Modelado y Simulación de Sistemas Dinámicos: Métodos, Algoritmos y Herramientas

Sistemas Continuos

Ernesto Kofman

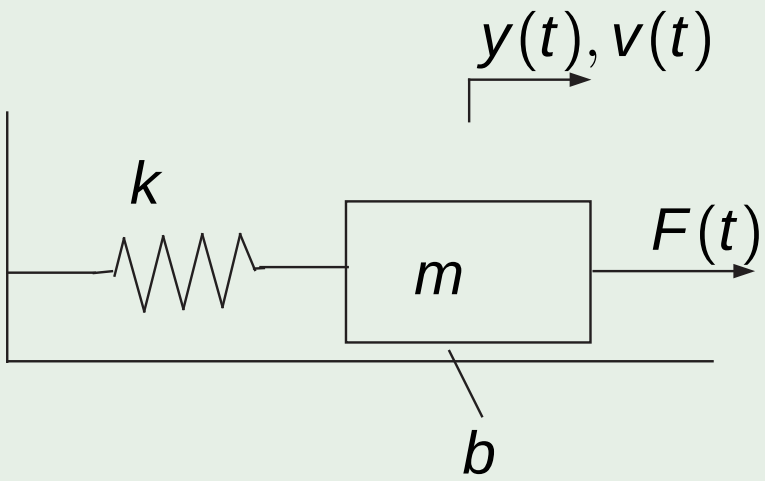
Laboratorio de Sistemas Dinámicos y Procesamiento de la Información
FCEIA - Universidad Nacional de Rosario.
CIFASIS – CONICET. Argentina

Organización de la Presentación

Organización de la Presentación

Ejemplo Introductorio

Sistema Masa–Resorte



Relaciones Constitutivas:

- $F_{neta} = m \cdot \ddot{y}(t)$ (Newton)
- $F_{res} = k \cdot \delta(t)$ (Resorte)
- $F_{roz} = b \cdot v_{roz}$ (Fricción)

Relaciones Estructurales:

- $F(t) - F_{roz} - F_{res} = F_{neta}$
- $\delta(t) = y(t)$
- $v_{roz} = \dot{y}(t)$

$$\Rightarrow m \cdot \ddot{y}(t) + b \cdot \dot{y}(t) + k \cdot y(t) = F(t)$$

Ecuaciones Diferenciales Ordinarias

El modelo que obtuvimos del sistema masa–resorte consituye una Ecuación Diferencial Ordinaria (EDO).

$$m \cdot \ddot{y}(t) + b \cdot \dot{y}(t) + k \cdot y(t) = F(t)$$

- $y(t)$ es la variable de salida del modelo.
- $F(t)$ es la variable de entrada.
- El modelo es de segundo orden.
- Es un modelo lineal y estacionario.

Ecuaciones de Estados

Una EDO de orden n puede reescribirse como un sistema de n EDOs de orden 1, denominadas Ecuaciones de Estado. Por ejemplo, para la EDO

$$m \cdot \ddot{y}(t) + b \cdot \dot{y}(t) + k \cdot y(t) = F(t)$$

definiendo $x(t) \triangleq y(t)$, $v(t) \triangleq \dot{x}(t)$, queda el sistema

$$\dot{x}(t) = v(t)$$
$$\dot{v}(t) = \frac{1}{m}[-k\,x(t) - b\,v(t) + F(t)]$$

en el cual las variables $x(t)$ y $v(t)$ se denominan variables de estado

Ecuaciones de Estado. Forma General

La forma general de las Ecuaciones de Estado (EE) es la que sigue:

$$\dot{x}_1(t) = f_1(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t), t)$$
$$\vdots$$
$$\dot{x}_n(t) = f_n(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t), t)$$

que se puede completar con Ecuaciones de Salida:

$$y_1(t) = g_1(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t), t)$$
$$\vdots$$
$$y_p(t) = g_p(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t), t)$$

Modelado con Ecuaciones Diferenciales

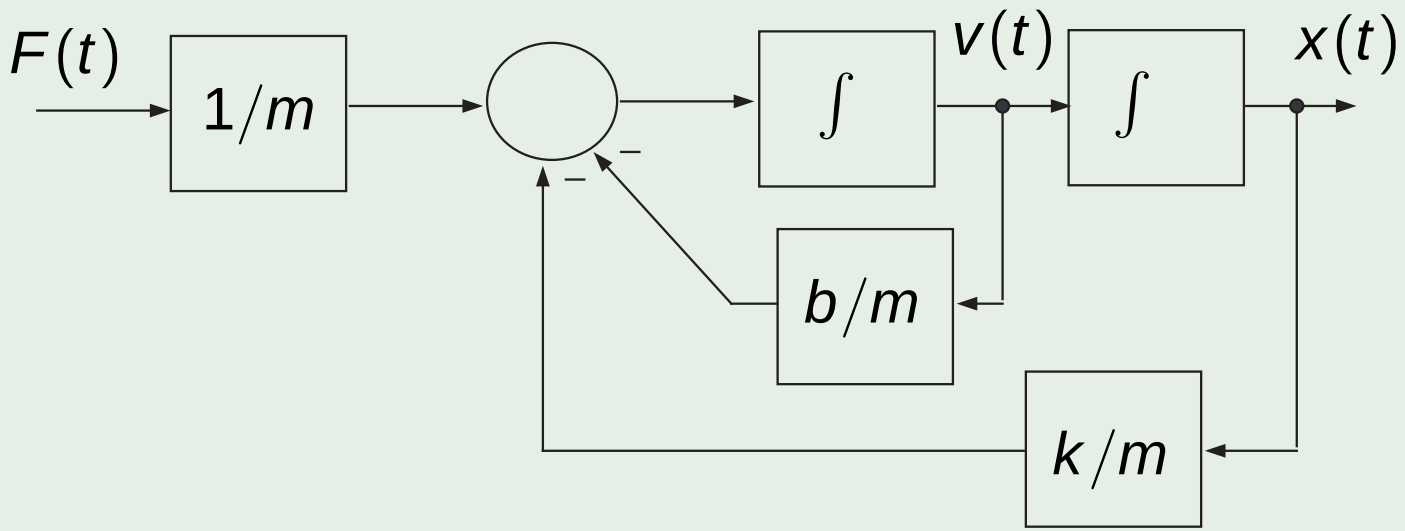
Si bien los Sistemas Continuos se representan mediante modelos de Ecuaciones Diferenciales, es muy dificil obtener directamente las ecuaciones correspondiente a un modelo complejo, ya que:

- los sistemas de ecuaciones no se pueden acoplar para obtener modelos complejos a partir de modelos simples,
- llegar a la EDO desde las relaciones constitutivas y estructurales requiere mucho trabajo algebraico,
- los modelos obtenidos no guardan similitud visual con los esquemas originales.

Debido a esto, para obtener modelos de sistemas continuos se prefiere utilizar lenguajes gráficos.

Diagramas de Bloques

Diagrama de Bloques



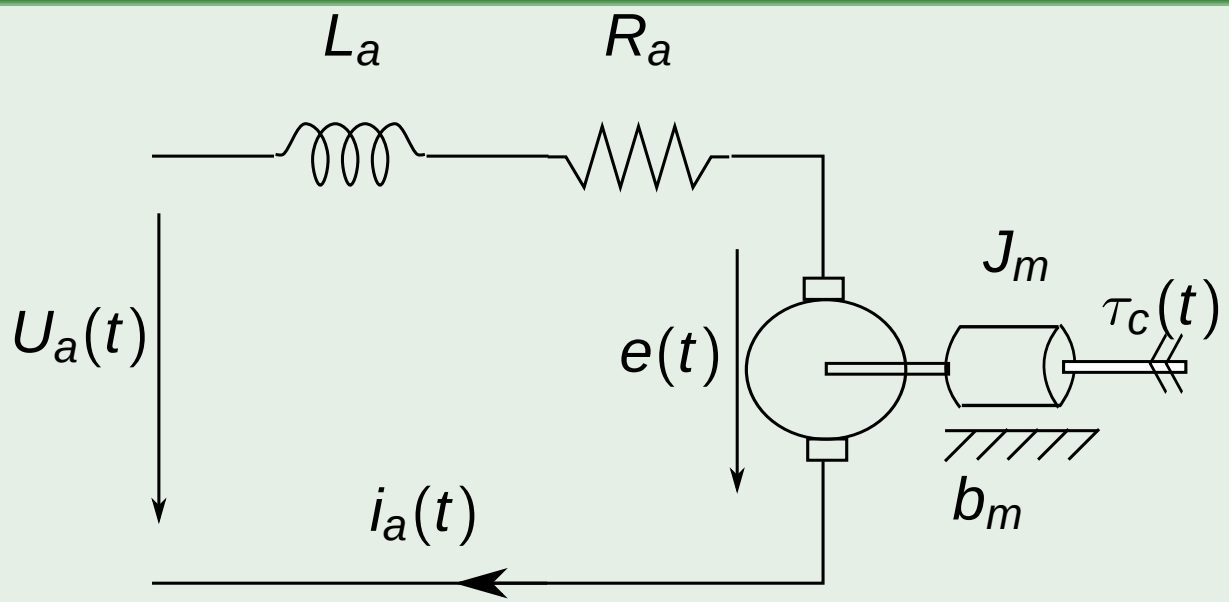
$$\dot{x}(t) = v(t)$$

$$\dot{v}(t) = -\frac{k}{m}x(t) - \frac{b}{m}v(t) + \frac{1}{m}F(t)$$

Modelado con Diagramas de Bloques

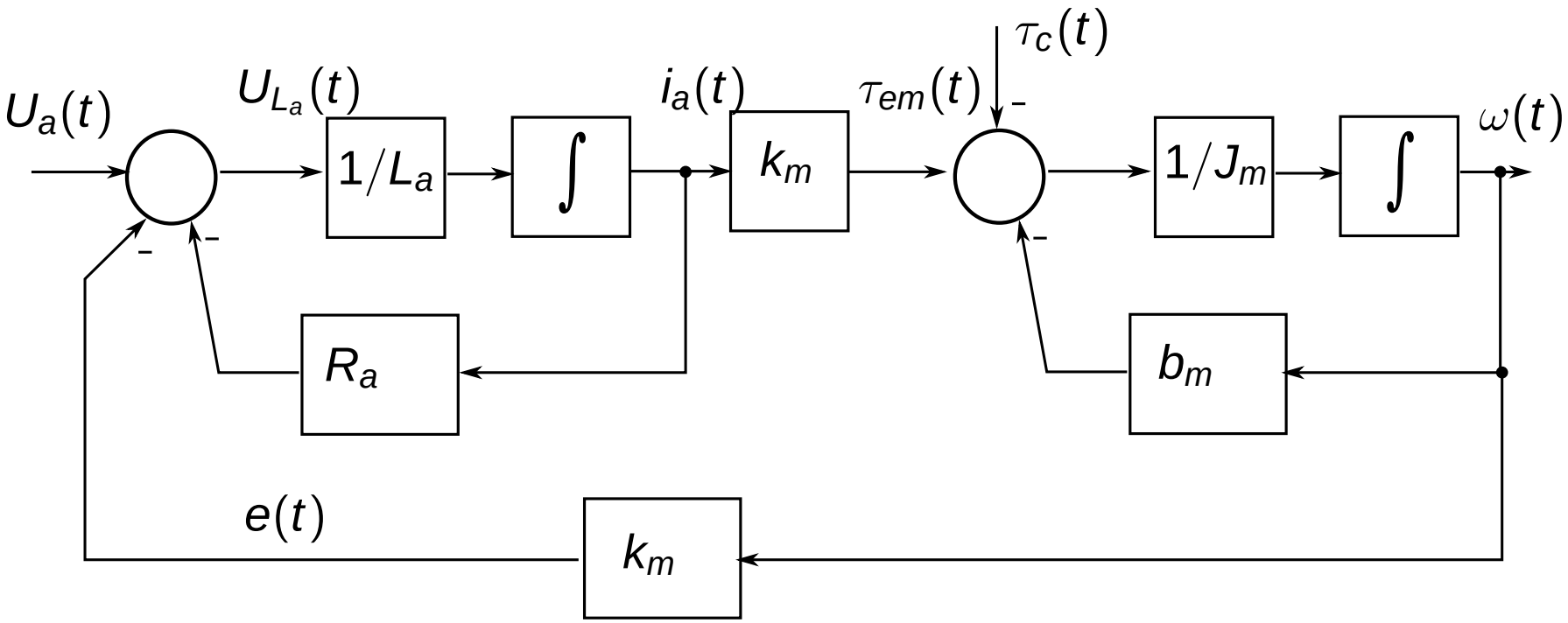
Los Diagramas de Bloques (DBs) simplifican el trabajo de manipulación algebraica de las relaciones constitutivas y estructurales.

Ejemplo: Motor de Corriente Continua



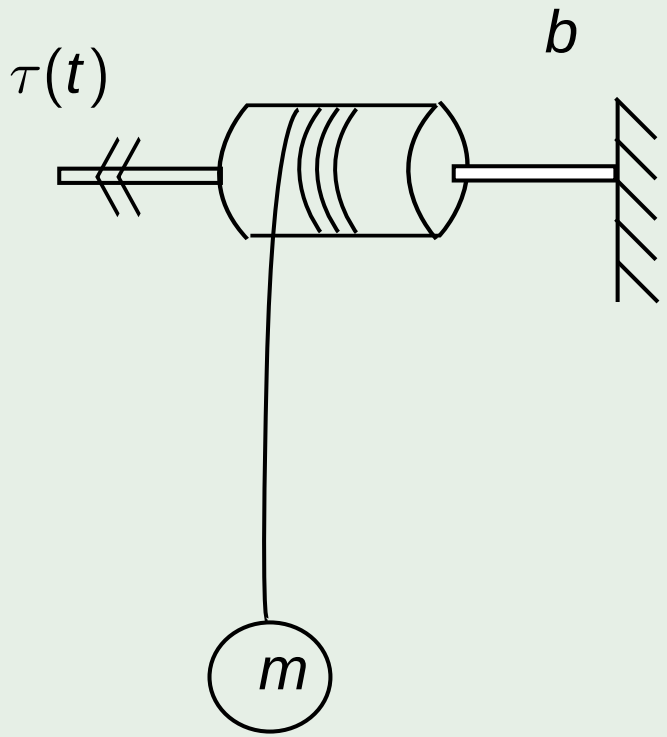
Modelado con Diagramas de Bloques

Diagrama de Bloques del Motor de Corriente Continua



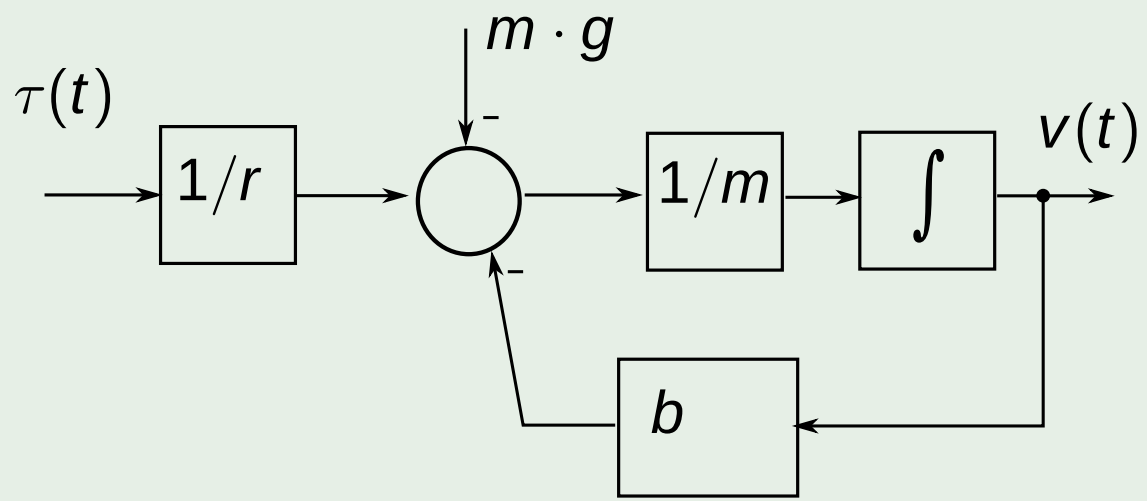
Modelado con Diagramas de Bloques

Sistema Roto–Translacional: Polea



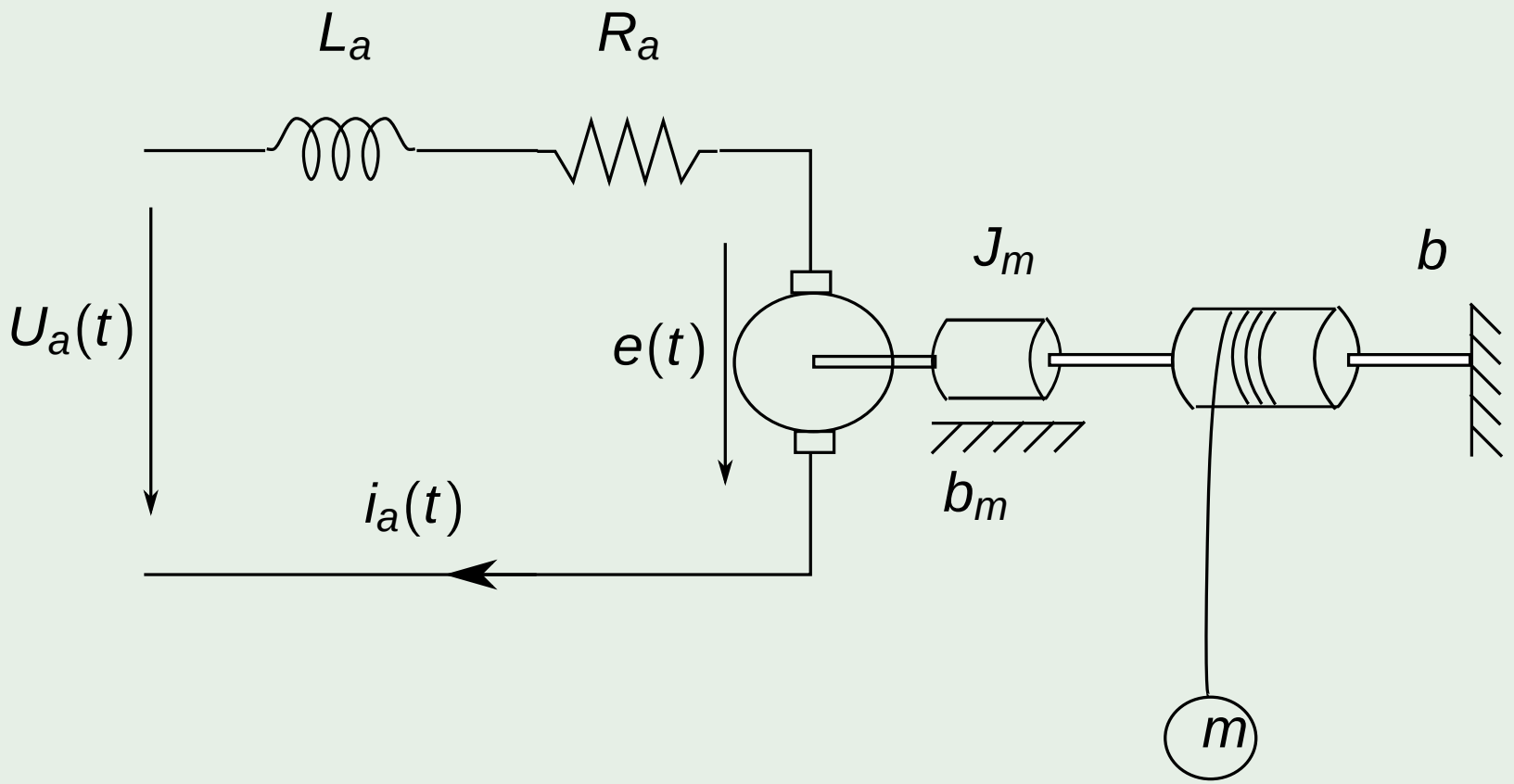
Modelado con Diagramas de Bloques

DB del Sistema Roto–Translacional



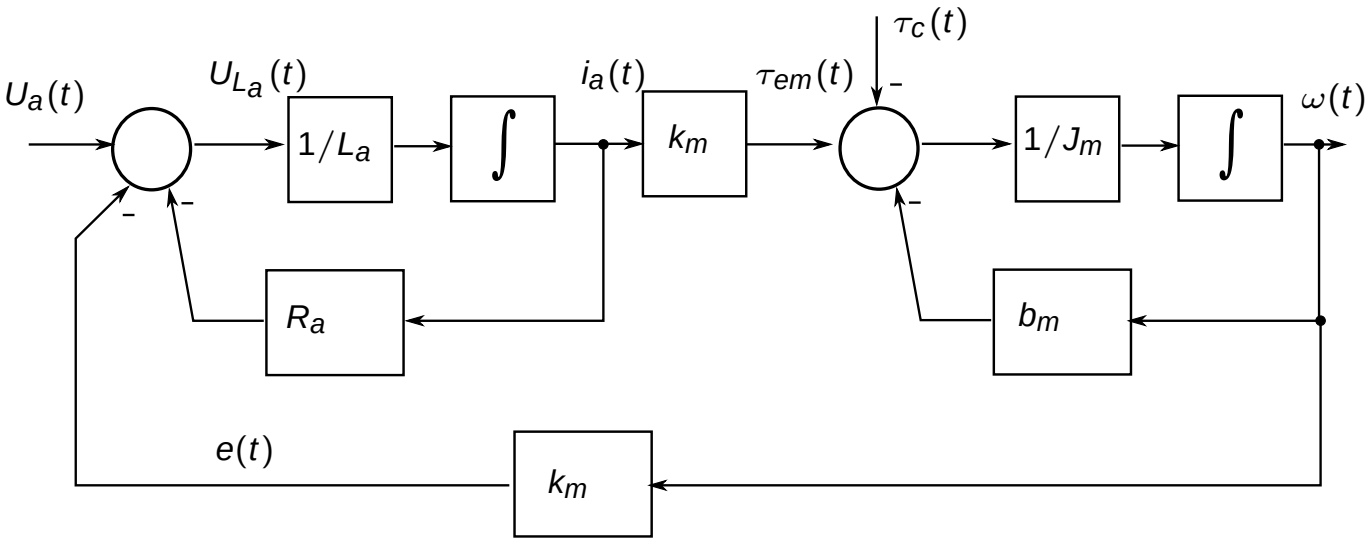
Modelado con Diagramas de Bloques

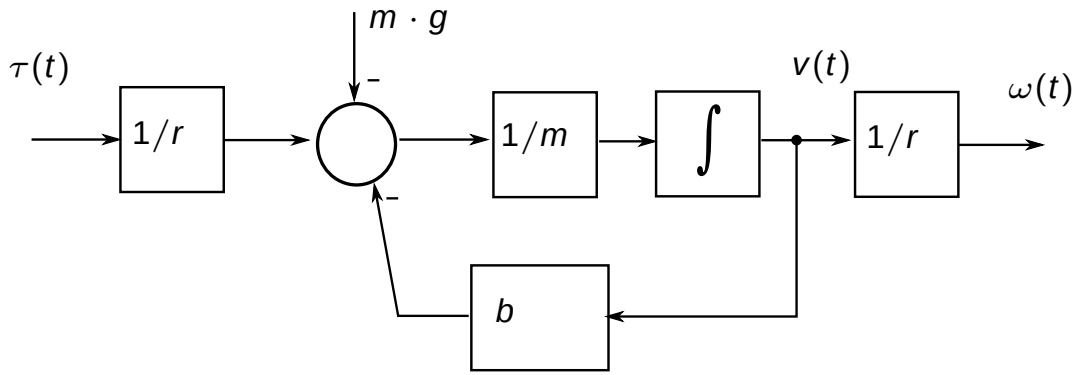
Ejemplo: Motor de Corriente Continua + Polea



¿Podemos construir el DB acoplando los Diagramas de Bloques del Motor y de la Polea?

Modelado con Diagramas de Bloques





Limitaciones de los Diagramas de Bloques

- Además de establecer relaciones matemáticas, los DB establecen relaciones causales entre las variables.
- Debido a esto, cada subsistema ve las variables de interacción como variables de entrada o de salida.
- Si dos subsistemas ven sus variables de interacción con la misma causalidad, se genera un conflicto causal y no se pueden acoplar los DB.

Un formalismo que permita acoplar subsistemas debe ser acausal.

Organización de la Presentación

Bond Graphs
Introducción

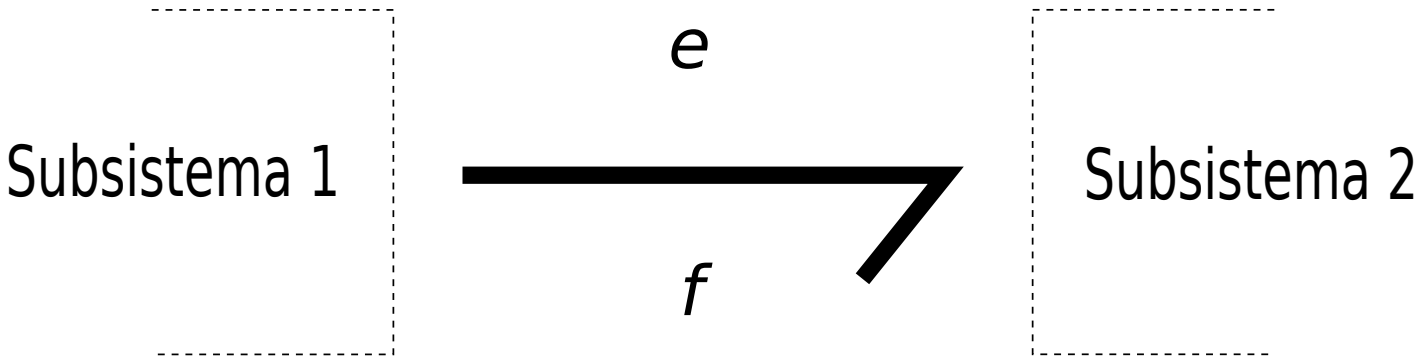
Los sistemas físicos se caracterizan por el intercambio de energía entre subsistemas. El lenguaje gráfico Bond Graph es una herramienta para obtener modelos matemáticos a partir de dicho intercambio, representando de manera unificada el flujo de potencia instantánea, los fenómenos de transformación, almacenamiento y disipación de energía y las relaciones estructurales.

El lenguaje Bond Graph es:

- acausal,
- orientado a objetos,
- aplicable a distintos dominios de la física: mecánica, hidráulica, electromagnetismo, termodinámica–química.

Bond Graphs
Enlaces (Bonds)

El flujo de energía (potencia) entre dos subsistemas se representa mediante un arpón denominado Enlace (Bond):



- Cada enlace tiene dos variables asociadas, cuyo producto es la potencia (*e* por esfuerzo y *f* por *flujo*),
- La punta del arpón indica el sentido positivo del flujo de potencia (en este caso la potencia es positiva si fluye del subsistema 1 al subsistema 2).

Bond Graphs

Variables de potencia generalizadas

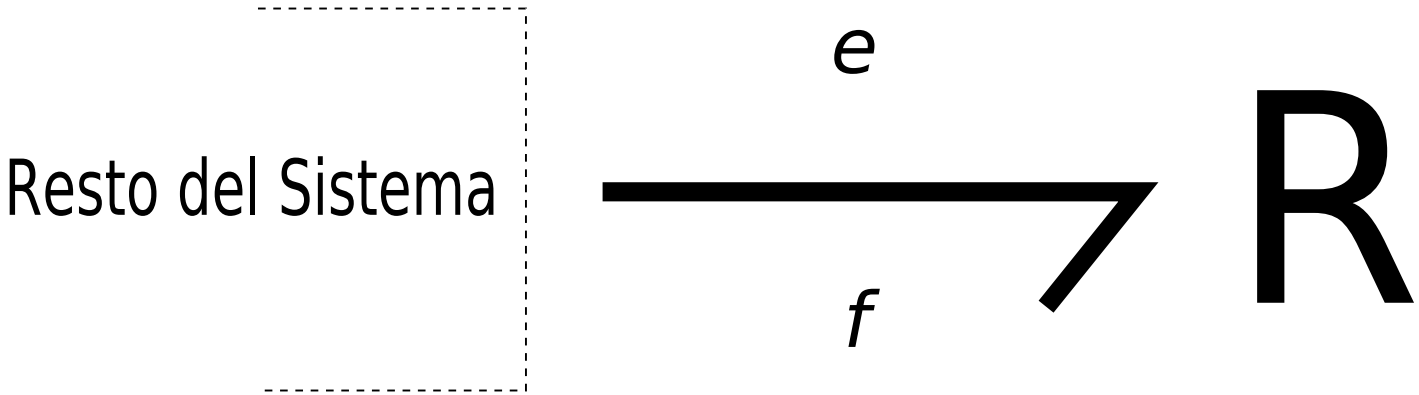
Las variables de esfuerzo (e) y flujo (f) se denominan variables generalizadas de potencia, y se corresponden con las siguientes magnitudes físicas:

- Mecánica traslacional: $e(t) = F(t)$ (fuerza), $f(t) = v(t)$ (velocidad lineal).
- Mecánica rotacional: $e(t) = \tau(t)$ (torque), $f(t) = \omega(t)$ (velocidad angular).
- Electromagnetismo: $e(t) = V(t)$ (voltaje), $f(t) = i(t)$ (corriente).
- Hidráulica: $e(t) = P(t)$ (presión), $f(t) = Q(t)$ (caudal).
- Termodinámica: $e(t) = T(t)$ (temperatura absoluta), $f(t) = \dot{S}(t)$ (flujo de entropía).

Bond Graphs

Disipación de Energía: Resistores

La disipación de energía en Bond Graphs se representa mediante el elemento **R** (resistor):

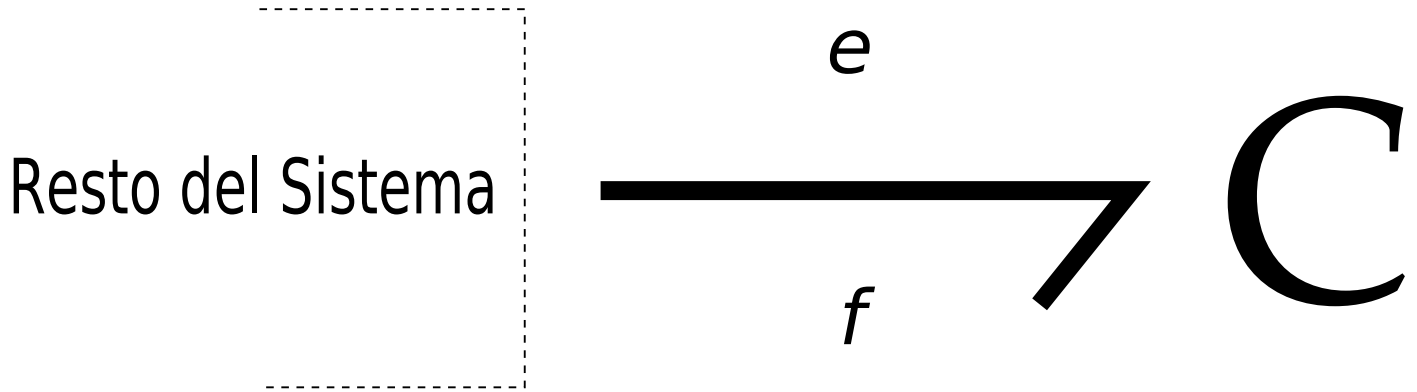


- Un elemento **R** puede representar rozamiento (dominio mecánico), resistencia eléctrica, resistencia hidráulica, etc.
- El resistor está caracterizado por una relación constitutiva del tipo $g(f) - e = 0$.
- En el caso lineal queda $R \cdot f(t) - e(t) = 0$.

Bond Graphs

Almacenamiento de Energía: Capacitor

El almacenamiento de energía se representa en ciertos casos mediante el elemento **C** (capacitor):

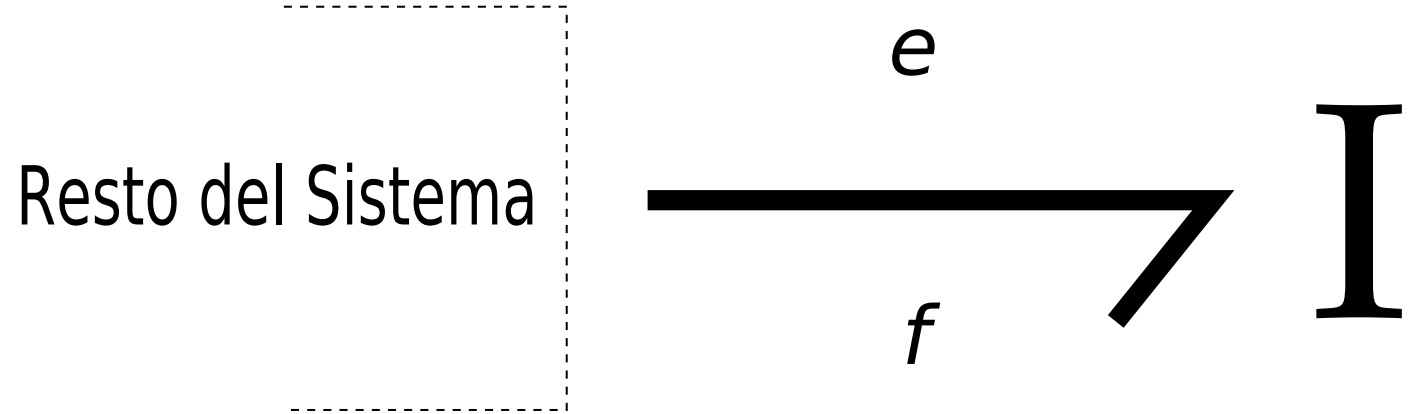


- El elemento **C** puede representar almacenamiento de energía potencial (elástica o gravitatoria), de energía en el campo eléctrico y de energía calórica.
- El capacitor se caracteriza por una relación constitutiva del tipo $g(q) - e = 0$, con $\dot{q}(t) \triangleq f(t)$. La variable de energía $q(t)$ se denomina desplazamiento generalizado.

Bond Graphs

Almacenamiento de Energía: Inercia

El almacenamiento de energía se representa en otro caso mediante el elemento **I** (inercia):

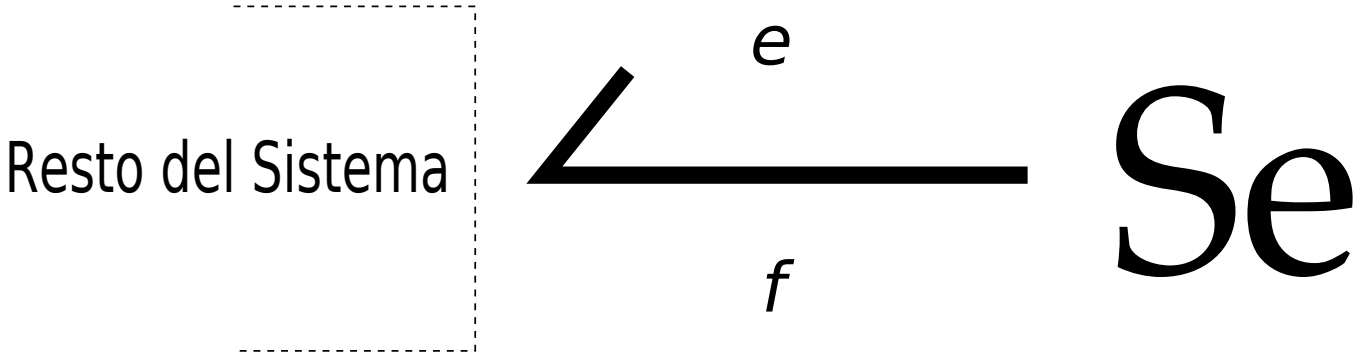


- El elemento **I** puede representar almacenamiento de energía cinética (mecánica e hidráulica) y de energía en el campo magnético.
- La inercia se caracteriza por una relación constitutiva del tipo $g(p) - f = 0$, con $\dot{p}(t) \triangleq e(t)$. La variable de energía $p(t)$ se denomina impulso generalizado.

Bond Graphs

Generación de Energía: Fuente de Esfuerzo

La generación de energía (incorporación desde el exterior) se representa en ciertos casos mediante el elemento **Se** (fuente de esfuerzo).

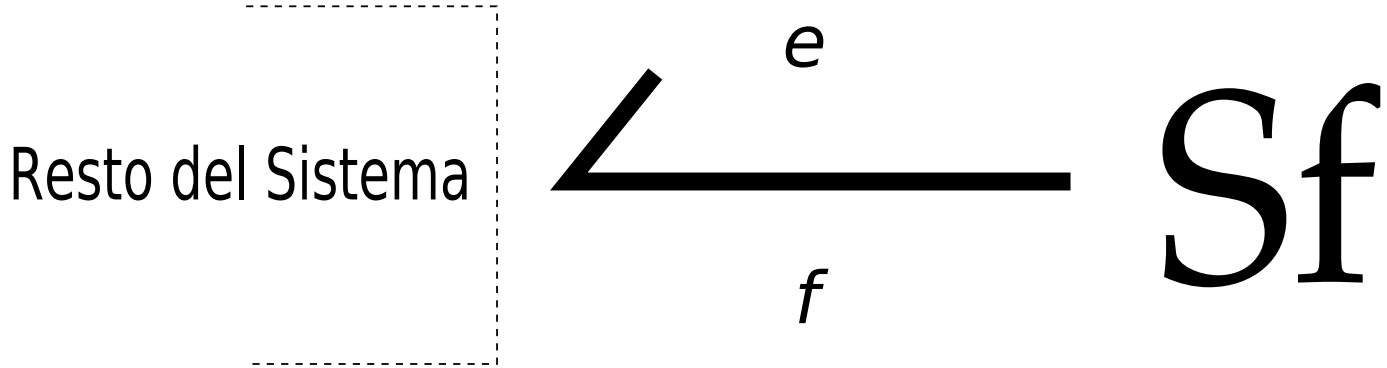


- El elemento **Se** puede representar el ingreso de energía por presencia de una fuente ideal de tensión, de fuerza, de presión, de temperatura, etc.
- La fuente de esfuerzo se caracteriza por una relación constitutiva en la que el esfuerzo $e(t)$ es independiente de lo que ocurra en el sistema.

Bond Graphs

Generación de Energía: Fuente de Flujo

La generación de energía (incorporación desde el exterior) se representa en otros casos mediante el elemento **Sf** (fuente de flujo).

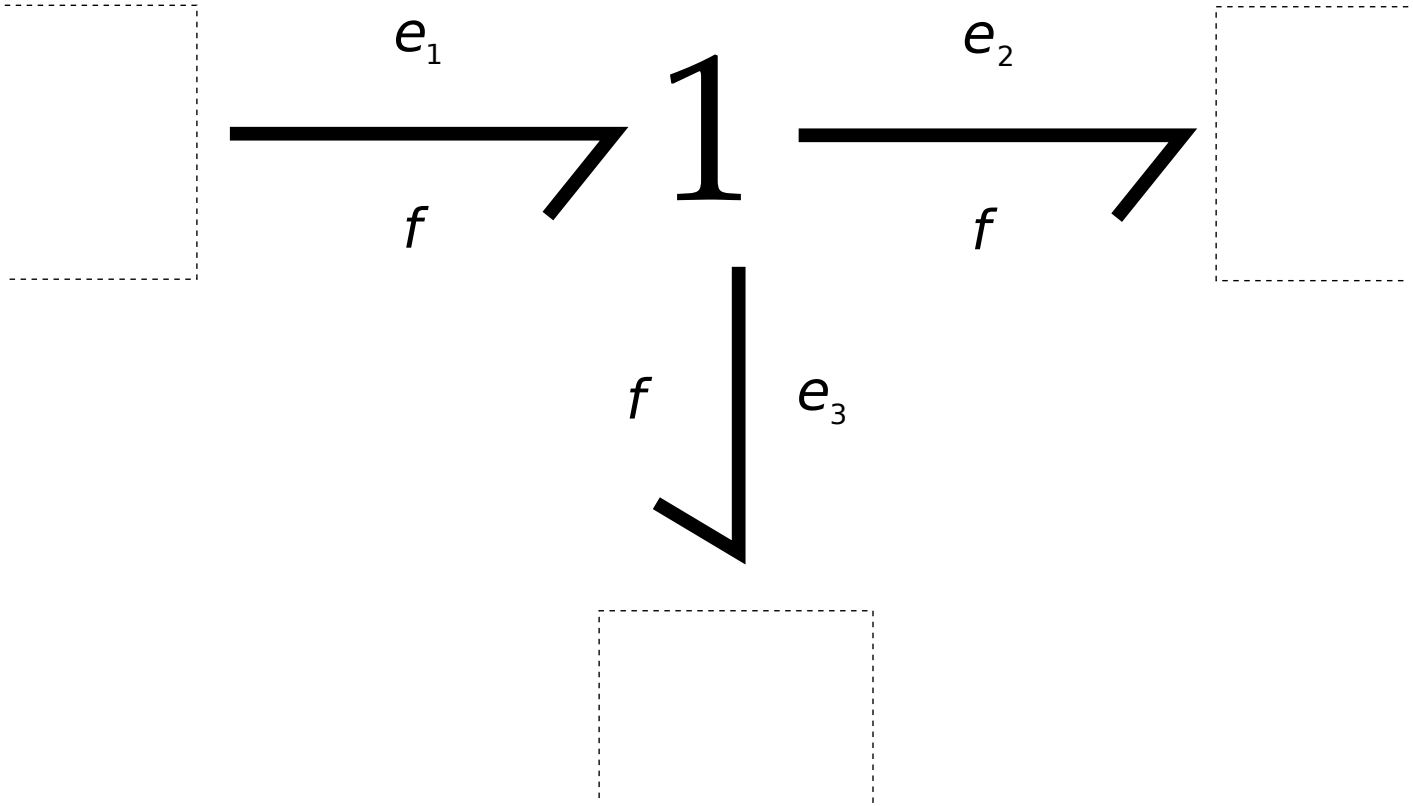


- El elemento **Sf** puede representar el ingreso de energía por presencia de una fuente ideal de corriente, de velocidad, de caudal, etc.
- La fuente de flujo se caracteriza por una relación constitutiva en la que el flujo $f(t)$ es independiente de lo que ocurra en el sistema.

Bond Graphs

Elementos Estructurales: Vínculo 1

El vínculo **1** representa la distribución de energía con el flujo como variable común:



Bond Graphs

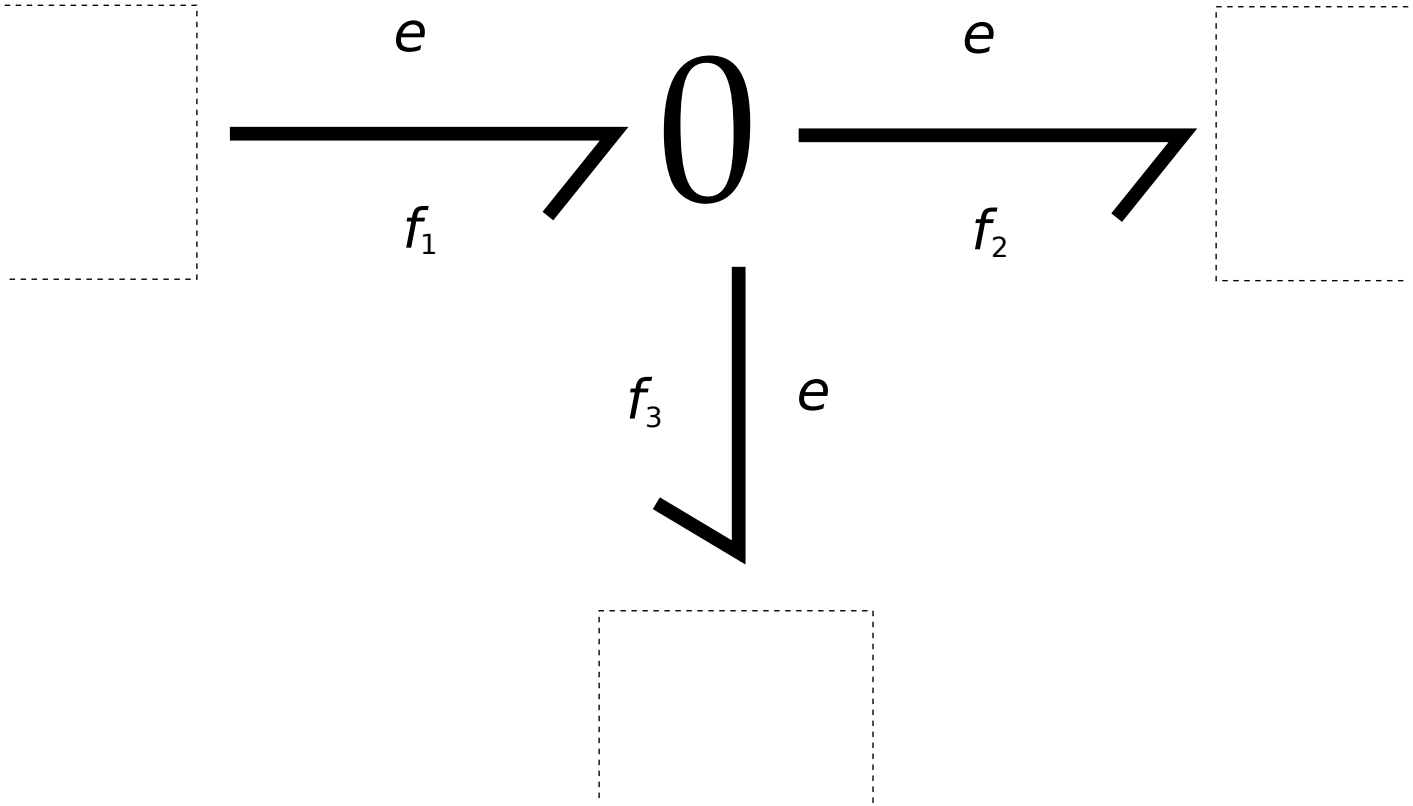
Elementos Estructurales: Vínculo 1

- El vínculo **1** puede representar una conexión serie eléctrica, un objeto mecánico puntual, etc.
- Es un elemento multipuerta, ya que puede conectarse a varios arpones.
- Siendo un vínculo estructural, el vínculo **1** no puede almacenar ni generar ni disipar energía, es decir, debe ocurrir que $\sum \pm P_i(t) = \sum \pm e_i(t) \cdot f_i(t) = 0$.
- Debido a lo anterior, debe cumplirse que $\sum \pm e_i(t) = 0$.

Bond Graphs

Elementos Estructurales: Vínculo 0

El vínculo **0** representa la distribución de energía con el esfuerzo como variable común:

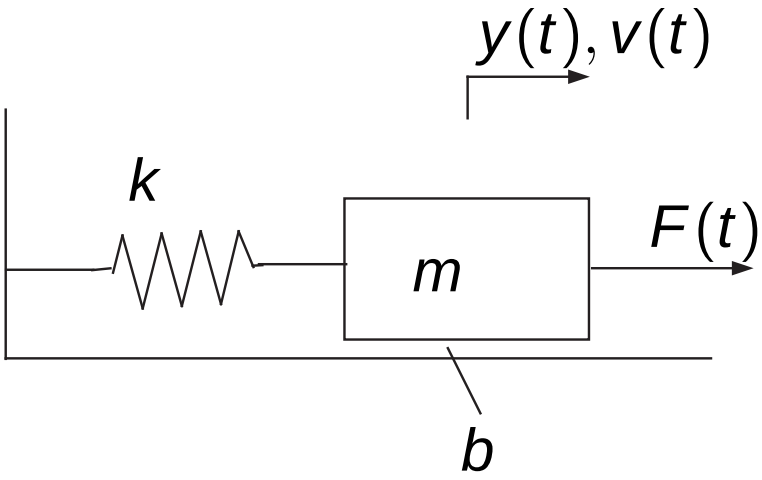


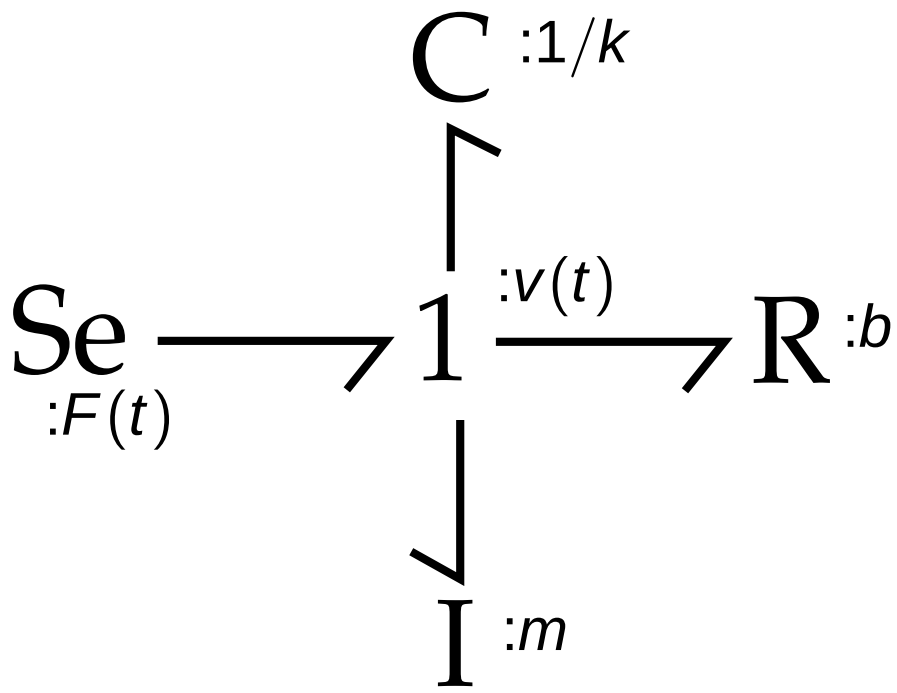
Bond Graphs

Elementos Estructurales: Vínculo 0

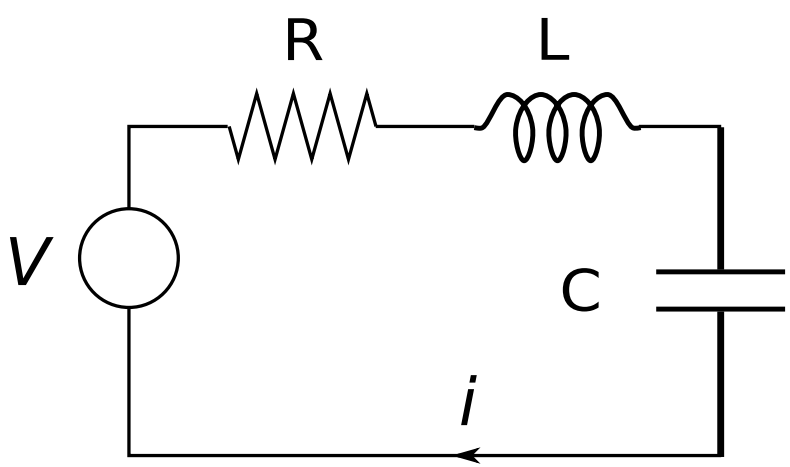
- El vínculo **0** puede representar una conexión paralela eléctrica, una conexión serie mecánica, un nodo en un circuito eléctrico o hidráulico, etc.
- Es un elemento multipuerta, ya que puede conectarse a varios arpones.
- Siendo un vínculo estructural, el vínculo **0** no puede almacenar ni generar ni disipar energía, es decir, debe ocurrir que $\sum \pm P_i(t) = \sum \pm e_i(t) \cdot f_i(t) = 0$.
- Debido a lo anterior, debe cumplirse que $\sum \pm f_i(t) = 0$.

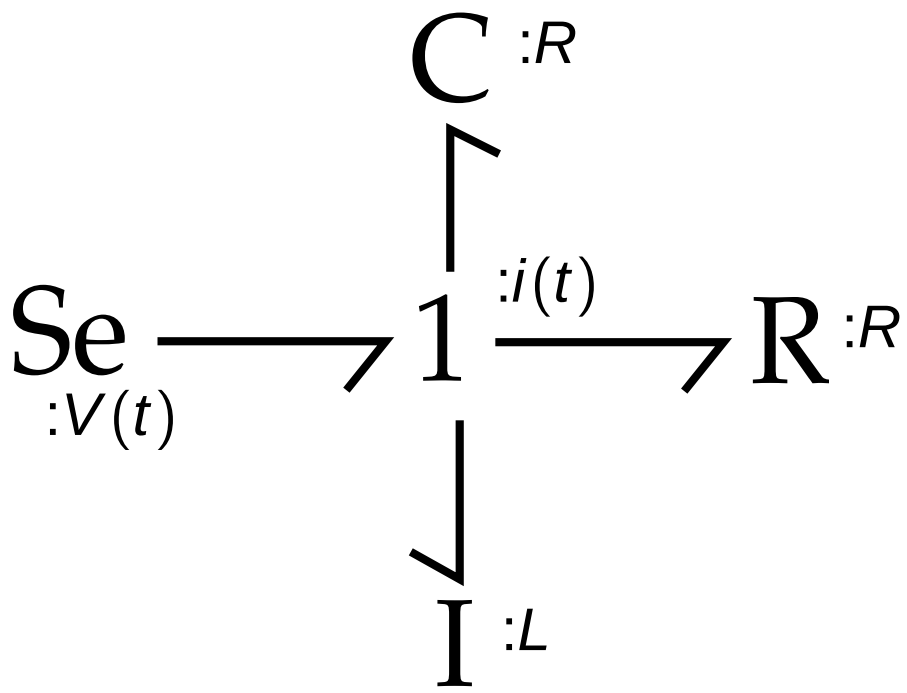
Ejemplo: Sistema Masa Resorte



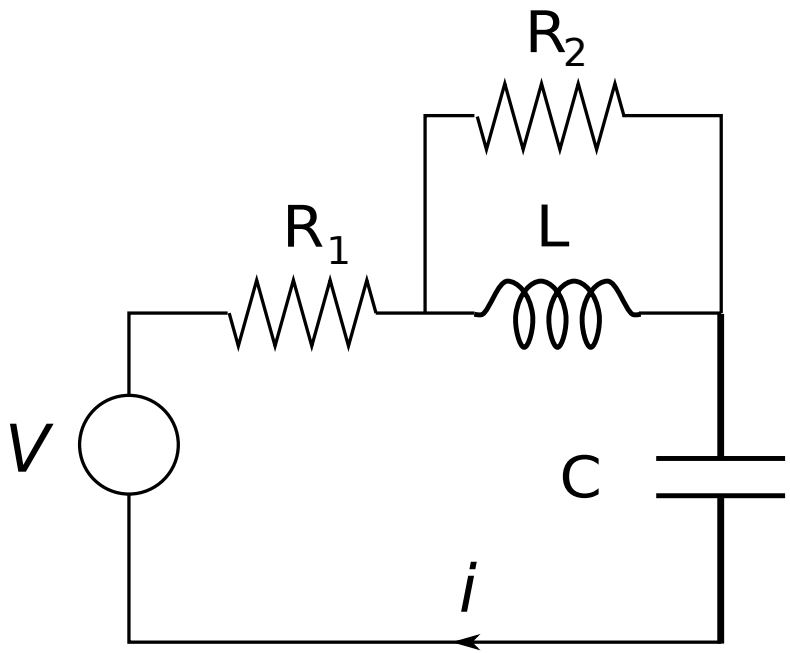


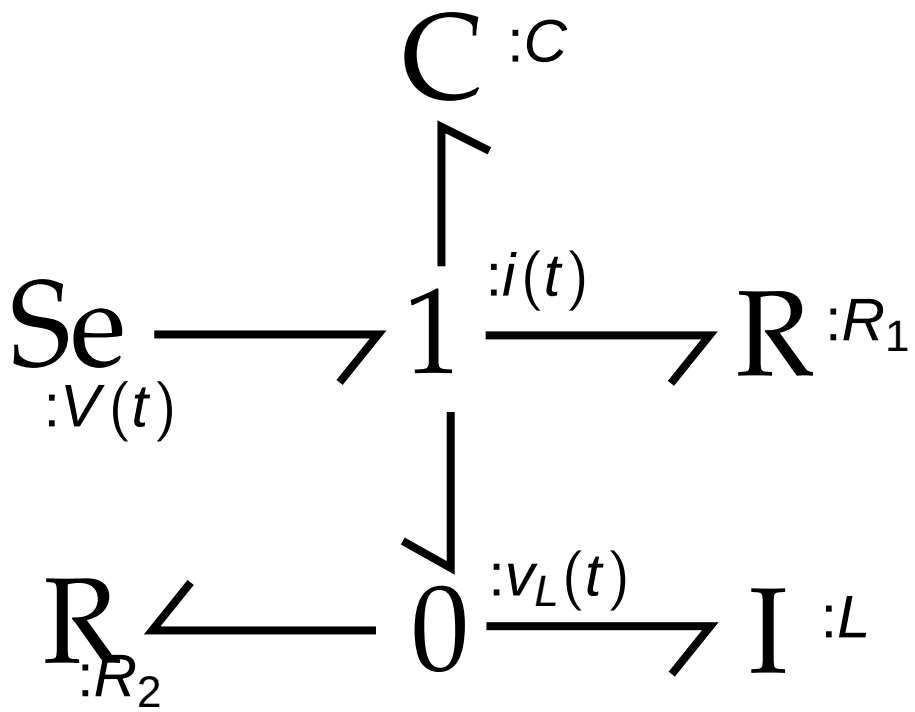
Ejemplo: Circuito RLC Serie





Ejemplo: Circuito RLC Modificado

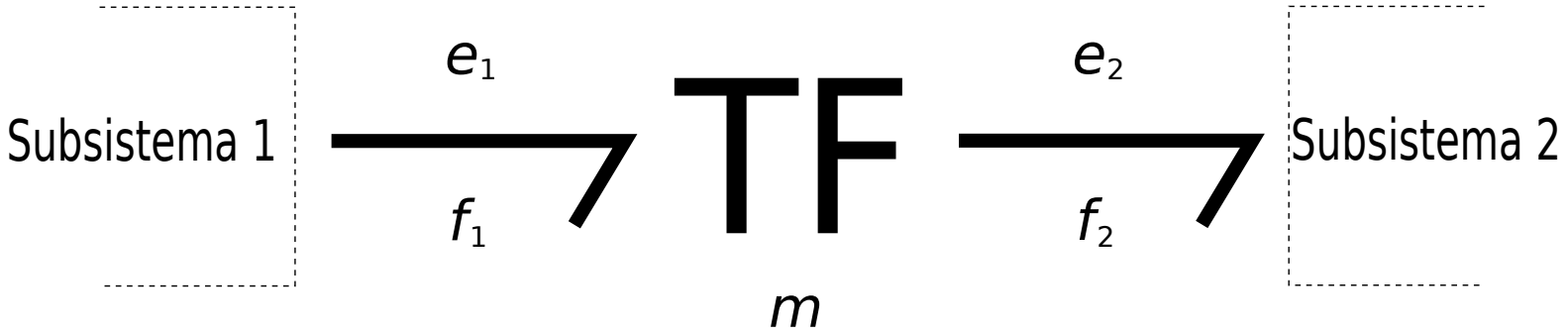




Bond Graphs

Elementos Estructurales: Transformador

El acople entre subsistemas también suele darse de forma tal que ambas variables (esfuerzo y flujo) se ven afectadas. En algunos casos, esto se representa mediante un transformador:



- Las relaciones a ambos lados del transformador son $m \cdot f_1(t) - f_2(t) = 0$ y $m \cdot e_2(t) - e_1(t) = 0$.
- El elemento **TF** puede representar poleas, palancas, tranformadores eléctricos, pistones, engranajes, etc.

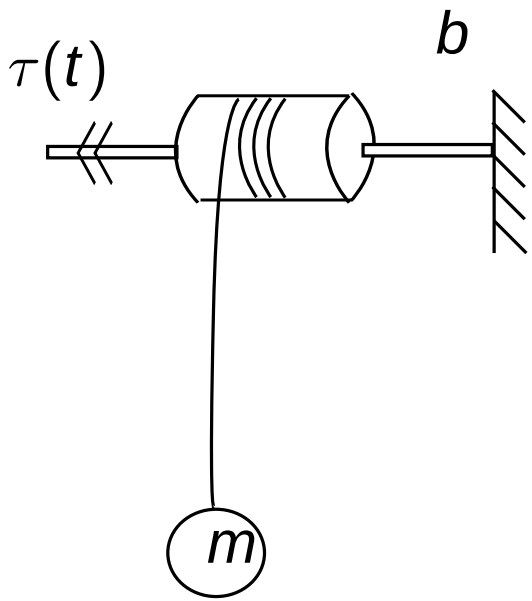
Bond Graphs

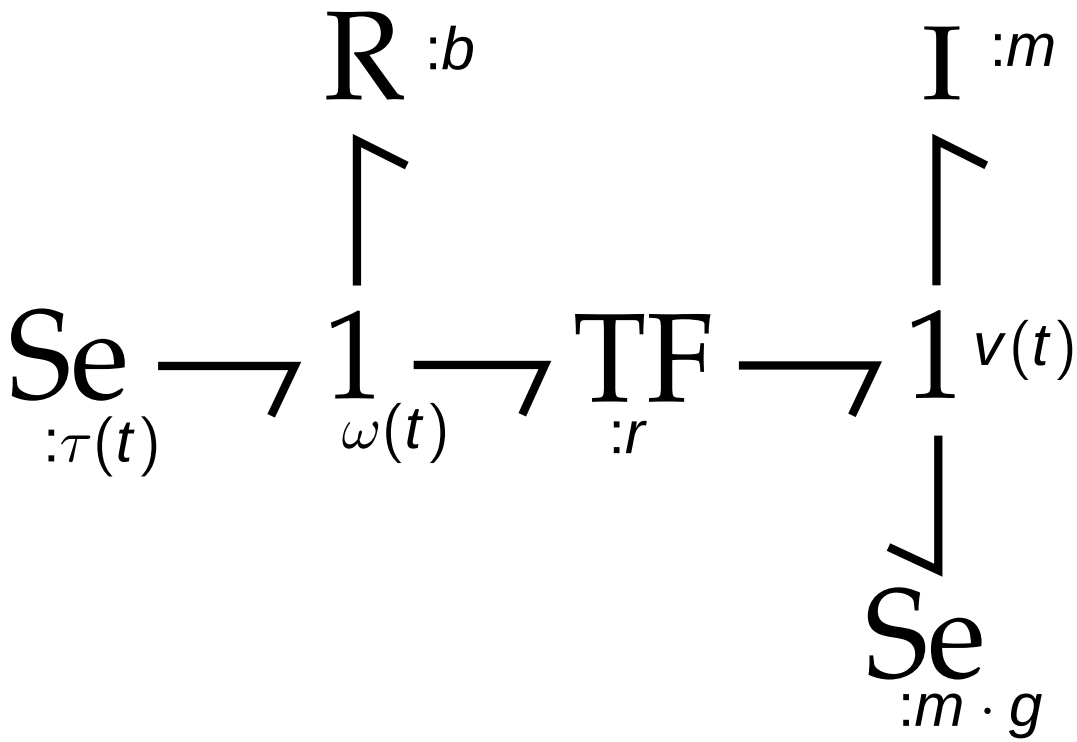
Elementos Estructurales: Girador

Similar al transformador, pero cuando las variables se relacionan de manera cruzada, el elemento estructural se denomina girador:

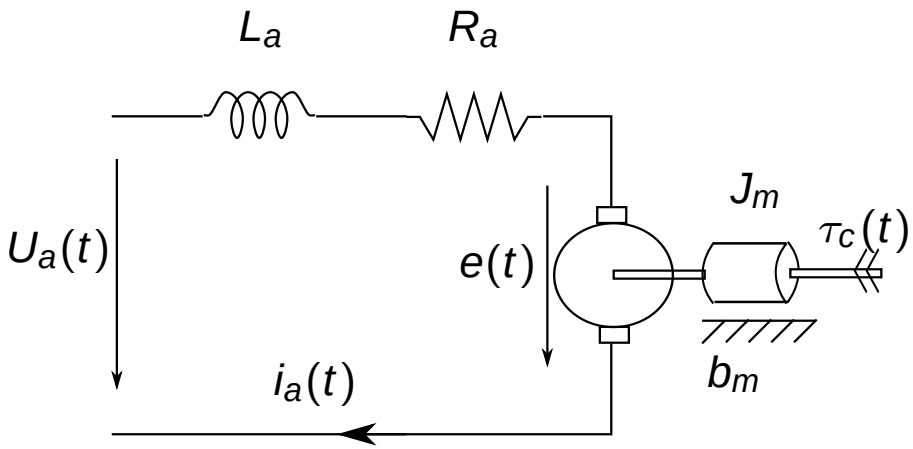
- Las relaciones a ambos lados del girador son $m \cdot f_1(t) - e_2(t) = 0$ y $m \cdot f_2(t) - e_1(t) = 0$.
- El elemento **GY** puede representar la conversión electromecánica de energía (motores y generadores) así como giróscopos, etc.

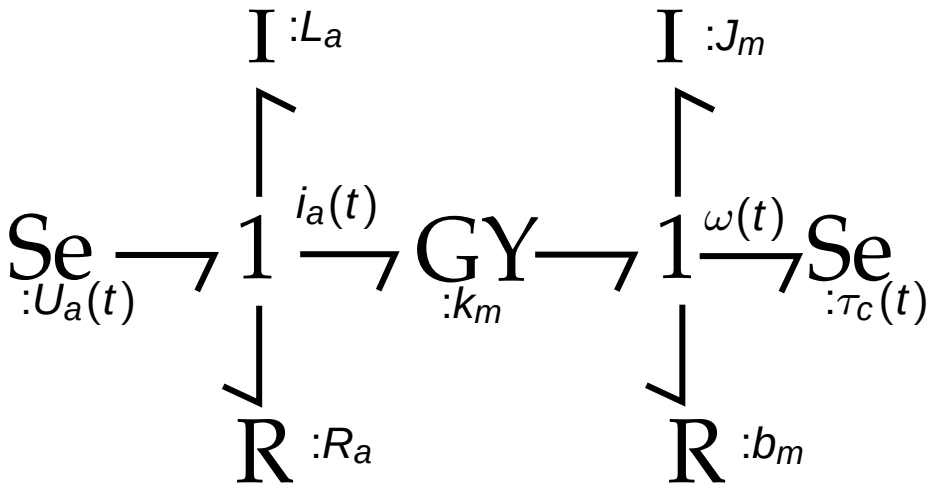
Ejemplo: Polea



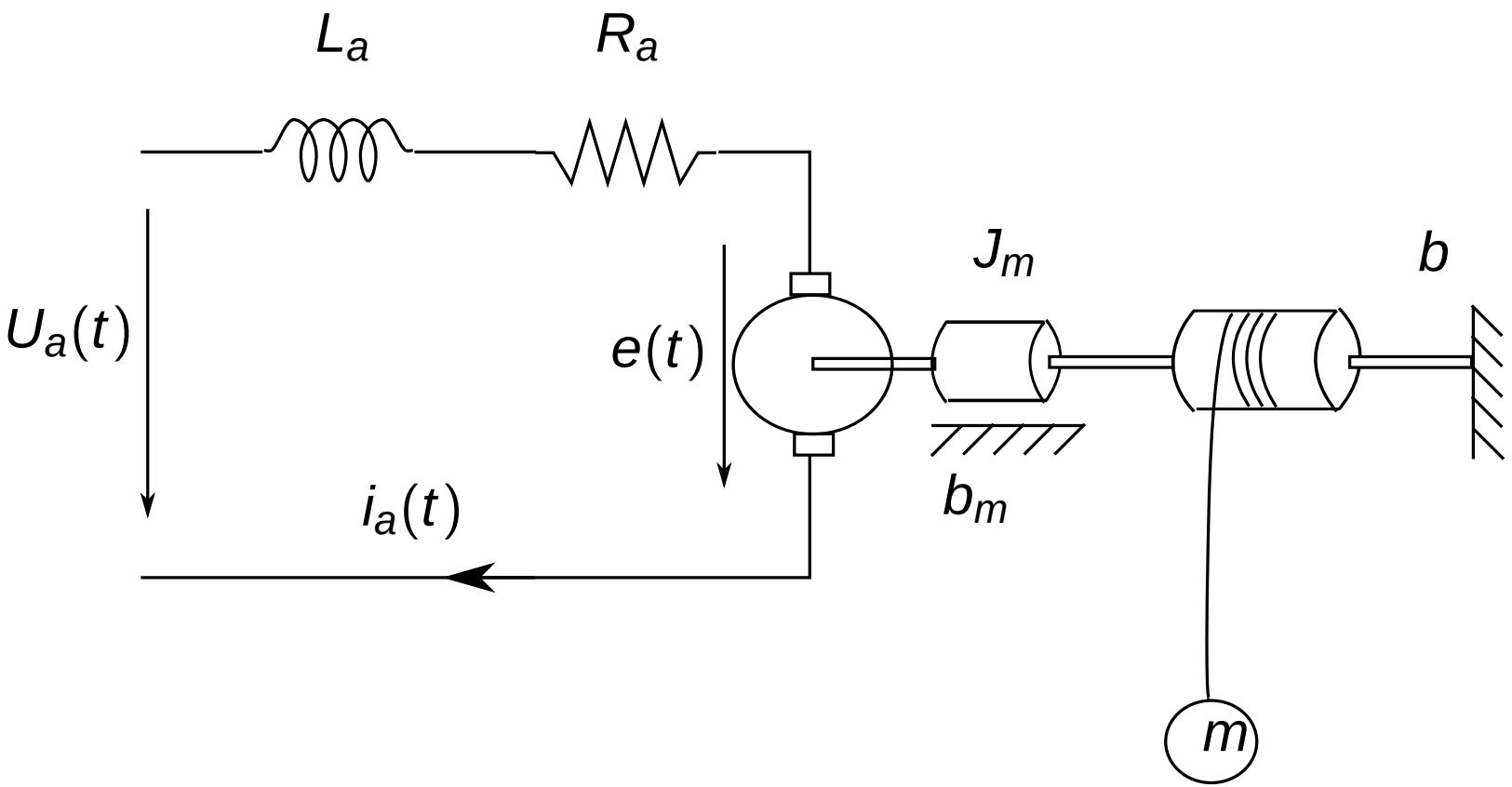


Ejemplo: Motor

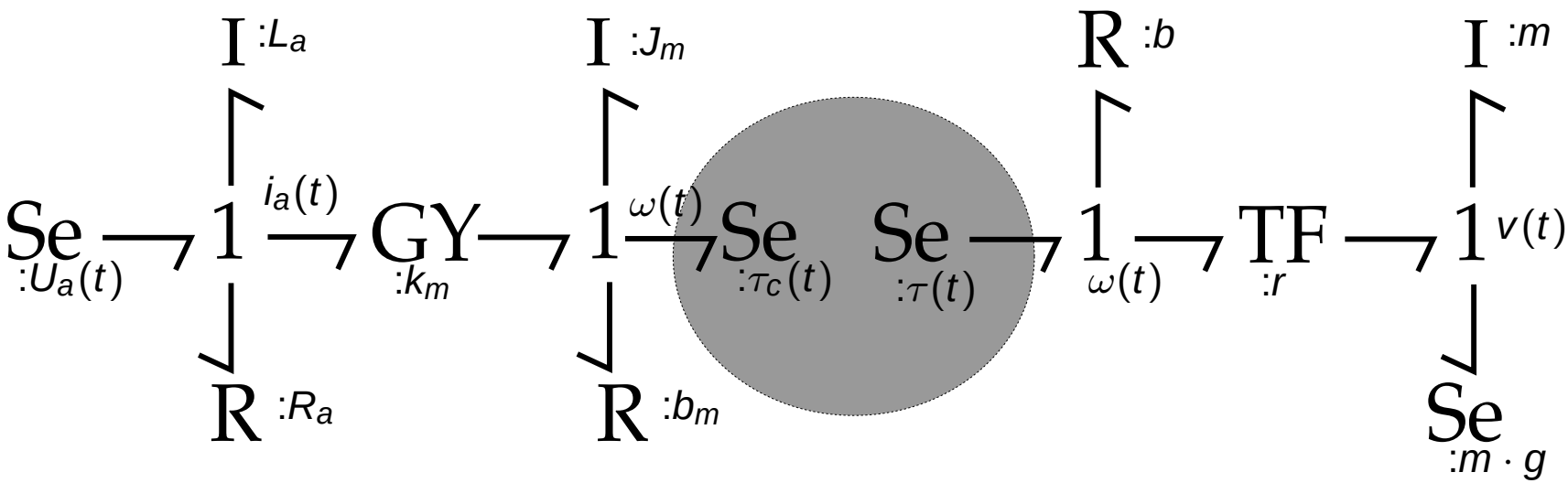




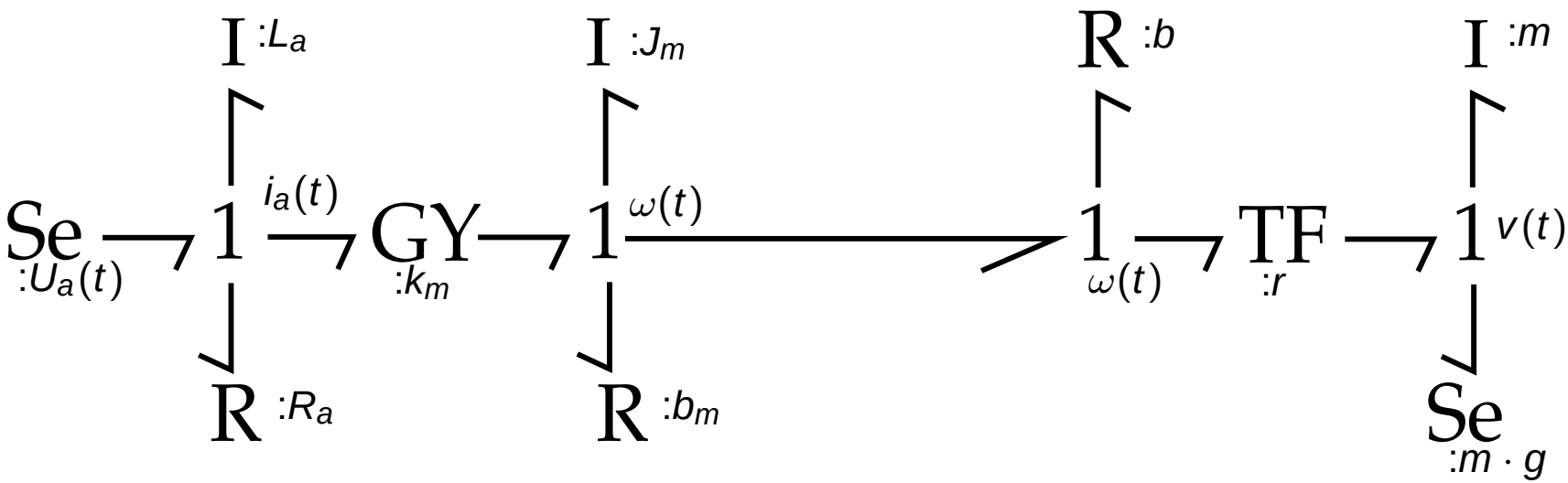
Ejemplo: Motor y polea



Ejemplo: Motor y polea



Ejemplo: Motor y polea



Modelo Matemático definido por un Bond Graphs

- El Bond Graphs anterior contiene todas las ecuaciones del sistema de manera acausal.
- La ausencia de causalidad es lo que permite acoplar subsistemas de manera trivial.
- Es posible, de manera totalmente algoritmizable, obtener a partir del Bond Graph un Diagrama de Bloques y/o las Ecuaciones de Estado correspondientes.
- Hay varias herramientas de software que permiten manipular y simular Bond Graphs: 20Sim, Dymola, PowerDynaMo.

Limitaciones de los Bond Graphs

- El lenguaje sólo permite modelar fenómenos de intercambio energético. Cualquier interacción no energética se representa con Diagramas de Bloque, dando lugar a Diagrams Mixtos. En estos casos, puede aparecer nuevamente el problema de la causalidad.
- Las variables que entran en juego en las ecuaciones son fijas (e, f, p, q), lo que puede resultar un problema. En robótica, por ejemplo, conviene trabajar con velocidades relativas (así se reduce enormemente el número de ecuaciones y el costo computacional de su manipulación).
- Estas limitaciones motivaron el desarrollo de un lenguaje más general de modelado denominado Modelica.

Organización de la Presentación

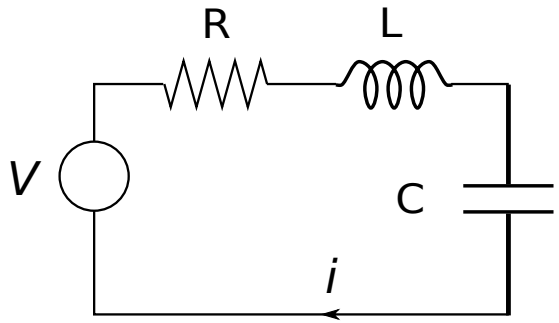
Introducción a Modelica

Modelica es un lenguaje estandarizado para el modelado de sistemas continuos, discretos e híbridos. Tiene las siguientes características:

- Es orientado a objetos, y cada componente de un modelo es una instancia de una clase.
- Permite una definición trivial de las relaciones constitutivas y estructurales de cada sub–modelo.
- Permite definir relaciones acausales o causales según sea necesario.
- Facilita el reuso de componentes, sub-modelos y modelos mediante la creación de paquetes y librerías.
- Cuenta con una librería estándar muy completa.
- Admite una representación gráfica totalmente intuitiva para los distintos dominios de la física y de la técnica.

Introducción a Modelica

Ejemplo: Circuito RLC usando la Librería Estándar




```
model RLC_Circuit
  Modelica.Electrical.Analog.Basic.Ground Ground1
  Modelica.Electrical.Analog.Basic.Resistor Resistor1
  Modelica.Electrical.Analog.Basic.Inductor Inductor1
  Modelica.Electrical.Analog.Basic.Capacitor Capacitor1
  Modelica.Electrical.Analog.Sources.ConstantVoltage ConstantVoltage1
equation
  connect(Resistor1.n, Inductor1.p)
  connect(Inductor1.n, Capacitor1.p)
  connect(Capacitor1.n, Ground1.p)
  connect(ConstantVoltage1.p, Resistor1.p)
  connect(ConstantVoltage1.n, Ground1.p)
end RLC_Circuit;
```

Introducción a Modelica

Algunas clases eléctricas de la Librería Estándar

```
 model Inductor
  extends Interfaces.OnePort;
  parameter SI.Inductance L=1;
equation
  L*der(i) = v;
end Inductor;
```

```
partial model OnePort
    SI.Voltage v;
    SI.Current i;
    PositivePin p;
    NegativePin n;
equation
  v = p.v - n.v;
  0 = p.i + n.i;
  i = p.i;
end OnePort;
```

- El comando extends explicita la herencia.
- der() indica la derivada.
- las ecuaciones son todas acausales.
- las clases SI.Voltage, etc., son variables reales provistas de unidades de medida.

Introducción a Modelica

Algunas clases eléctricas de la Librería Estándar

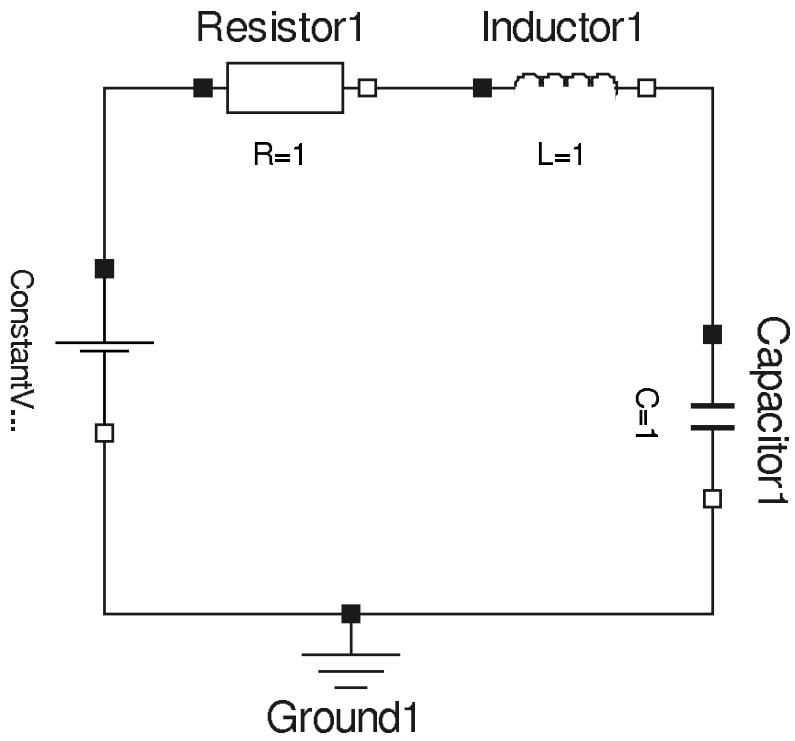
```
connector PositivePin
  SI.Voltage v;
  flow SI.Current i;
end PositivePin;

connector NegativePin
  SI.Voltage v;
  flow SI.Current i;
end NegativePin;
```

- Las clases tipo connector permiten vincular variables mediante la ecuación connect.
- Todas las variables tipo flow de conectores conectados entre sí suman cero.
- Todas las demás variables de conectores conectados entre sí son iguales.

Introducción a Modelica

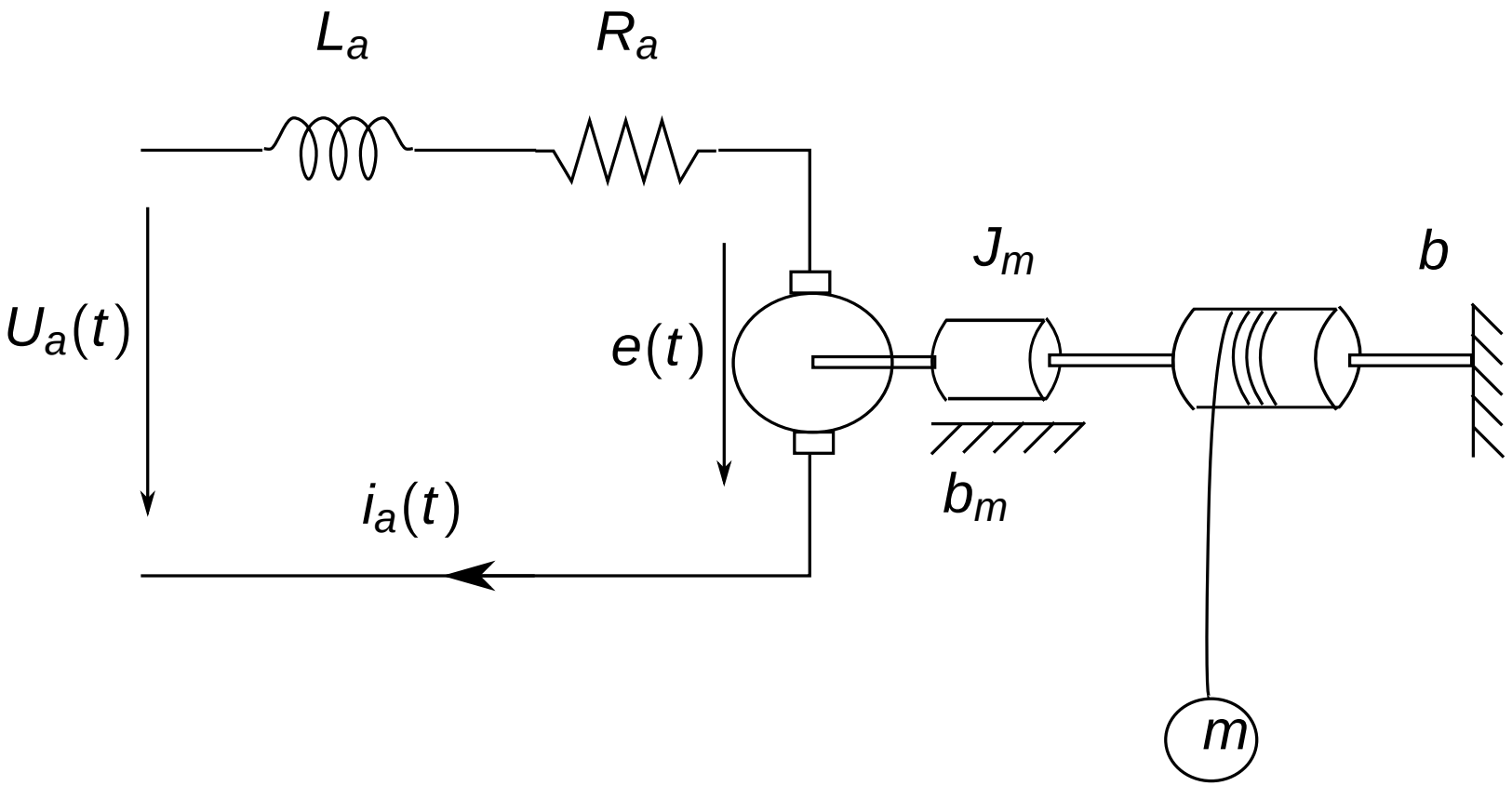
Representación gráfica de modelos

Hay varias interfaces gráficas para la edición de modelos. Dymola es una de ellas (MathModelica y SimForge son otras alternativas).



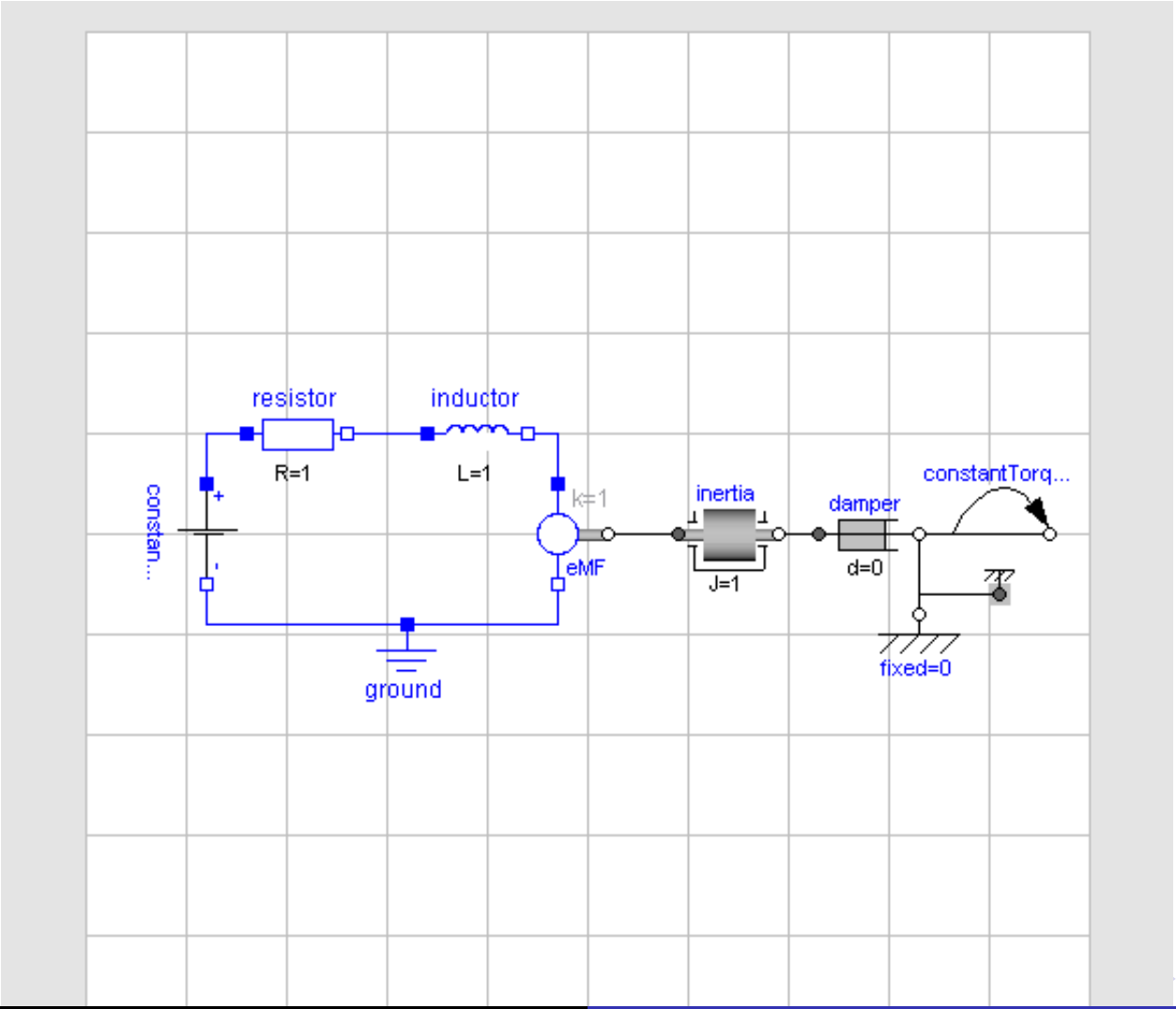
Introducción a Modelica

Representación gráfica de modelos



Introducción a Modelica

Representación gráfica de modelos



Introducción a Modelica

Algunas características adicionales

- El lenguaje Modelica permite representar Bond Graphs, Diagramas de Bloques y sistemas de ecuaciones diferenciales en general de manera trivial.
- En virtud de lo anterior, Modelica es una generalización de los formalismos de modelado existentes.
- Además, el lenguaje cuenta con funcionalidades que permiten representar sistemas de tiempo discreto y de eventos discretos, que pueden a su vez interactuar con subsistemas continuos dando lugar a sistemas híbridos.

Introducción a Modelica
Compilación y Simulación de Modelos

- Para simular modelos de Modelica estos deben convertirse primero en Ecuaciones de Estado. Esta conversión la llevan a cabo los Compiladores de Modelica.
- Hay actualmente tres compiladores desarrollados: Dymola, Mathmodelica y Open Modelica.
- El proceso de compilación de un modelo complejo de Modelica es bastante complicado, debido a la presencia frecuente de singularidades estructurales (Ecuaciones Diferenciales Algebraicas).

Bibliografía


Modelica Association.
Language Specification - Modelica - A Unified Object-Oriented Language for Physical Systems Modeling.
Linköping, Sweden, 2005.
Disponible en www.modelica.org.


François Cellier.
Continuous System Modeling.
Springer Verlag, New York, 1991.


Dean Karnopp, Donald Margolis, and Ronald Rosenberg.
System Dynamics. Modeling and Simulation of Mechatronic Systems.
John Wiley & Sons, Hoboken, New Jersey, 4th. edition, 2006.


Ernesto Kofman.
Introducción a Modelica.
Notas de Clase, 2007.
Disponible en www.fceia.unr.edu.ar/control/modelica.