R = I_L = I_C \\
V_s &= (V_R + V_L + V_C)
\end{aligned}
$$

$\Rightarrow$ power balance:

$$
\frac{dH}{dt} = I_S V_S - \underbrace{V_R I_R}_{\geq 0} = P_{ext}(t) - RI^2
$$

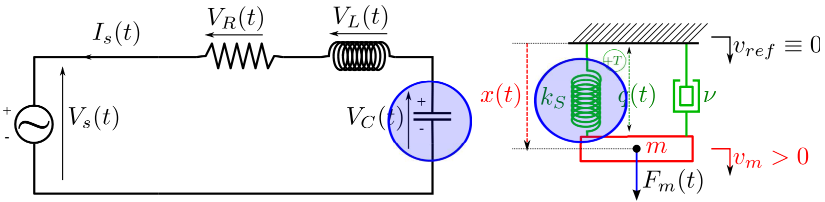
Kinematic constraints

$$
\begin{aligned}
\dot{q}_S &= \dot{q}_\nu = \dot{x} = \dot{x}_F \\
F_{ext} &= (F_m - F_\nu - F_S)
\end{aligned}
$$

$\Rightarrow$ power balance:

$$
\begin{aligned}
\frac{dH}{dt} &= F_S \dot{q} + F_m v_m = F_{ext} v_m - \nu \dot{q} \dot{q} \\
&= P_{ext}(t) - \nu \dot{q}^2
\end{aligned}
$$

Mechanical and electrical circuits (4)



Storage elements (1/2)

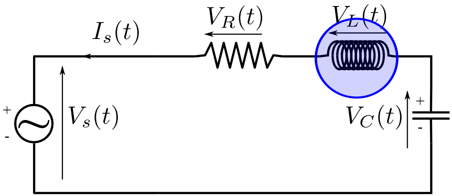
Capacitor constitutive equation

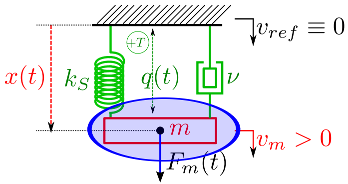
$$
\begin{aligned}
\dot{Q} &= I_C \\
V_C &= \frac{\partial}{\partial Q} H_C(Q) \\
&= \frac{\partial}{\partial Q}\left[\frac{Q^2}{2C}\right] \\
&= \frac{Q}{C}
\end{aligned}
$$

Spring constitutive equation

$$
\begin{aligned}
\dot{q} &= v_S \text{ (spring relative velocity)} \\
F_S &= \frac{\partial}{\partial q} H_S(q) \\
&= \frac{\partial}{\partial q}\left[\frac{k_S q^2}{2}\right] \\
&= k_S q \text{ (spring restoring force)}
\end{aligned}
$$

Mechanical and electrical circuits (5)





Storage elements (2/2)

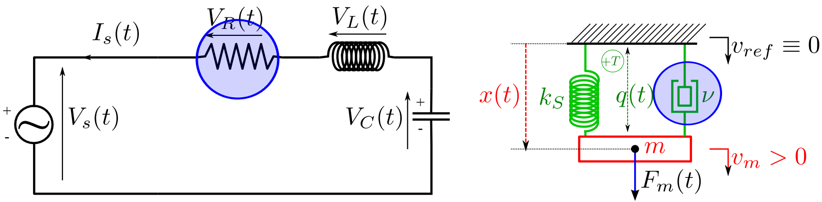
Inductor constitutive equation

$$\begin{aligned}
\dot{\phi} &= V_L \\
I_L &= \frac{\partial}{\partial \phi} H_L(\phi) \\
&= \frac{\partial}{\partial \phi}\left[\frac{\phi^2}{2L}\right] \\
&= \frac{\phi}{L}
\end{aligned}$$

Inductor constitutive equation

$$\begin{aligned}
\dot{p} &= F \text{ (Newton's law)} \\
v_m &= \frac{\partial}{\partial p} H_L(p) \\
&= \frac{\partial}{\partial p}\left[\frac{p^2}{2m}\right] \\
&= \frac{p}{m} \text{ mass absolute velocity}
\end{aligned}$$

Mechanical and electrical circuits (6)



Dissipation elements

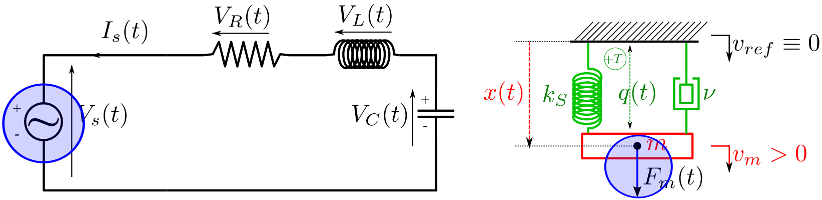
Resistor constitutive equation

$$\begin{aligned} f_R &= I_R \\ e_R &= V_R \\ &= R \cdot f_R \\ &\text{ with } R \geq 0 \end{aligned}$$

Dashpot constitutive equation (linear viscous damping)

$$\begin{aligned} f_d &= \dot{q} \\ e_d &= F_\nu \text{ (friction forces)} \\ &= \nu \cdot f_d \\ &\text{ with } \nu \geq 0 \end{aligned}$$

Mechanical and electrical circuits (7)



Sources

Voltage source constitutive equation

$$\begin{aligned} e_s &= V_s(t) \text{ (input)} \\ f_s &= I_s(t) \end{aligned}$$

The supplied power is:

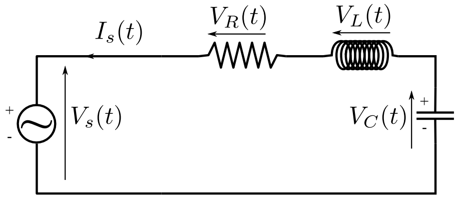
$$P_s(t) = e_s \cdot f_s$$

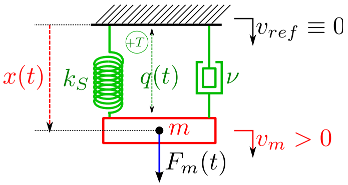
External force constitutive equation

$$\begin{aligned} e_s &= F_m(t) \text{ (input)} \\ f_s &= v_m(t) \end{aligned}$$

The supplied power is:

$$P_s(t) = e_s \cdot f_s$$





State space models: using constitutive and structure (topology) equations, one gets:

$$\begin{bmatrix} \dot{Q} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{L} \\ \frac{-1}{C} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} Q \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ V_s \end{bmatrix} \qquad \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} \\ -k_S & \frac{-\nu}{m} \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ F_m \end{bmatrix}$$

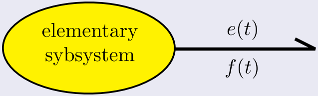
- The **lumped parameters** assumption, **constitutive equations** (from electromagnetism or mechanics) and **interconnection equations** result in a **state-space system** of ODEs
- This state space system may be **implicit** and **non linear**

Section 2: Bond Graphs

Outline

1. Effort and flow variables in different energy domains
2. Energy Conservative elements
3. Dissipative elements
4. Ideal sources
5. Junction structures
6. Conclusion

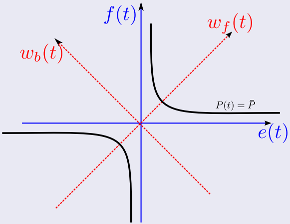
Co-energy variables: efforts and flows



Power flow:

$$P(t) = e(t)f(t)$$

Co-energy variables vs. scattering variables



Scattering (wave) variables:

$$\begin{bmatrix} w_f \\ w_b \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix}$$

Power quadratic form:

$$P(t) = e(t)f(t) = \frac{w_f^2}{2} - \frac{w_b^2}{2}$$

Energy domain	State x	flow $f_x := \dot{x}$	effort $e_x := \frac{\partial H}{\partial x} = \delta_x H(x)$
Potential Mechanics translation	Displacement, q	Velocity $v(t) := \dot{q}(t)$	Force $F := \delta_q H(q)$
Kinetic Mechanics translation	Momentum, p	Force $F(t) := \dot{p}(t)$	Velocity $v := \delta_p H(p)$
Potential Mechanics rotation	Angle, θ	Angular velocity $\omega(t) := \dot{\theta}(t)$	Torque $\tau := \delta_\theta H(\theta)$
Kinetic Mechanics rotation	Angular momentum, $p = I\omega$	Torque $\tau(t) := I\dot{\omega}(t)$	Angular velocity $\omega := \delta_p H(p)$
Electric	Charge, Q	Current $I(t) := \dot{Q}(t)$	Voltage $U := \delta_Q H(Q)$
Magnetic	Flux, ϕ	Voltage $U(t) := \dot{\phi}(t)$	Current $I := \delta_\phi H(\phi)$
Potential Hydraulics	Volume, V	Flow rate $Q(t) := \dot{V}(t)$	Pressure $P := \delta_V H(V)$
Kinetic Hydraulics	Pressure momentum, p_P	Pressure $P(t) := \dot{p_P}(t)$	Flow rate $\dot{V} := \delta_{p_P} H(p_P)$
Thermal	Entropy, S	Entropy flow rate $\dot{S}(t)$	Temperature $T := \delta_S H(S)$
Chemical	Number of moles, N	Molar flow rate $\dot{N}(t)$	Chemical potential $\mu := \delta_N H(N)$

Table: Energy and co-energy variables in different energy domains

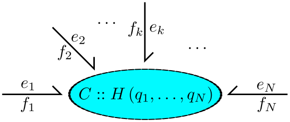
C-field or N-ports capacitor-like element

- store and “restore” energy without losses
- first principle element:

$$H = H\left(q_1, \dots, q_N\right)$$

- the power exchanged through the ports may be written

$$P(t) = \frac{dH}{dt} = \sum_{i=1}^{N} \frac{\partial H}{\partial q_i} \cdot \dot{q}_i$$



$$\begin{aligned} e_i(t) &:= \frac{\partial H}{\partial q_i} = e_i\left(q_1, \dots, q_N\right) \\ f_i(t) &:= \frac{dq_i}{dt}(t) \end{aligned}$$

C-field defined with the effort functions

$$\begin{aligned} e_i(t) &= e_i\left(q_1, \ldots, q_N\right) \\ f_i(t) &= \frac{dq_i}{dt}(t) \end{aligned}$$

The efforts derive from an energy function H:

$$\frac{\partial e_j}{\partial q_i} = \frac{\partial e_i}{\partial q_j} \Leftrightarrow \frac{\partial^2 H}{\partial q_i \partial q_j} = \frac{\partial^2 H}{\partial q_j \partial q_i}$$

L-field

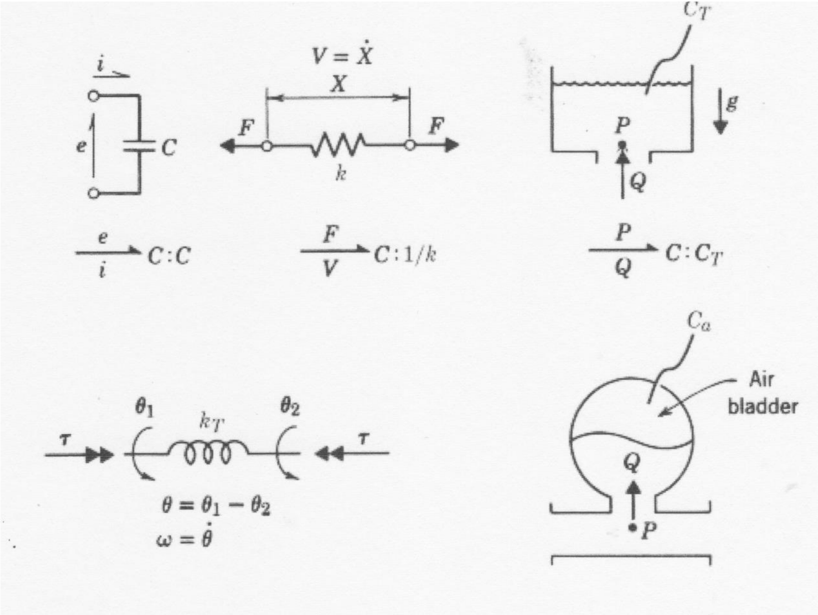
The energy is given by

$$H = H\left(p_1, \ldots, p_N\right)$$

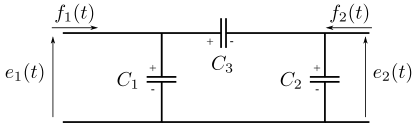
and

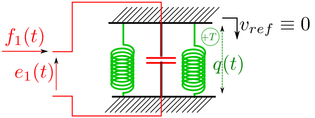
$$\begin{aligned} f_i(t) &:= \frac{\partial H}{\partial p_i} = f_i\left(p_1, \ldots, p_N\right) \\ e_i(t) &:= \frac{dp_i}{dt}(t) \end{aligned}$$

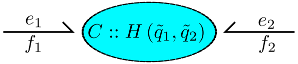
Energy conservative elements

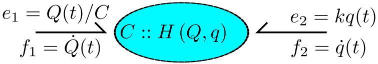


Energy conservative elements









R-field or N-ports resitor-like element

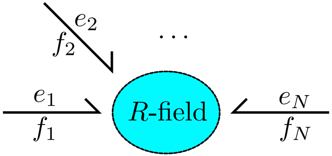
- Convert energy into heat (irreversible entropy production)
- Second principle element:

$$P(t) = \sum_{i=1}^{N} e_i \cdot f_i = T\frac{dS}{dt} \geq 0$$

- no internal dynamics:

$$\mathbf{F}(\mathbf{e}, \mathbf{f}) = 0$$

where $\mathbf{e}, \mathbf{f} \in \mathbb{R}^N$ denote respectively the effort and flow vectors and where the implicit function theorem applies



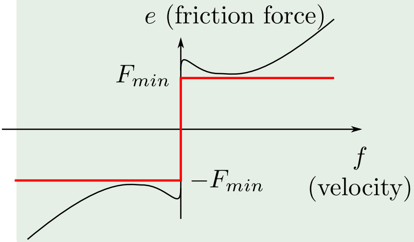
Example (R-field)

$$\mathbf{e} = \mathbf{R}(\mathbf{f})\,\mathbf{f}$$

with the matrix $R(\mathbf{f})$ positive semi definite:

$$\langle \mathbf{f} \,|\mathbf{R}(\mathbf{f})\,\mathbf{f}\rangle_{\mathbb{R}^N} = \mathbf{f}^T\mathbf{R}(\mathbf{f})\,\mathbf{f} \geq 0 \ , \ \forall \mathbf{f} \in \mathbb{R}^N$$

Example (Coulomb friction)



Implicit definition:

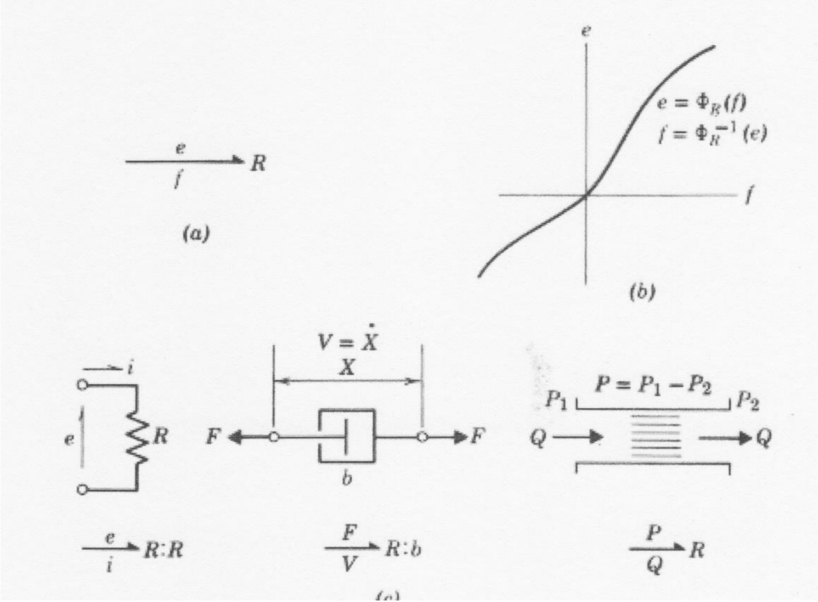
If	$\lvert e \rvert \leq F_{min}$ then $f = 0$
Else	$e = e(f)$
	with $f \cdot e(f) \geq 0$

RS-field notation

A resistor may be viewed as a power preserving transducer:

$$P(t) = e(t)f(t) - T\dot{S} \equiv 0$$

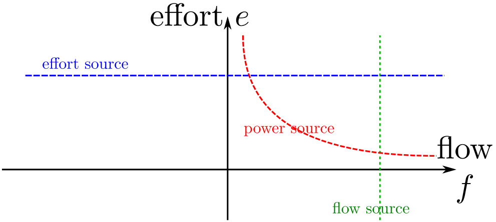
Dissipative elements



- **Power transducers** which convert energy between two domains

  Example: batteries, chemical $\rightarrow$ electrical

- One energy domain (power source) is in the **environment**
- **Ideal sources** are combined with resistors to represent real sources
- Types of ideal sources are **effort**, **flow** and **power** sources



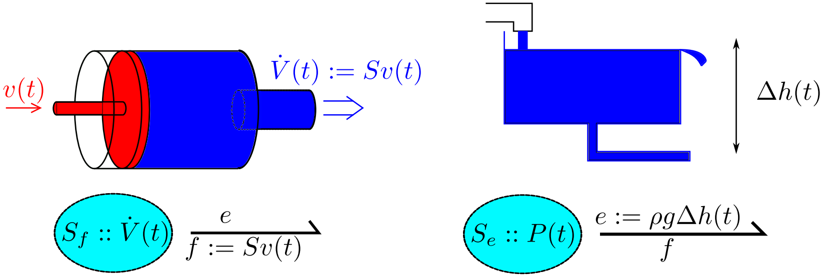
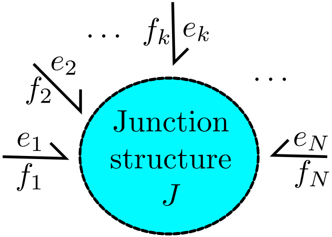


Figure: Examples of flow source (left) and effort source (right) in the hydraulic domain

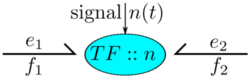
Junction structures characteristics

- interconnect N elements with N independent relations
- **power-preserving interconnection**
- **no storage** (i.e. no dynamics), **no dissipation**
- may include **serial and parallel connections**
- may include **transformers** and **gyrators**
- may have **complex topologies**



Ideal transformers *TF* or transducers

- **2-port** power-preserving element
- **transducer** when power conversion is between distinct energy domains
- no internal dynamics, no dissipation
- the ***transformer modulus***, n may be modulated (signal)



TF constitutive equations

$$\begin{bmatrix} e_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & n \\ -n & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix}$$

Power balance equation:

$$P(t) = e_1 f_1 + e_2 f_2 = n(t) e_2 f_1 - n(t) e_2 f_1 \equiv 0$$

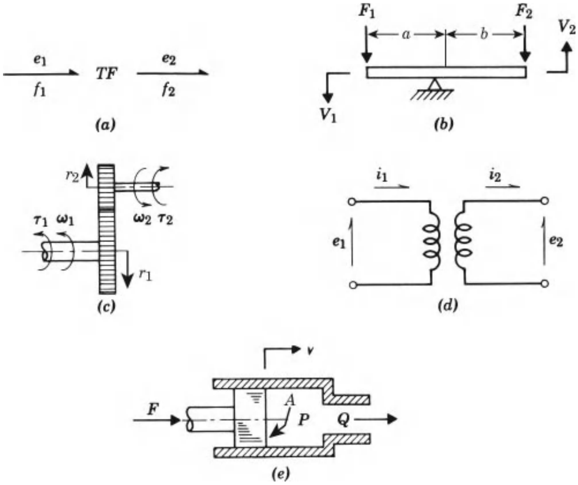


Figure: general Bond Graph transformer (a) ; levers (b) ; toothed gears (c) ; electrical transformers (d) ; piston (transducer) (e)

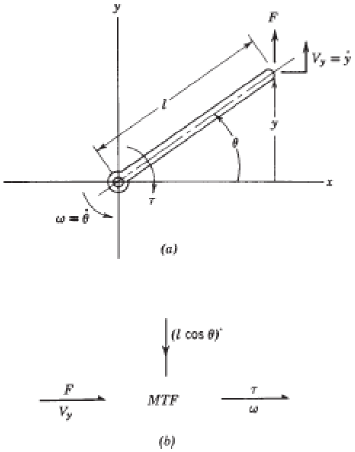
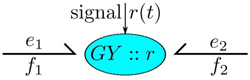


Figure: the connecting rod - a modulated TD between rotational and translational mechanics: schematic view (a) ; Bond Graph (b)

Ideal gyrators *GY* or gyro-transducers

- **2-port** power-preserving element
- **gyro-transducer** if power conversion between distinct energy domains
- no internal dynamics, no dissipation
- the ***gyrator modulus***, r may be modulated (signal)



GY constitutive equations

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 & r \\ -r & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

Power balance equation:

$$P(t) = e_1 f_1 + e_2 f_2 = n(t) f_2 f_1 - n(t) f_1 f_2 \equiv 0$$

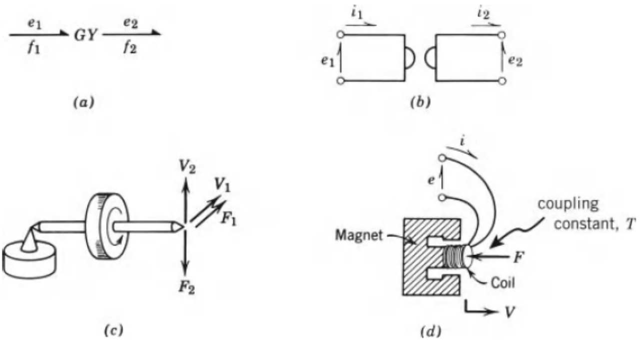


Figure: general Bond Graph gyrator (a) ; electrical gyrator (b) ; gyroscope (c) ; voice coil transducer (d)

GBG or "symplectic" domain classification

$$\xrightarrow[f]{e} \boxed{GY :: 1} \xrightarrow[f_2]{e_2} \boxed{I :: H(p)} \equiv \xrightarrow[f]{e} \boxed{C :: H(q)}$$

$$\begin{cases} e &= f_2 = \dfrac{\partial H}{\partial p} \\ f &= e_2 = \dot{p} \end{cases} \qquad \begin{cases} e &= \dfrac{\partial H}{\partial q} \\ f &= \dot{Q} \end{cases}$$

$\Rightarrow$ one energy state variable (storage element) / energy domain

Remark: **no inductance** (no gyrator) in the **thermal domain**

$$\boxed{S_f :: \dot{S}_0} \xrightarrow[\dot{S}(t)]{T(t)} \boxed{I_{th} :: \frac{p^2}{2I_{th}}}$$

Would give $\dot{S}(t) = \dot{S}_0 > 0$ and $T(t) = I_{th}\dfrac{d\big(\dot{S}(t)\big)}{dt} \equiv 0$,
which contradicts the Nernst principle

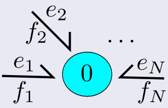
0-junction

- 0-junction **constitutive equations**

$$
\begin{aligned}
e_1 &= e_2 = \ldots = e_N \\
f_1 &+ f_2 + \ldots + f_n + 0
\end{aligned}
$$

Or in **matrix form**

$$
\begin{bmatrix} f_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} 0 & -1 & \ldots & -1 \\ 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}
$$



- **Power balance** equation:

$$
P(t) = e_1 f_1 + \ldots + e_N f_N \equiv 0
$$

Examples: parallel connection in electrical circuits, serial connection in translation mechanics, etc.

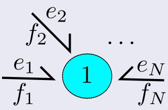
1-junction

- 1-junction **constitutive equations**

$$\begin{aligned} f_1 &= f_2 = \ldots = f_N \\ e_1 &+ e_2 + \ldots + e_n + 0 \end{aligned}$$

Or in **matrix form**

$$\begin{bmatrix} e_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} 0 & -1 & \ldots & -1 \\ 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$



- **Power balance** equation:

$$P(t) = e_1 f_1 + \ldots + e_N f_N \equiv 0$$

Examples: serial connection in electrical circuits, parallel connection in translation mechanics, etc.

Junction structures

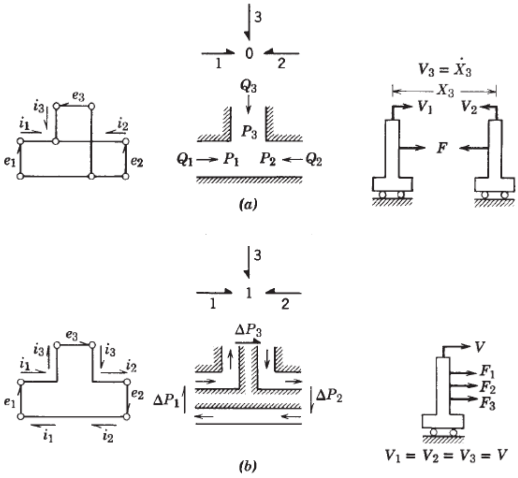
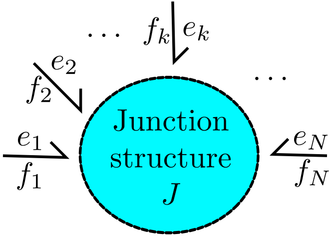


Figure: examples of 3-ports 0-junction (a) and 1-junction (b) in electrical circuits, hydraulics and translational mechanics



Hybrid pairs of input/output power-conjugated variables $\mathbf{u}, \mathbf{y} \in \mathbb{R}^N$:

1. $\mathbf{u}_i, \mathbf{y}_i \in \{f_i, e_i\}$, $\forall i \in \{1, \dots, N\}$
2. $\mathbf{u}_i = f_i \Rightarrow \mathbf{y}_i = e_i$
   $\mathbf{u}_i = e_i \Rightarrow \mathbf{y}_i = f_i$

$$\Rightarrow P(t) := \sum_{k=1}^{N} e_k(t) f_k(t) = \sum_{k=1}^{N} u_k(t) y_k(t)$$

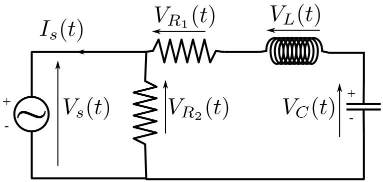
General junction structures

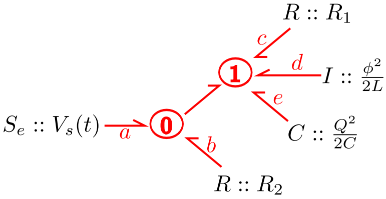
There exists a square matrix $\mathbf{J}(t) \in \mathbb{R}^{N \times N}$ such that:

1. $\mathbf{y} = \mathbf{J}(t)\mathbf{u}$
2. $\operatorname{rank}(\mathbf{J}(t)) = N$ (full rank)
3. $\mathbf{J}(t) = -\mathbf{J}(t)^T$ (skew-symmetry)

Hence

$$P(t) = \mathbf{u}^T \mathbf{y} = \mathbf{u}^T \mathbf{J}(\mathbf{t}) \mathbf{u} \equiv 0$$





- **One 5-port composed interconnection** structure

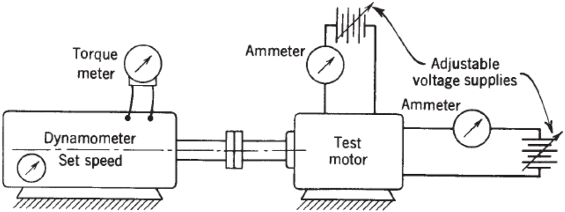
$$\begin{bmatrix} f_a \\ e_b \\ f_c \\ e_d \\ f_e \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & -1 & 0 \\ +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 \\ +1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & +1 & 0 \end{bmatrix} \begin{bmatrix} e_a \\ f_b \\ e_c \\ f_d \\ e_e \end{bmatrix}$$

- **Five 1-port elements** with corresponding constitutive equations
- **Two state variables** ($Q(t)$ and $\phi(t)$) associated to energy storing elements and **one exogeneous variable** ($V_s(t)$)
  $\rightarrow$ 2^{nd} order system with a scalar input

Section 3: The Bond Graph *causality* and *state space equation* algorithms

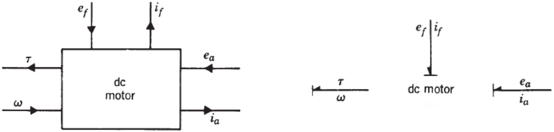
Outline

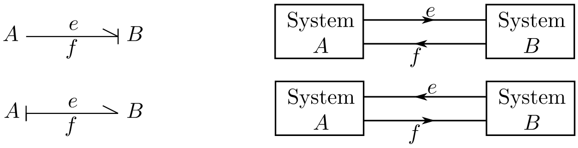
1. What is causality?
2. Causality for basic elements
3. **Causality assignment** algorithm
4. State space models?
5. Procedural state space system (PS^3) construction
6. Conclusion



System theoretic point of view: inputs and outputs

- **inputs** and **outputs** variables may be defined by the plant/problem
- at each port, one can control either the effort or flow, but **not both**
- then the “system” will **give back** the value of the conjugate port-variable





Modeling point of view: subsystems and reticulation

- the modeling process results in **interconnected** subsystems
- some subsystems "interact" with the environment (**inputs**/**outputs**)
- **When** the information can be transmitted at **negligible power levels**, signal connection are used (**no back effect**)
- **Otherwize** subsystems are interconnected by the **matching** of a pair of signals representing the **power variables**
- the **causal stroke** is independent from the **power half arrow**

What comes with causality and Bond Graphs

- automatic generation of the **state space model** / **simulation codes**
  $\rightarrow$ works identically both for the *linear* and *nonlinear* cases
- automatic **feedbacks on the modelling process**
  $\rightarrow$ *contradictions*, *indeterminacies*, *robustness/stiffness* problems
- detection of **algebraic loops**
  $\rightarrow$ differential algebraic systems, *analytical redundancy*
- detection of **structural** observability/reachability problems
  $\rightarrow$ *actuators/sensors placement* in complex network

Element	Acausal Form	Causal Form	Causal Relation
Effort source	$S_e \rightharpoonup$	$S_e \rightharpoonup\!\mid$	$e(t) = E(t)$
Flow source	$S_f \rightharpoonup$	$S_f \mid\!\rightharpoonup$	$f(t) = F(t)$
Resistor	$R \leftharpoondown$	$R \mid\!\leftharpoondown$	$e = \Phi_R(f)$
		$R \mid\!\leftharpoondown$	$f = \Phi_R^{-1}(e)$
Capacitor	$C \leftharpoondown$	$C \leftharpoondown\!\mid$	$e = \Phi_C^{-1}\left(\int^t f\,dt\right)$
		$C \mid\!\leftharpoondown$	$f = \frac{d}{dt}\Phi_C(e)$
Inertia	$I \leftharpoondown$	$I \mid\!\leftharpoondown$	$f = \Phi_I^{-1}\left(\int^t e\,dt\right)$
		$I \leftharpoondown\!\mid$	$e = \frac{d}{dt}\Phi_I(f)$

Figure: Causal forms for basic 1-port elements

Comments on the causality of basic 1-ports

- when (e_R, f_R) is **multiple-valued** in one direction or the other, the **single-valued causality** is preferable
- constitutive equations for C and I elements are usuall expressed in **integral causality**:

$$
\begin{aligned}
e_C(t) &:= \Phi_C^{-1}\left(\int^t f_C(\tau)\, d\tau\right) \\
f_L(t) &:= \Phi_I^{-1}\left(\int^t e_C(\tau)\, d\tau\right)
\end{aligned}
$$

- when **derivative causality** is used (C and I elements):
  - numerical problems related to **rounding off errors** or **signal noise**
  - state of a storage element in derivative causality **statically related** to others states
- often related to **neglected dissipation** and **stiffness** problems

Causality for basic elements

Element	Acausal Graph	Causal Graph	Causal Relations
Transformer	$\overset{1}{\rightharpoonup} TF \overset{2}{\rightharpoonup}$	$\overset{1}{\vdash\!\rightharpoonup} TF \overset{2}{\vdash\!\rightharpoonup}$	$e_1 = me_2$ $f_2 = mf_1$
		$\overset{1}{\rightharpoonup\!\mid} TF \overset{2}{\rightharpoonup\!\mid}$	$f_1 = f_2/m$ $e_2 = e_1/m$
Gyrator	$\overset{1}{\rightharpoonup} GY \overset{2}{\rightharpoonup}$	$\overset{1}{\vdash\!\rightharpoonup} GY \overset{2}{\rightharpoonup\!\mid}$	$e_1 = rf_2$ $e_2 = rf_1$
		$\overset{1}{\rightharpoonup\!\mid} GY \overset{2}{\vdash\!\rightharpoonup}$	$f_1 = e_2/r$ $f_2 = e_1/r$
0-Junction	$\overset{1}{\rightharpoonup} \underset{3\upharpoonleft}{0} \overset{2}{\leftharpoondown}$	$\overset{1}{\rightharpoonup\!\mid} \underset{3\upharpoonleft\!\bot}{0} \overset{2}{\mid\!\leftharpoondown}$	$e_2 = e_1$ $e_3 = e_1$ $f_1 = -(f_2 + f_3)$
1-Junction	$\overset{1}{\rightharpoonup} \underset{3\upharpoonleft}{1} \overset{2}{\leftharpoondown}$	$\overset{1}{\vdash\!\rightharpoonup} \underset{3\overline{\upharpoonleft}}{1} \overset{2}{\leftharpoondown\!\dashv}$	$f_2 = f_1$ $f_3 = f_1$ $e_1 = -(e_2 + e_3)$

Figure: Causal forms for basic 2-ports and 3-ports

Augmenting the Bond Graph

To derive automatically the Bond Graph equations, we will:

1. **number all the bonds in consecutive order**
2. **assign to each bond a causal stroke**

Assumptions

- **state variables** are q variables for the C elements and p variables for the I elements
- **integral causality** is preferred whenever possible
- **Causality rules** for basic Bond Graph elements **apply**
- **Resistors** are assumed to accept **any causality** (no multiple-valued constitutive equation)

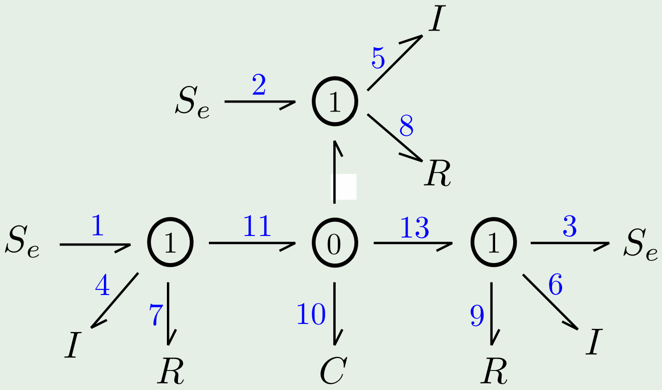
Sequential causal assignment procedure

1. Choose any **source**, assign causality, **propagate** through constraint elements as far as possible
2. **Repeat step 1** untill all sources have been assigned a causality
3. Choose any **storage element** and assign integral causality, then **propagate** as far as possible
4. **Repeat step 3** untill all storage elements have been assigned a causality
5. Choose any **unassigned resistor** element and assign an arbitrary causality, then **propagate** as far as possible
6. **Repeat step 5** untill all resistors have been assigned a causality
7. Choose any **unassigned bond** and assign an arbitrary causality, then **propagate** as far as possible
8. **Repeat step 7** untill all resistors have been assigned a causality

Class 1: graph is complete after step 4 and all storage elements have integral causality

Example

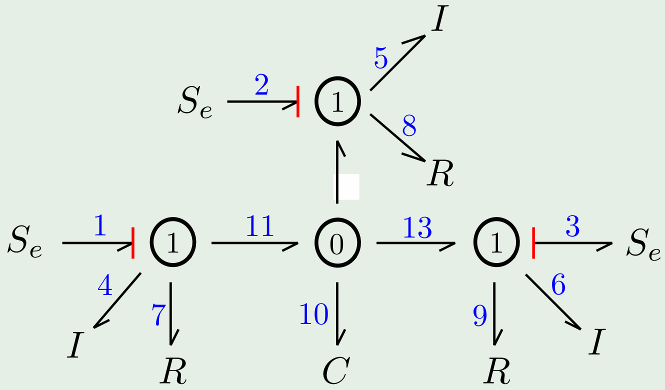
Acausal form (with bonds numbering)



Class 1: graph is complete after step 4 and all storage elements have integral causality

Example

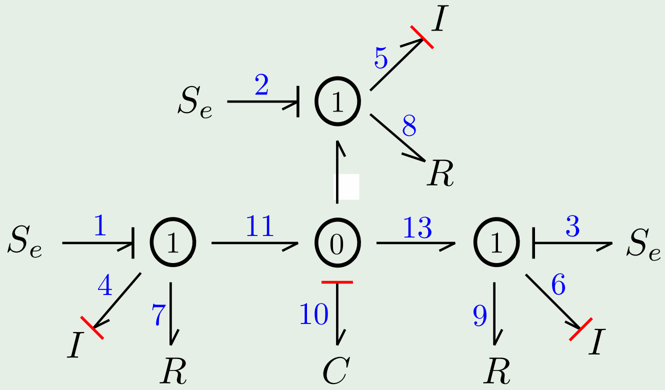
Steps 1-2: assign causality for sources and propagate



Class 1: graph is complete after step 4 and all storage elements have integral causality

Example

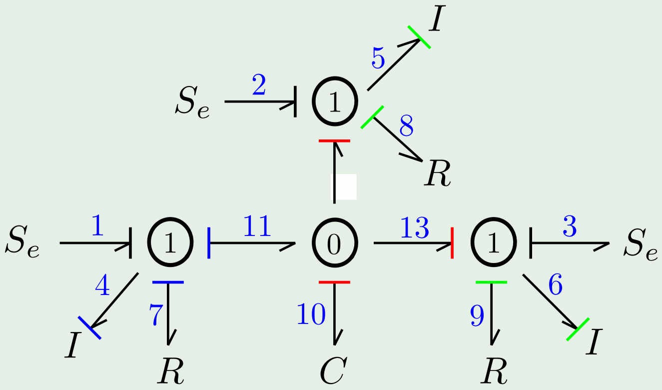
Steps 3-4: assign causality for storage elements and propagate



Class 1: graph is complete after step 4 and all storage elements have integral causality

Example

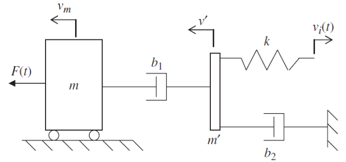
Steps 3-4: assign causality for storage elements and propagate

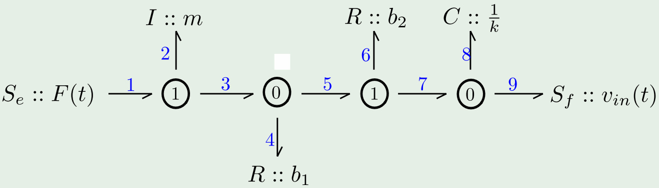


Class 2: causality must be completed using steps 5-8 (arbitrary causality for resistors and/or bonds)

Example

Acausal form (with bonds numbering)

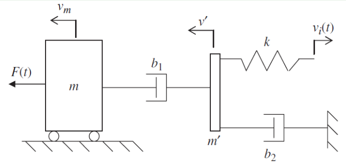


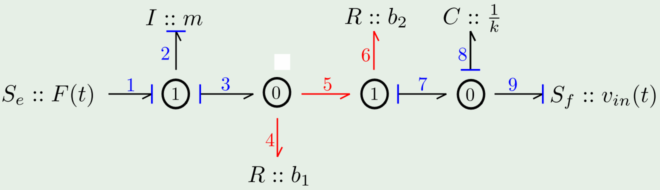


Class 2: causality must be completed using steps 5-8 (arbitrary causality for resistors and/or bonds)

Example

Steps 1-4: assign causality for sources and storages and propagate

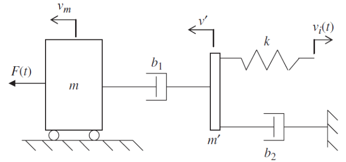


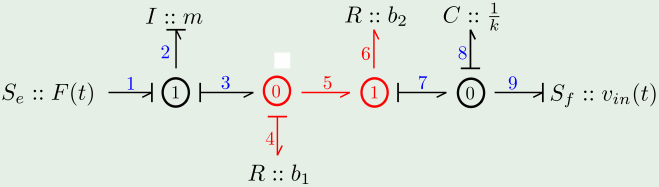


Class 2: causality must be completed using steps 5-8 (arbitrary causality for resistors and/or bonds)

Example

Steps 5-8: assign freely resistor or bond causality and propagate

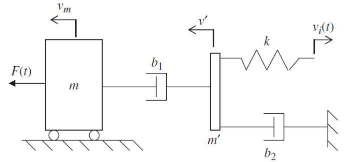


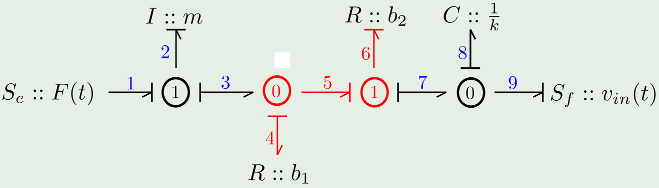


Class 2: causality must be completed using steps 5-8 (arbitrary causality for resistors and/or bonds)

Example

Steps 5-8: assign freely resistors or bonds causality and propagate

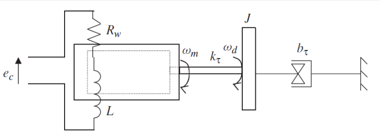


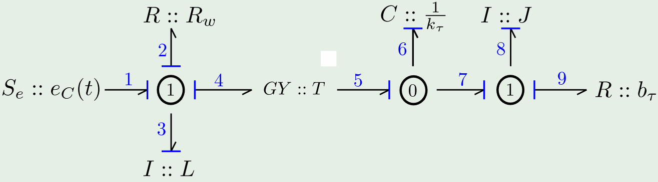


Class 3: some storage elements are forced into derivative causality

Example

Choosing integral causality for I_3

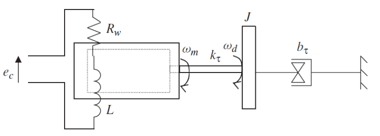


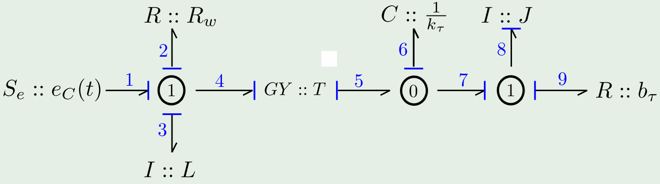


Class 3: some storage elements are forced into derivative causality

Example

Choosing integral causality for C_6

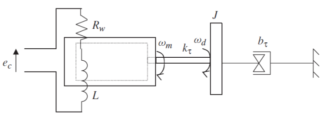


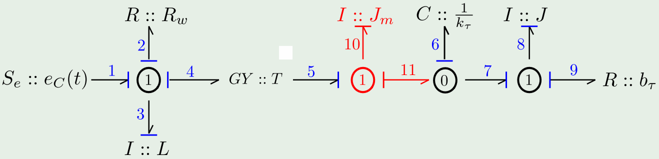


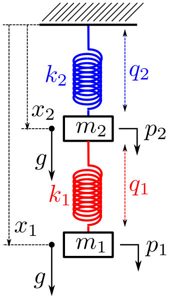
Class 3: some storage elements are forced into derivative causality

Example

Taking into account the rotational inertia of the motor, J_m







- Fourth order ODE (Classical approach)

$$x_2^{(4)} + \left[\frac{k_2}{m_2} + k_1 \frac{m_1 + m_2}{m_1 m_2} \right] x_2^{(2)} + \frac{k_1 k_2}{m_1 m_2} x_2 = k_1 \frac{m_1 + m_2}{m_1 m_2} g$$

- Two second order ODEs (Lagrangian approach)

$$\begin{aligned} \ddot{q}_1 + k_1 \frac{m_1 + m_2}{m_1 m_2} q_1 - \frac{k_2}{m_2} q_2 &= 0 \\ \ddot{q}_2 - \frac{k_1}{m_2} q_1 + \frac{k_2}{m_2} q_2 &= g \end{aligned}$$

- Four first order ODEs (Hamiltonian approach)

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -k_1 & 0 \\ 0 & 0 & +k_1 & -k_2 \\ \frac{1}{m_1} & \frac{-1}{m_2} & 0 & 0 \\ 0 & +m_2^{-1} & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ q_1 \\ q_2 \end{bmatrix} + \begin{bmatrix} m_1 g \\ m_2 g \\ 0 \\ 0 \end{bmatrix}$$

$$\left\{\begin{array}{lcl} \dot{x}_1(t) & = & f_1\left(t, x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_r(t)\right) \\ \dot{x}_2(t) & = & f_2\left(t, x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_r(t)\right) \\ & \vdots & \\ \dot{x}_n(t) & = & f_n\left(t, x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_r(t)\right) \end{array}\right.$$

Various forms for the differential equations model:

1. **one n^{th} order** differential equation of **one unknown** variable
2. *n* **first order** differential equations with *n* **unknown** variables
3. **combinations** of differential-algebraic equations of **appropriate orders**

State space form $\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}, \mathbf{u}\right)$

- convenient for system **analysis**, **control** design and **simulations**
- states are **stored energy variables** and **initial conditions** are obvious
- transformation from one form to another may be **difficult to discover** in the non linear case

Procedural construction of the State-Space System (PS³)

To **convert** a Bond Graph model to state space differential equations:

1. select **input** variables ($\mathbf{u}(t)$) and **state** variables ($\mathbf{x}(t)$)
2. generate an initial set of **executable equations** according to **causality**
3. **reduce** to the state space form

State space equations and causality classes

- Class 1: causality is automatically assigned, no derivative causality

  $\rightarrow$ **basic formulation** and reduction

- Class 2: causality is **arbitrarily chosen**, no derivative causality

  $\rightarrow$ **extended formulation** (algebraic loops)

- Class 3: some storage elements are with **derivative causality**

  $\rightarrow$ **extended formulation**

$\mathbf{PS^3}$ algorithm

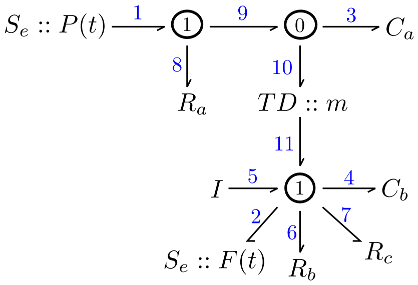


Figure: Example: pressure-controlled valve system, the Bond Graph model with bonds numbering

The PS^3 algorithm for Class 1 Bond Graphs

Class1: causality is completed without any arbitrary choice and all storage elements have integral causality

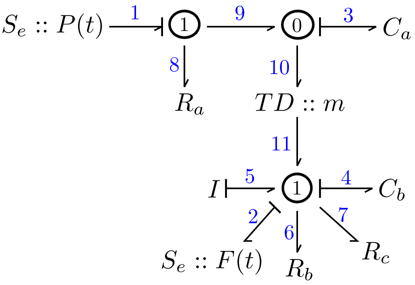


Figure: Example: pressure-controlled valve system, the Bond Graph model with bonds numbering

The PS^3 algorithm for Class 1 Bond Graphs

Class1: causality is completed without any arbitrary choice and all storage elements have integral causality

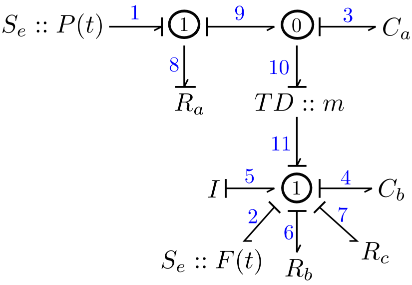
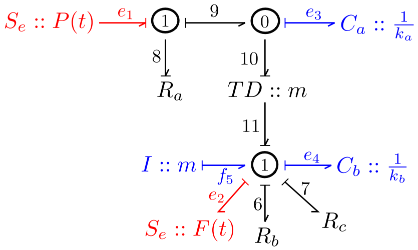


Figure: Example: pressure-controlled valve system, the Bond Graph model with bonds numbering

The PS^3 algorithm for Class 1 Bond Graphs

Class1: causality is completed without any arbitrary choice and all storage elements have integral causality

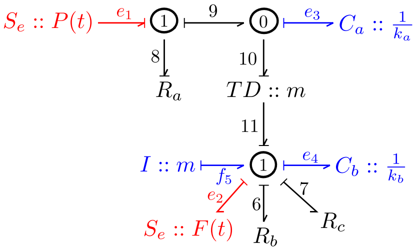


The PS^3 algorithm for Class 1 Bond Graphs

Initialization: select input variables

$$e_1 := u_1 \text{ with } (u_1 = P(t))$$
$$e_2 := u_2 \text{ with } (u_2 = F(t))$$

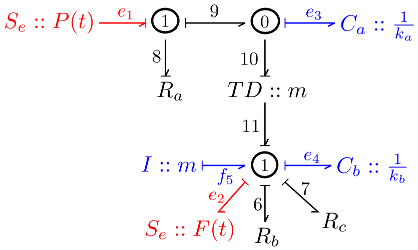
PS^3 algorithm



The PS^3 algorithm for Class 1 Bond Graphs

Initiatlization: select state variables

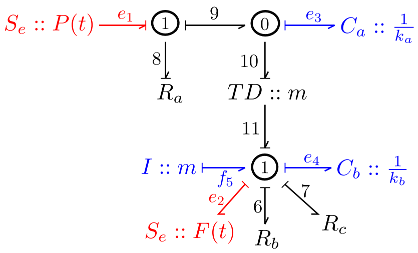
$$q_a := x_1$$
$$q_b := x_2$$
$$\phi := x_3$$



The PS3 algorithm for Class 1 Bond Graphs

Initialization: constitutive equations for the storage elements (integral causality)

$$e_3 := k_a q_a$$
$$e_4 := k_b q_b$$
$$f_5 := \frac{\phi}{m}$$

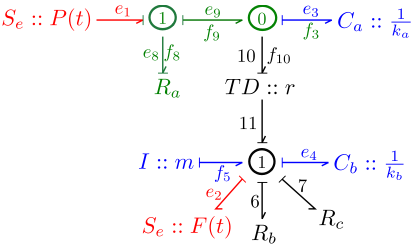


The PS^3 algorithm for Class 1 Bond Graphs

Step 1: compute the rate of change for all the energy state variables (integral causality)

$$\dot{x}_1 := f_3$$
$$\dot{x}_2 := f_4$$
$$\dot{x}_3 := e_5$$

Remark: we start with $\dot{x}_1$, according to the index increasing order. Then f_3 must be computed using causal relations

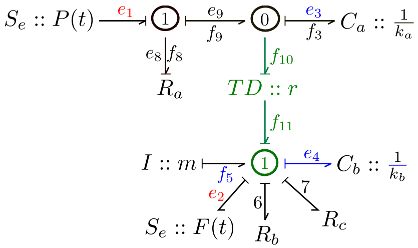


The PS³ algorithm for Class 1 Bond Graphs

Step k: compute unkown effort/flow variables in step $k-1$

$$
\begin{aligned}
f_3 &:= f_9 - f_{10}; \ \text{(0-junction)} \\
f_9 &:= f_8; \ \text{(1-junction)} \\
f_8 &:= R_a^{-1} e_8; \ \text{(Resistor } R_a \text{ constitutive equation)} \\
e_8 &:= e_1 - e_9; \ \text{(1-junction)} \\
e_9 &:= e_3; \ \text{(0-junction)}
\end{aligned}
$$

Remark: variable e_1 is an input and e_3 already computed from the C_a constitutive equation ; f_{10} must be computed using causal relations

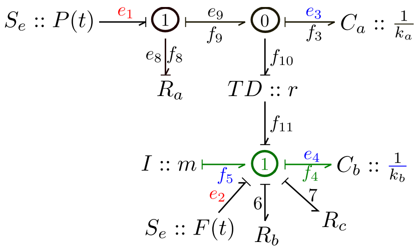


The PS^3 algorithm for Class 1 Bond Graphs

Step k: compute unkown effort/flow variables in step $k-1$

$$f_{10} := r^{-1} f_{11}; \ (TD \text{ constitutive equation})$$
$$f_{11} := f_5; \ (1\text{-junction})$$

Remark: variable f_5 is already computed from the I constitutive equation
$\rightarrow$ **second state equation**



The PS^3 algorithm for Class 1 Bond Graphs

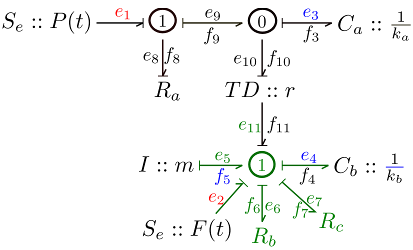
$$\dot{x}_2 := f_4$$

with

$$f_4 := f_5; \text{ (1-junction)}$$

Remark: variable f_5 is already computed
$\rightarrow$ **third state equation**

PS^3 algorithm

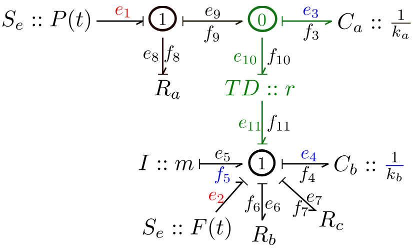


The PS^3 algorithm for Class 1 Bond Graphs

$$
\begin{aligned}
\dot{x}_3 &:= e_5;\\
e_5 &:= e_4 + e_7 + e_6 - e_2 - e_{11};\\
e_6 &:= R_b f_6;\\
f_6 &:= f_5;\\
e_7 &:= R_c f_7;\\
f_7 &:= f_5;
\end{aligned}
$$

Remark: variable e_{11} must be computed using causal relations

PS^3 algorithm



The PS^3 algorithm for Class 1 Bond Graphs

$$e_{11} := r^{-1} e_{10}; \ (TD \text{ constitutive equation})$$
$$e_{10} := e_3; \ (0\text{-junction})$$

Termination condition:

- **all state equations** have been computed
- **No more unknown variables** in the state equations

Remark: $f_1(t)$ and $f_2(t)$ are not necessary but may be computed easily

The PS^3 algorithm for Class 1 Bond Graphs

Last step: obtained equations written backward gives an executable code

$$e_1 := u_1;$$
$$e_2 := u_2;$$

$$q_a := x_1$$
$$q_b := x_2$$
$$\phi := x_3$$

$$e_3 := k_a q_a$$
$$e_4 := k_b q_b$$
$$f_5 := \frac{\phi}{m}$$

$$e_{10} := e_3;$$
$$e_{11} := r^{-1} e_{10};$$
$$f_7 := f_5;$$
$$e_7 := R_c f_7;$$
$$f_6 := f_5;$$
$$e_6 := R_b f_6;$$
$$e_5 := e_4 + e_7 + e_6 - e_2 - e_{11};$$
$$\dot{x}_3 := e_5;$$

$$f_4 := f_5;$$
$$\dot{x}_2 := e_4;$$
$$f_{11} := f_5;$$
$$f_{10} := r^{-1} f_{11};$$
$$e_9 := e_3;$$
$$e_8 := e_1 - e_9;$$
$$f_8 := R_a^{-1} e_8;$$
$$f_9 := f_8;$$
$$f_3 := f_9 - f_{10};$$
$$\dot{x}_1 := f_3;$$

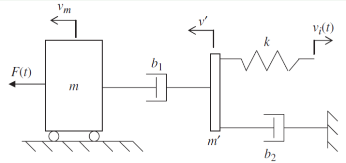
Automatic substitutions gives the state space form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \frac{-1}{R_a C_a} & 0 & \frac{-1}{rm} \\ 0 & 0 & \frac{1}{m} \\ \frac{1}{rC_a} & \frac{-1}{C_b} & \frac{-(R_b+R_c)}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{R_a} & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Class 2: causality must be completed using steps 5-8 (arbitrary causality for resistors and/or bonds)

Example

Acausal form (with bonds numbering)

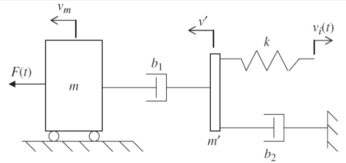


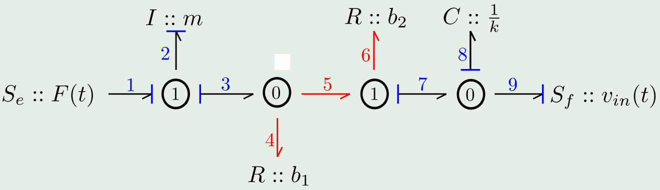
$$\begin{tikzcd}
 & I :: m & & R :: b_2 & C :: \frac{1}{k} & \\
S_e :: F(t) \arrow[r, "1"] & 1 \arrow[u, "2"] \arrow[r, "3"] & 0 \arrow[d, "4"] \arrow[r, "5"] & 1 \arrow[u, "6"] \arrow[r, "7"] & 0 \arrow[u, "8"] \arrow[r, "9"] & S_f :: v_{in}(t) \\
 & & R :: b_1 & & &
\end{tikzcd}$$

Class 2: causality **must be completed** using **steps 5-8** (arbitrary causality for resistors and/or bonds)

Example

Steps 1-4: assign causality for sources and storages and propagate

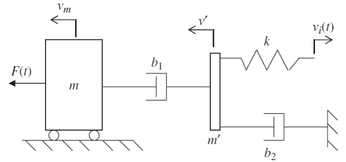


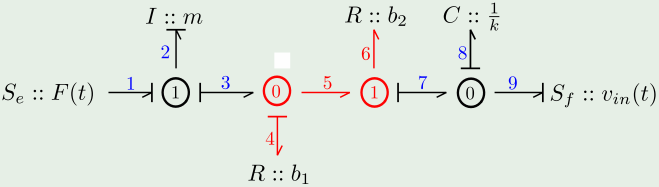


Class 2: causality must be completed using steps 5-8 (arbitrary causality for resistors and/or bonds)

Example

Steps 5-8: assign freely resistor or bond causality and propagate

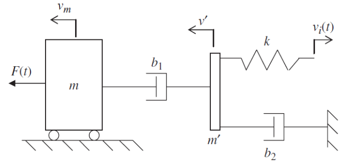


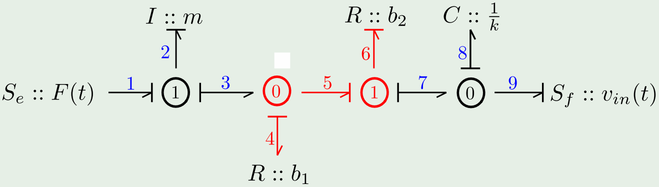


Class 2: causality must be completed using steps 5-8 (arbitrary causality for resistors and/or bonds)

Example

Steps 5-8: assign freely resistors or bonds causality and propagate





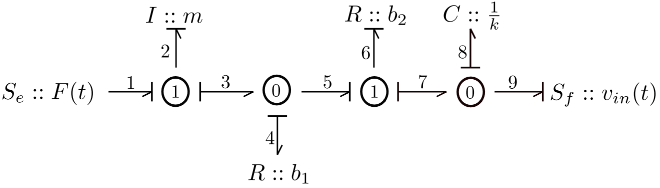
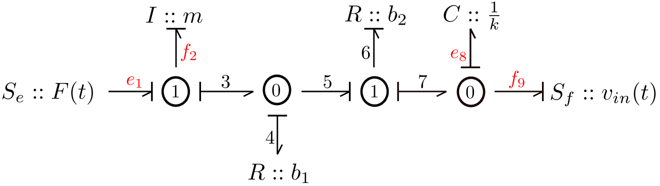


Figure: A class 2 example: mass-spring-damper system

The PS^3 algorithm for Class 2 Bond Graphs

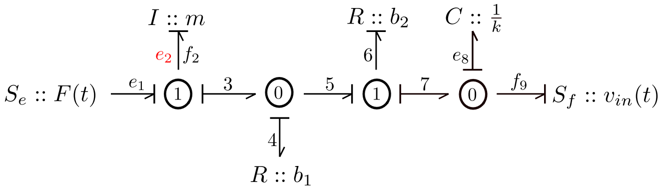
Class 2: causality must be assigned arbitrarily for some bonds - no storage elements in integral causality



The PS^3 algorithm for Class 2 Bond Graphs

Initialization:

$$
\begin{aligned}
e_1 &:= u_1 \quad \text{with } (u_1 = F(t)) \\
f_9 &:= u_2 \quad \text{with } (u_2 = v_{in}(t)) \\
p_m &:= x_1; \\
q_s &:= x_2; \\
f_2 &:= \frac{p_m}{m}; \\
e_8 &:= kq_s;
\end{aligned}
$$

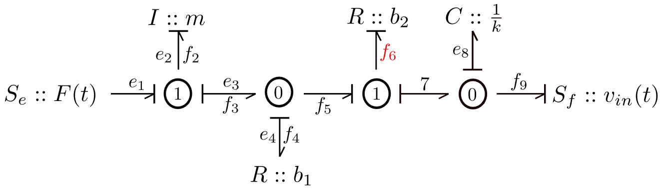


The PS^3 algorithm for Class 2 Bond Graphs

Step 1: compute the rate of change for all the energy state variables (integral causality)

$$\dot{x}_1 := e_2$$
$$\dot{x}_2 := f_8$$

Remark: we start with $\dot{x}_1$. Thus e_2 must be computed



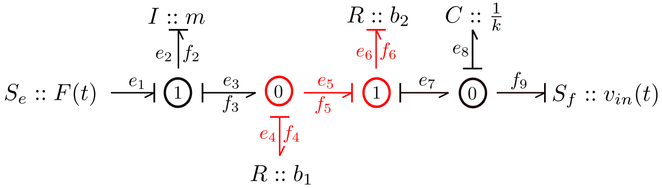
The PS³ algorithm for Class 2 Bond Graphs

Step k: compute unkown effort/flow variables in step $k-1$

$$
\begin{aligned}
e_2 &:= e_1 - e_3;\\
e_3 &:= e_4;\\
e_4 &:= b_1 f_4;\\
f_4 &:= f_3 - f_5;\\
f_3 &:= f_2;\\
f_5 &:= f_6;
\end{aligned}
$$

Remark: f_6 must be computed using causal relations

PS³ algorithm

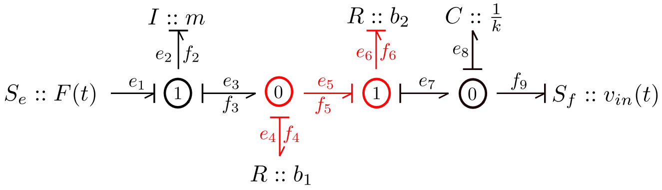


The PS³ algorithm for Class 2 Bond Graphs

Step k: compute unkown effort/flow variables in step $k-1$

$$
\begin{aligned}
f_6 &:= b_2^{-1} e_6;\\
e_6 &:= e_5 - e_7;\\
e_7 &:= e_8;\\
e_5 &:= e_4;\\
e_4 &:= b_1 f_4 = b_1(f_3 - f_5) = b_1(f_2 - f_6);
\end{aligned}
$$

Problem: algebraic loop (endless substitution)!



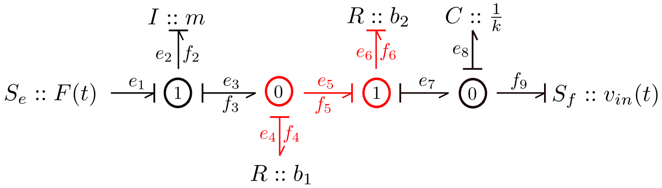
The PS3 algorithm for Class 2 Bond Graphs

To solve the algebraic loop problem

- **Select the output variable** e_4 of the element with forced causality and derive the corresponding algebraic relation

$$\begin{aligned} e_4 &:= b_1 f_4 = b_1(f_3 - f_5) = b_1(f_2 - f_6) = b_1(f_2 - b_2^{-1} e_6) \\ &= b_1(f_2 - b_2^{-1}(e_5 - e_7)) = b_1(f_2 - b_2^{-1}(e_4 - e_8)) \\ &= b_1(\tfrac{x_1}{m} - b_2^{-1}(e_4 - k_s x_2)) \end{aligned}$$

PS^3 algorithm



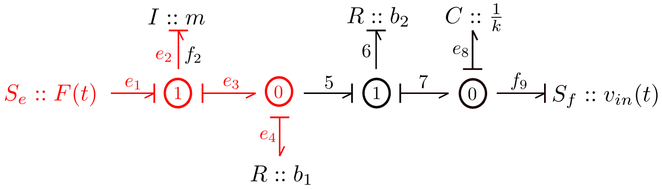
The PS^3 algorithm for Class 2 Bond Graphs

To solve the algebraic loop problem

- **solve the algebraic relation** to find the forced output variable e_4 as a function of the state and input variables

$$e_4 = \frac{b_1}{b_1 + b_2}\left(\frac{b_2}{m} x_1 + k_s x_2\right)$$

- Derive the state equation following the standard procedure but **substitute the output variable** e_4 when it enters the formulation



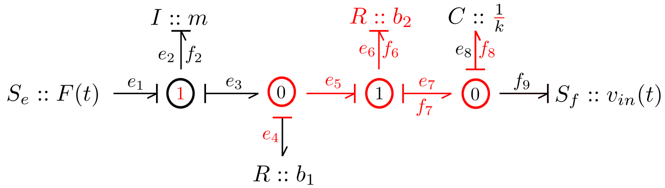
The PS^3 algorithm for Class 2 Bond Graphs

Applying the standard procedure results in:

$$e_2 := e_1 - e_3;$$
$$e_3 := e_4;$$
$$e_4 = \frac{b_1}{b_1 + b_2}\left(\frac{b_2}{m} x_1 + k_s x_2\right)$$

Remark: the first state equation is complete and one looks now for $\dot{x}_2 = f_8$

PS^3 algorithm



The PS^3 algorithm for Class 2 Bond Graphs

$$
\begin{aligned}
f_8 &:= f_7 - f_9;\\
f_7 &:= f_6;\\
f_6 &:= b_2^{-1} e_6;\\
e_6 &:= e_5 - e_7;\\
e_7 &:= e_8;\\
e_5 &:= e_4;
\end{aligned}
$$

Remark: the second state equation is complete!

Remark: f_3, f_4 and f_5 are **not necessary** but may be computedeasily

The PS^3 algorithm for Class 2 Bond Graphs

The set of obtained equations written backward gives the executable code

$$e_1 := u_1;$$
$$f_9 := u_2;$$

$$f_2 := \frac{x_1}{m};$$
$$e_8 := kx_2;$$

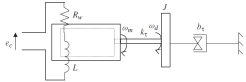
$$e_4 := \frac{b_1}{b_1 + b_2}\left(\frac{b_2}{m}x_1 + k_s x_2\right)$$

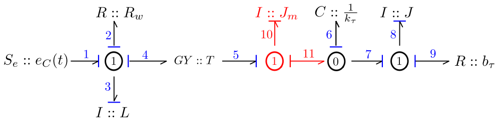
$$\begin{aligned}
e_5 &:= e_4;\\
e_7 &:= e_8;\\
e_6 &:= e_5 - e_7;\\
f_6 &:= b_2^{-1} e_6;\\
f_7 &:= f_6;\\
f_8 &:= f_7 - f_9;\\
\dot{x}_2 &:= f_8\\
e_3 &:= e_4;\\
e_2 &:= e_1 - e_3;\\
\dot{x}_1 &:= e_2
\end{aligned}$$

Automatic substitutions gives the state space form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \frac{1}{m(b_1 + b_2)} \begin{bmatrix} -b_1 b_2 & -b_1 k \\ b_1 & \frac{k(b_1 - m(b_1 + b_2))}{b_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

PS^3 algorithm





Remarks (class 2)

- state equations are **independent** from the chosen *bond* and causality
- the algebraic loop may result in a **non linear equation**
- solving the algebraic loop **may not be possible** (multiple valued relation such as in the Coulomb friction)
- the causality undetermination problem may be solved by **modifying the modelling assumptions** (e.g. adding a mass in the previous example)

Class 3: some storage elements are forced into derivative causality

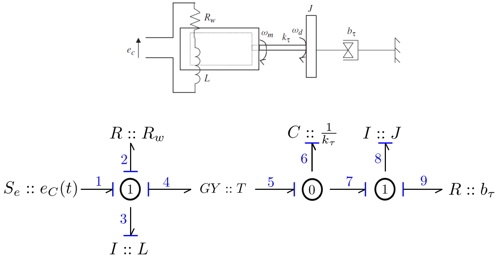


Figure: Permanent magnet DC motor driving a rotational load

- the motor inductance and rotational inertia are with integral causality.
- the **rotational spring** is with **derivative causality**.

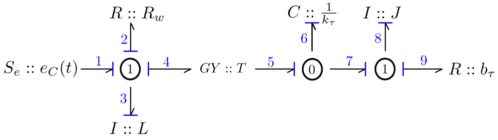
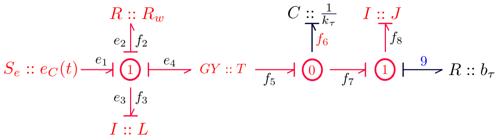


Figure: Permanent magnet DC motor driving a rotational load

Storage elements in derivative causality are not **dynamically independent**

- **no differential equation** for the rate of change of the corresponding energy variable in the final state space model
- state equations for the other states (corresponding to elements in integral causality) arise in **implicit form** during the formulation
- the dependent storage variables may not be simply ignored in the formulation

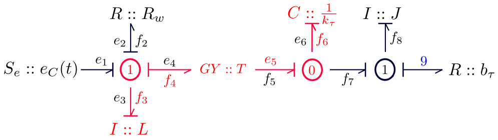
PS^3 algorithm



State equation for the L element (integral causality)

$$\begin{aligned}
\dot{p}_3 &:= e_3; \\
e_3 &:= e_1 - e_2 - e_4; \\
e_2 &:= R_w f_2; \\
e_4 &:= T f_5; \\
f_2 &:= f_3; \\
f_3 &:= \frac{p_3}{L}; \\
f_5 &:= f_6 + f_7; \\
f_7 &:= f_8; \\
f_8 &:= \frac{p_8}{J};
\end{aligned}$$

Problem: $f_6 = \dot{q}_6$ is a function of other independent variables



Compute f_6 and solve for $\dot{p}_3$

$$f_6 := \dot{q}_6;$$

$$q_6 := \frac{1}{k_\tau} e_6 = \frac{1}{k_\tau} e_5 = \frac{1}{k_\tau} T f_4 = \frac{1}{k_\tau} T f_3 = \frac{1}{k_\tau} T \frac{p_3}{L};$$

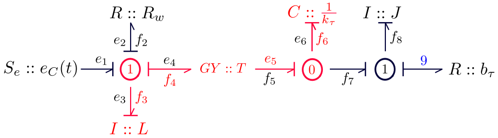
$$\Rightarrow \quad \dot{q}_6 = \frac{1}{k_\tau} T \frac{\dot{p}_3}{L}$$

$$\Rightarrow \quad \dot{p}_3 = e_1 - R_w \frac{p_3}{L} - T \left(\frac{1}{k_\tau} T \frac{\dot{p}_3}{L} + \frac{p_8}{J} \right)$$

$$\Rightarrow \quad \dot{p}_3 = \frac{k_\tau e_1}{k_\tau + T^2 L} - \frac{k_\tau R_w / L}{k_\tau + T^2 L} p_3 - \frac{k_\tau T / J}{k_\tau + T^2 L} p_8$$

Remark: the second state equation $f_8 = \dot{q}_8$ is still needed

PS[3] algorithm



State equation for the J element (integral causality)

$$
\begin{aligned}
\dot{p}_8 &:= e_8;\\
e_8 &:= e_7 - e_9;\\
e_7 &:= e_5;\\
e_9 &:= b_\tau f_9;\\
e_5 &:= T f_4;\\
f_9 &:= f_8;\\
f_8 &:= \frac{p_8}{J};\\
f_4 &:= f_3;\ f_3 := \frac{p_3}{L};
\end{aligned}
$$

Remark: no specific problem for $f_8 = \dot{q}_8$

The set of obtained equations written backward gives the executable code

$$
\begin{aligned}
e_1 &:= u_1; \\
p_3 &:= x_1; \\
p_8 &:= x_2; \\
f_3 &:= \frac{p_3}{L}; \\
f_8 &:= \frac{p_8}{J}; \\
e_3 &:= \frac{k_\tau}{k_\tau + T^2 L}\left[e_1 - \frac{R_w}{L} p_3 - \frac{T}{J} p_8\right]; \\
\dot{p}_3 &:= e_3;
\end{aligned}
$$

$$
\begin{aligned}
e_5 &:= e_4; \\
f_4 &:= f_3; \\
f_9 &:= f_8; \\
e_5 &:= T f_4; \\
e_9 &:= b_\tau f_9; \\
e_7 &:= e_5; \\
e_8 &:= e_7 - e_9; \\
\dot{p}_8 &:= e_8;
\end{aligned}
$$

Automatic substitutions gives the state space form

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{-k_\tau R_w}{L(k_\tau + T^2 L)} & \frac{k_\tau T}{J(k_\tau + T^2 L)} \\ \frac{T}{L} & \frac{-b\tau}{J} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{k_\tau}{k_\tau + T^2 L} \\ 0 \end{bmatrix} u_1
$$

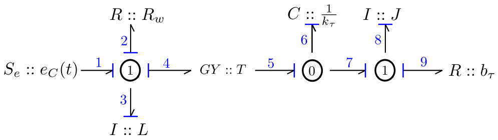


Figure: Permanent magnet DC motor driving a rotational load

Remarks on the PS³ for Class 3 examples

Storage elements in derivative causality are not **dynamically independent**

- **same method** when another storage element is in derivative causality
  $\rightarrow$ state equations are different but **predicted dynamics are the same**
- **solving the implicit differential equations** in the non linear case?
  $\rightarrow$ Differential-Algebraic Equations Solvers (DAES) packages exist
- Modifying the model may solve the "derivative causality" problem
  - adding the rotational axis inertia will **increase the order**
  - adding a **small dissipation** will create a **stiffness problem**

Achievements in section 3

- **Causality** is introduced to allow automatic generation of the state space model
- **Causality assignment algorithm** detects modelling issues / problems
- **PS^3 algorithm** generates the state space formulation for:
  - **Class 1** models (no causality issues)
  - **Class 2** models (bonds with arbitrary causality)
  - **Class 3** models (storage elements in derivative causality)