

# Modelado y Simulación de Sistemas Dinámicos

## Trabajo Práctico 3 Modelica

El objetivo de este trabajo es modelar un sistema continuo complejo, esquematizado en la Figura 1, consistente en una bomba *aspirante-impelente* impulsada por un *motor de corriente continua* a través de un sistema *biela-manivela* y utilizada para subir agua a un tanque.

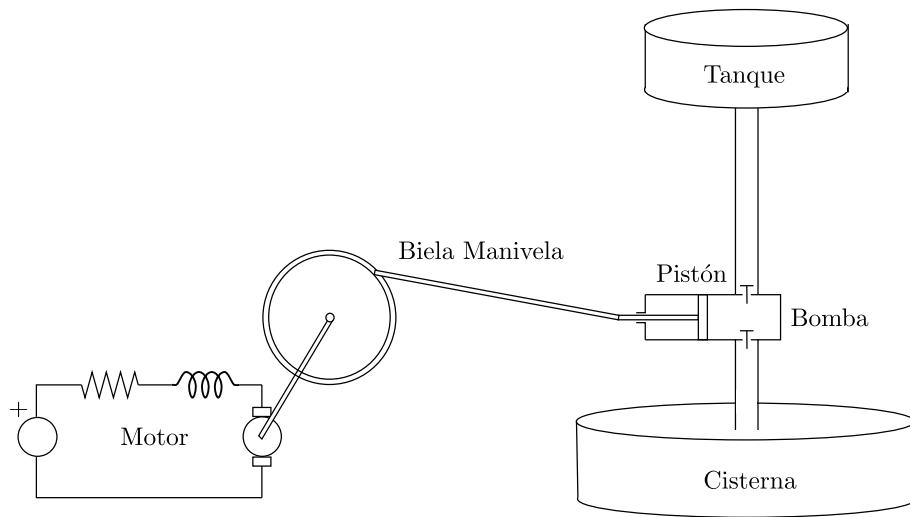


Figura 1: Sistema de Bombeo de Agua

Para formular el modelo utilizando el lenguaje Modelica, construiremos pequeñas librerías de los distintos dominios físicos involucrados (eléctrico, mecánico traslacional, mecánico rotacional e hidráulico), y algunos elementos de interfaz entre los distintos dominios. Con estas librerías iremos construyendo subsistemas y finalmente armaremos el modelo completo.

Si bien habrá muchísimas alternativas para construir los modelos, se pretende respetar los nombres propuestos para todas las clases de manera que los componentes se puedan eventualmente compartir entre los distintos grupos que realizan el trabajo. Los paquetes correspondientes a cada librería, además, deberán colocarse dentro de un paquete llamado **TPModelica**.

No es necesario resolver todo el trabajo. Cada alumno podrá optar por resolver los siguientes puntos:

- Las secciones 1, 2 y 5.
- Las secciones 2, 3 y 6.
- Las secciones 3, 4 y 7.

Finalmente, para realizar el problema completo (Sección 8), se podrán utilizar las librerías hechas por otros alumnos.

# 1. Sistema Eléctrico

Para representar la parte puramente eléctrica del modelo, construiremos una librería de modelos eléctricos y luego la utilizaremos para hacer un modelo simple.

## 1.1. Librería Eléctrica

Llamaremos a esta librería **Electrical** y contará con las siguientes clases:

### 1.1.1. Tipos

Definiremos los tipos **Voltage**, cuya unidad es voltios (V), y **Current**, cuya unidad es el amperio (A).

### 1.1.2. Conectores

Llamaremos **Pin** al conector eléctrico, que contará con un voltaje **v** y una corriente **i**, y esta última será la variable de flujo.

### 1.1.3. Modelos

La librería contendrá los siguiente componentes:

- **ConstVoltage**, que representa una fuente de tensión constante, y tendrá la siguiente definición:

```
model ConstVoltage
  Pin p,n;
  parameter Voltage V=1;
equation
  p.v-n.v=V;
  p.i+n.i=0;
end ConstVoltage;
```

- **Ground**, que representa un punto a tierra y tendrá un único conector **Pin p** cuyo voltaje deberá ser 0, es decir  $p.v=0$ .
- Una clase parcial que representa dipolos llamada **OnePort** definida como sigue:

```
partial model OnePort
  Pin p,n;
  Voltage v;
  Current i;
equation
  v=p.v-n.v;
  i= p.i;
  i= -n.i;
end OnePort;
```

- Un modelo llamado **Resistor**, heredado de la clase **OnePort**, que representa una resistencia eléctrica con parámetro  $R=1$  y ley constitutiva  $v=R*i$ .
- Un modelo llamado **Inductor**, también heredado de **OnePort**, que representa una inductancia con parámetro  $L=1$  y ley constitutiva  $L*der(i)=v$ .

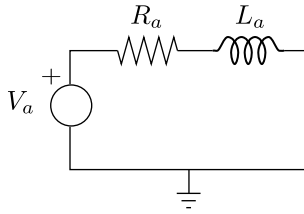


Figura 2: Circuito RL Serie

## 1.2. Ejemplo Eléctrico

Agregaremos a la librería el circuito de la Figura 2, que cuenta con una fuente de tensión constante `ConstVoltage`, un resistor, un inductor y una tierra, que se llamarán `va`, `ra`, `la` y `gr` respectivamente. Los parámetros respectivos serán `V=12`, `R=1` y `L=0.001`.

Al modelo lo construiremos en dos partes, primero haremos un modelo llamado `RLOpen` que contendrá los todos componentes y las conexiones de `va` a `gr`, de `va` a `ra` y de `ra` a `la`, pero no tendrá conexión desde el segundo conector del inductor.

Luego haremos un modelo llamado `RL` que contendrá el modelo anterior agregando la conexión faltante. Finalmente, simularemos este último sistema.

## 2. Sistema Mecánico Rotacional

Construiremos para esta parte una librería mecánica rotacional y la probaremos con un ejemplo.

### 2.1. Librería Rotacional

La librería llamada `Rotational` contará con las siguiente clases:

#### 2.1.1. Tipos

Definiremos los tipos que representarán el torque, el ángulo y la velocidad angular como sigue:

```
type Torque=Real(unit="N.m");
type Angle=Real(unit="rad");
type AngSpeed= Real(unit="rad/s");
```

### 2.2. Conectores

El conector, llamado `Flange`, contendrá dos variables: el ángulo llamado `th` y el torque llamado `tau`, este último será de tipo flujo.

### 2.3. Modelos

Esta librería tendrá los siguientes componentes:

- Una inercia rotacional `Inertia` con un conector definida como sigue

```
model Inertia
  Flange flange;
  AngSpeed w;
  parameter Real J=1;
equation
  J*der(w)=flange.tau;
  der(flange.th)=w;
end Inertia;
```

- Un componente de fricción **Friction** con un conector, similar al modelo anterior pero con parámetro **b=1** y ley `flange.tau=b*w`.
- Una fuente de torque constante **ConstTorque** con un conector, un parámetro **Torque Tau=1** y una ecuación `flange.tau=-Tau`.

## 2.4. Ejemplo Rotacional

Agregaremos a la librería un modelo consistente en una inercia (llamada **inertia**) conectada a una fricción, al que llamaremos **FricInertia**.

Luego, conectaremos la inercia de dicho modelo a una fuente de torque constante, creando un modelo que llamaremos **FricInertiaTau**, que simularemos observando la velocidad de la inercia.

## 3. Mecánica Traslacional

Para el subsistema traslacional haremos también una pequeña librería y un ejemplo.

### 3.1. Librería Traslacional

La librería llamada **Translational** contará con las siguiente clases:

#### 3.1.1. Tipos

Definiremos al menos los siguientes tipos:

```
type Position = Real(unit="m");
type Distance = Real(unit="m");
type Velocity = Real(unit="m/s");
type Force = Real(unit="N");
```

#### 3.1.2. Conectores

Utilizaremos un conector **Flange** definido como sigue:

```
connector Flange
  Position s;
  flow Force f;
end Flange;
```

#### 3.1.3. Modelos

La librería contará con los siguientes componentes:

- Una masa puntual, definida como sigue:

```
model PointMass
  Flange flange;
  Position s;
  Velocity v;
  parameter Real m=1;
equation
  s=flange.s;
  v=der(s);
  m*der(v)=flange.f;
end PointMass;
```

- Una clase llamada **Fixed** representando un punto fijo con un conector **Flange flange** y con la ecuación  $\text{flange.s}=\text{s0}$ , donde el parámetro  $\text{s0}=0$  es la posición del punto.
- Una clase parcial llamada **Compliant** con dos conectores y definida como sigue

```
partial model Compliant
  Flange flange1, flange2;
  Distance srel;
  Force f;
equation
  srel=flange2.s-flange1.s;
  flange2.f=f;
  flange1.f=-f;
end Compliant;
```

- Un modelo de resorte heredado del anterior y llamado **Spring**, con ley  $f=k*(\text{srel}-\text{srel0})$  donde  $k=1$  y  $\text{srel0}=0$  son parámetros.
- Un modelo de amortiguador heredado también de **Compliant** y llamado **Damper**, con ley  $f=b*\text{der}(\text{srel})$  donde  $b=1$  es un parámetro.
- Un modelo de Fuente senoidal denominado **SinForce** y definido como sigue

```
model SinForce
  Flange flange;
  parameter Force F=1;
  parameter Real f=1;
equation
  flange.f=-F*sin(6.2832*f*time);
end SinForce;
```

### 3.2. Ejemplo Traslacional

Agregaremos a la librería **Translational** un modelo de un sistema masa–resorte–amortiguador como el de la Figura 3

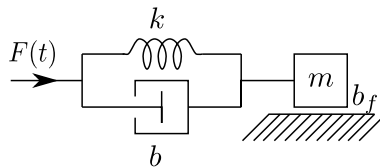


Figura 3: Sistema Masa Resorte Amortiguador

Para esto, en primer lugar representaremos la fricción con el piso mediante un modelo **Damper** con parámetro  $b=100$  conectado entre la masa y un punto fijo. Conectaremos los restantes componentes (sin incluir la fuente de fuerza) con parámetros  $m=0.1$ ,  $k=10000$ ,  $b=100$ . Este modelo se llamará **SpringDamperMass**.

Luego, armaremos el modelo **SpringDamperMassForce** agregando la fuente de fuerza senoidal y conectándola al resorte del modelo anterior. La fuente tendrá parámetro  $F=500$ . Simularemos este modelo observando la posición de la masa y la deformación del resorte.

## 4. Sistema Hidráulico

Para la parte hidráulica haremos también una librería y un modelo de ejemplo.

### 4.1. Librería Hidráulica

Construiremos una librería llamada `Hydraulic` que tendrá las siguientes clases:

#### 4.1.1. Tipos

Los tipos utilizados serán los siguientes:

```
type Volume=Real(unit="m^3");
type Flow=Real(unit="m^3/s");
type Pressure=Real(unit="Pa");
```

#### 4.1.2. Conectores

El conector hidráulico se llamará `FluidPort` y tendrá como variables la presión `Pressure p` y el caudal `flow Flow q` (esta última tipo flujo).

#### 4.1.3. Componentes

Tendremos además los siguientes componentes:

- Una fuente de presión constante llamada `ConstPressure`, con un conector `FluidPort port`, un parámetro `Pressure P=0` y una ley `port.p=P`.
- Una fuente de presión senoidal llamada `SinPressure`, idéntica a la anterior pero con ley `port.p=P*sin(6.2832*f*time)`, donde `f=1` es un parámetro.
- Una clase parcial con dos puertos definida como sigue:

```
partial model TwoPort
  FluidPort port1,port2;
  Pressure prel;
  Flow q;
equation
  prel=port1.p-port2.p;
  q=port1.q;
  q=-port2.q;
end TwoPort;
```

- Un modelo de válvula hidráulica llamada `Valve`, heredado de la clase anterior, con ley `prel=R*q` donde `R=1` es un parámetro (resistencia hidráulica).
- Un modelo de válvula hidráulica de un sólo sentido, llamada `OneWayValve` idéntico al anterior pero con ley `prel=if q>0 then q*R else q*Roff`, donde `Roff=1e9` es un parámetro (resistencia de fuga).
- Un modelo de columna vertical de agua, denominado `WaterColumn`, también heredado de `TwoPort`, pero con ley `prel=h*g*rho`, donde el parámetro `h=1` es la altura de la columna, la constante `g=9.8` es la aceleración de la gravedad y el parámetro `rho=997` es la densidad del agua.
- Un modelo de tanque denominado `Tank`, con un único conector, y definido como sigue:

```
model Tank
  FluidPort port;
  Volume vol;
  parameter Real A=1;
  constant Real g=9.8;
```

```

    constant Real rho=997;
equation
    port.p*A=vol*rho*g;
    der(vol)=port.q;
end Tank;

```

El parámetro A es el área de la base del tanque.

## 4.2. Ejemplo Hidráulico

Agregaremos a la librería un modelo que represente la parte hidráulica del sistema de la Fig.1, reemplazando el efecto del pistón por una fuente de presión.

Para esto, en primer lugar construiremos un modelo que llamaremos **PumpTank** y que tenga los siguientes componentes:

- La cisterna, representada por una fuente de presión cero **ConstPressure** llamada **constpressure**.
- Dos válvulas de un sentido **OneWayValve**, la primera llamada **valve1** y la segunda **valve2**. En ambas, utilizar el parámetro  $R=1e5$ .
- Una columna de agua **WaterColumn** denominada **wc** de 3 mts de altura.
- Un tanque **Tank** denominado **tank**.

Para que el modelo tenga la estructura de la Fig.1, deberá conectarse la cisterna **constpressure** con el primer conector de **valve1**. Luego, el segundo conector de **valve1** con el primero de **valve2** y el segundo conector de **valve2** con el primero de **wc**. Finalmente, se debe conectar el segundo conector de **wc** con el tanque.

Construiremos luego un modelo denominado **Pumping**, que contendrá un modelo **PumpTank** y una fuente de presión senoidal **SinPressure** con parámetro  $P=50000$ . Además habrá una conexión entre la fuente y el primer conector de la segunda válvula del modelo de la clase **PumpTank**. Simularemos este modelo observando la evolución del volumen del tanque y el caudal por la columna de agua.

## 5. Subsistema Electromecánico

En base a las librerías **Electrical** y **Rotational**, construiremos una librería electromecánica elemental.

### 5.1. Librería Electromecánica

Esta librería, llamada **ElectroMec** contendrá un modelo de conversión ideal de energía Electro-Magnética Mecánica, denominado **ElMecConv**, y que tiene la siguiente definición:

```

model ElMecConv
    extends Electrical.OnePort;
    Rotational.Flange flange;
    Rotational.AngSpeed w;
    parameter Real K=1;
equation
    flange.tau=-K*i;
    v=K*w;
    der(flange.th)=w;
end ElMecConv;

```

Notar que tiene dos conectores eléctricos y un conector mecánico rotacional. Este componente se corresponde con el etiquetado como EMC en la Figura 4.

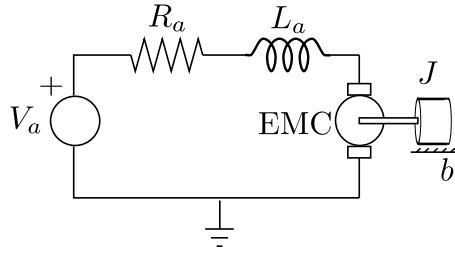


Figura 4: Motor de Corriente Continua

## 5.2. Ejemplo Electromecánico

Agregaremos a la librería Electromecánica el modelo de un motor de corriente continua, como el de la Fig.4.

Este modelo, llamado `DCMotor`, contendrá un modelo `RLOpen` definido en la Sección 1.2 y un modelo `FricInertia` definido en la Sección 2.4, vinculados a través de un modelo `ElMecConv`. Simularemos el ejemplo y observaremos la evolución de la velocidad angular y la corriente.

## 6. Subsistema Roto-Traslacional

En base a las librerías `Rotational` y `Translational` construiremos una librería roto-traslacional elemental.

### 6.1. Librería Roto-Traslacional

La librería, denominada `RotoTranslational`, tendrá un componente correspondiente a un sistema *biela-manivela* como el del centro de la Fig.1. Dicho modelo, denominado `RodCrank`, tiene un conector traslacional y uno rotacional. Las variables son la fuerza transmitida por la biela  $F$ , la distancia horizontal de la punta de la biela al centro de la manivela  $x$  y el ángulo de la manivela  $\theta$ . Los parámetros, en tanto, son el radio de la manivela  $r$ , el largo de la biela  $l$  y la posición del centro de la manivela  $x_0$ .

```
model RodCrank
  Rotational.Flange flangerot;
  Translational.Flange flange;
  Translational.Force F;
  Translational.Position x;
  Rotational.Angle th;
  parameter Real r=0.1;
  parameter Real l=1;
  parameter Real x0=-r-l;
equation
  th=flangerot.th;
  x+x0=flange.s;
  r^2+x^2=l^2-2*r*l*cos(th);
  flange.f=F*sqrt(1-(r*sin(th)/l)^2);
  flangerot.tau=-r*x*sin(th)*F;
end RodCrank;
```

Notar que la posición por defecto del centro de la manivela es tal que la punta de la manivela queda inicialmente en el origen.

### 6.2. Ejemplo Roto-Traslacional

Como ejemplo agregaremos a la librería `RotoTranslational` el modelo de un sistema Biela-Manivela como el de la Figura 5.

Para construir este modelo, conectaremos en primer lugar un modelo `RodCrank` con un modelo `SpringDamperMass` definido en la Sección 3.2 (uniendo el conector traslacional del primero con el primer conector del resorte del otro modelo). Al modelo resultante lo llamaremos `RodCrankMass`.



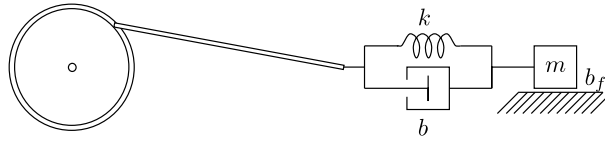


Figura 5: Sistema Biela-Manivela

A partir de este último modelo, construiremos un nuevo modelo denominado **RodCrankMassTau**, en el cual agregaremos una fuente de torque constante **ConstTorque** con parámetro **Tau=0.5** y una inercia **inertia** en el mismo eje, y simularemos el sistema.

## 7. Subsistema Hidráulico-Mecánico

En base a las librerías **Translational** y **Hydraulic** construiremos una librería hidráulico-mecánica elemental.

### 7.1. Librería Hidro-Mecánica

Esta librería, denominada **HydroMec** tendrá un único componente que representa un pistón de área **A**, llamado **Piston** y definido como sigue:

```
model Piston
  Translational.Flange flange;
  Hydraulic.FluidPort port;
  parameter Real A=1;
equation
  flange.f=port.p*A;
  der(flange.s)*A=-port.q;
end Piston;
```

### 7.2. Ejemplos Hidro-Mecánicos

Agregaremos un ejemplo que representará el subsistema desde el pistón hacia la derecha de la Figura 1.

Para esto, en primer lugar formaremos un sistema denominado **PistonPumpTank** conectando un pistón de área **A=0.01** con el sistema **PumpTank** definido en la Sección 4.2.

Luego, conectaremos una fuente de fuerza senoidal **SinForce** de la librería **Translational** con parámetro **F=150** al pistón del **PistonPumpTank**, formando un modelo llamado **PistonPumpTankForce** que simularemos para observar como se llena el tanque.

## 8. Sistema Completo

Construiremos por último el sistema completo de la Figura 1 conectando la inercia del sistema **DCMotor** de la Sección 5.2 con la manivela del sistema **RodCrankMass** de la Sección 6.2 y conectando la masa de este último sistema con el pistón del sistema **PistonPumpTank** de la Sección 7.2.

El objetivo es hacer simulaciones para como se llena el tanque según vamos modificando el voltaje del motor y otros parámetros que considere de interés.