

TALLER 1. VARIABLES DE ESTADO CIRCUITOS ELECTRICOS

OBJETIVOS

1. Utilizar datos, indicios e información para formular las ecuaciones de un sistema (CDIO 2.1.1)
2. Obtener modelos conceptuales y cualitativos de un circuito eléctrico (CDIO 2.1.2)
3. Computar las soluciones del problema empleando varios métodos y herramientas (CDIO 2.1.5)

CONTENIDO:

1. Plantear modelos en espacio de estado sistema continuo
2. Obtener soluciones empleando MATLAB

1. Planteo y solución ecuación entrada – salida de circuitos eléctricos.

Para el circuito eléctrico de la figura 1

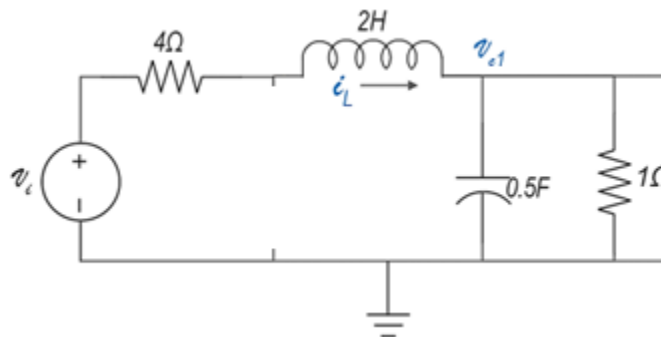


Figura 1.

- a. Plantear la ecuación diferencial de entrada – salida. La variable de salida deseada es el voltaje sobre el condensador.

Para la solución de la ecuación se puede emplear el comando dsolve:

dsolve

Solve system of differential equations

Syntax

```
S = dsolve(eqn)
S = dsolve(eqn,cond)
S = dsolve(__,Name,Value)
```

```
[y1,...,yN] = dsolve(__)
```

Description

`S = dsolve(eqn)` solves the differential equation `eqn`, where `eqn` is a symbolic equation. Use `diff` and `==` to represent differential equations. For example, `diff(y,x) == y` represents the equation $dy/dx = y$. Solve a system of differential equations by specifying `eqn` as a vector of those equations.

`S = dsolve(eqn,cond)` solves `eqn` with the initial or boundary condition `cond`.

`S = dsolve(__,Name,Value)` uses additional options specified by one or more `Name,Value` pair arguments.

`[y1,...,yN] = dsolve(__)` assigns the solutions to the variables `y1, ..., yN`.

Para graficar las variables se emplea el comando `fplot`

fplot

Syntax

```
fplot(f)
fplot(f,xinterval)
fplot(funx,funy)
fplot(funx,funy,tinterval)
fplot(__,LineStyle)
fplot(__,Name,Value)
fplot(ax,__)
fp = fplot(__)
[x,y] = fplot(__)
```

Description

`fplot(f)` plots the curve defined by the function $y = f(x)$ over the default interval $[-5 \ 5]$ for x .

`fplot(f,xinterval)` plots over the specified interval. Specify the interval as a two-element vector of the form $[xmin \ xmax]$.

`fplot(funx,funy)` plots the curve defined by $x = funx(t)$ and $y = funy(t)$ over the default interval $[-5 \ 5]$ for t .

`fplot(funx,funy,tinterval)` plots over the specified interval. Specify the interval as a two-element vector of the form $[tmin \ tmax]$.

`fplot(__,LineStyle)` specifies the line style, marker symbol, and line color. For example, `'-r'` plots a red line. Use this option after any of the input argument combinations in the previous syntaxes.

`fplot(__,Name,Value)` specifies line properties using one or more name-value pair arguments. For example, `'LineWidth',2` specifies a line width of 2 points.

[example](#)

[example](#)

[example](#)

[example](#)

`fplot(ax, __)` plots into the axes specified by `ax` instead of the current axes (`gca`). Specify the axes as the first input argument.

example

`fp = fplot(__)` returns a `FunctionLine` object or a `ParameterizedFunctionLine` object, depending on the inputs. Use `fp` to query and modify properties of a specific line. For a list of properties, see [FunctionLine Properties](#) or [ParameterizedFunctionLine Properties](#).

`[x,y] = fplot(__)` returns the abscissas and ordinates for the function without creating a plot. This syntax will be removed in a future release. Use the `XData` and `YData` properties of the line object, `fp`, instead.

- Si las condiciones iniciales son $i_l(0) = 0,2A$ y $v_{c1}(0) = 0,2V$, obtener la respuesta (para entrada cero) analítica y gráfica.
- Si la entrada es una señal paso de la forma $v_i = 10 \cdot 1(t)$, obtener la respuesta (para estado cero) analítica y grafica.
- Obtener la respuesta completa, analítica y grafica.

2. Planteo y solución de modelo de estado de circuitos eléctricos.

- Para el mismo circuito de la Figura 1 plantear la ecuación dinámica de estado y la ecuación de salida en forma matricial. Identificar las dimensiones de las matrices A , B , C y D . La variable de salida de interés es el voltaje sobre el condensador.

Funciones de MATLAB

Con el software Control System Toolbox™ de Matlab, se pueden representar sistemas dinámicos como objetos de modelo. Los objetos de modelo son contenedores de datos especializados que encapsulan datos de los modelos y otros atributos de manera estructurada. Los objetos de modelo pueden representar sistemas de entrada y salida únicas (SISO) o sistemas de entrada y salida múltiples (MIMO). Puede representar sistemas lineales de tiempo continuo y de tiempo discreto y sistemas con retardos de tiempo. Los objetos de modelo básicos, como las funciones de transferencia y los modelos de espacio de estados, representan sistemas con coeficientes numéricos fijos.

Modelo en Variables de Estado	Función de Transferencia
$\dot{X}(t) = AX(t) + BU(t)$	$H(s) = \frac{Y(s)}{U(s)} = \frac{N(s)}{D(s)}$
$Y(t) = CX(t) + DU(t)$	$H(s) = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_n}$
$X(0) = X_0$	

En Matlab existe un conjunto de funciones para resolver las ecuaciones de estado de sistemas lineales e invariantes con el tiempo.

Función	Descripción	
<u>expm</u>	Calcula la exponencial de la matriz X	$y = \text{expm}(X)$
<u>eig</u>	Calcula los valores y vectores propios de una matriz X	$[Vec, Val] = \text{eig}(X)$
<u>ss</u>	Construye un modelo en variables de estado	$\text{sys} = \text{ss}(A, B, C, D)$
<u>step</u>	Calcula la respuesta a estado cero frente a una entrada paso de un sistema sys que puede ser una función de transferencia o modelo en variables de estado. T indica el valor final en tiempo y no es un parámetro obligatorio.	$[y, t] = \text{step}(\text{sys}, T)$
<u>initial</u>	Calcula la respuesta a entrada cero de un sistema sys (modelo en variables de estado) con condiciones iniciales X_0 . T indica el valor final en tiempo y no es un parámetro obligatorio.	$[y, t] = \text{initial}(\text{sys}, X_0, T)$
<u>lsim</u>	Calcula la respuesta completa para una entrada arbitraria a un sistema sys (puede ser una función de transferencia o modelo en variables de estado). Para el caso de analizar el sistema bajo condiciones iniciales X_0 , el sistema debe modelarse como variables de estado.	$y = \text{lsim}(\text{sys}, u, t, X_0)$

Usando los comandos de MATLAB:

- Si las condiciones iniciales son $i_l(0) = 0,2A$ y $v_{c1}(0) = 0,2V$, obtener la respuesta (para entrada cero) analítica y gráfica.
- Si la entrada es una señal paso de la forma $v_i = 10 \cdot 1(t)$, obtener la respuesta (para estado cero) analítica y gráfica.
- Obtener la respuesta completa, analítica y gráfica.

3. Obtención de la función de transferencia

A partir de la representación de estado se puede obtener la función de transferencia del sistema. Esta es la representación entrada – salida en el dominio de la frecuencia o plano s

Se pueden emplear el comando:

tf

Transfer function model

Description

Use `tf` to create real-valued or complex-valued transfer function models, or to convert [dynamic system models](#) to transfer function form.

Transfer functions are a frequency-domain representation of linear time-invariant systems. For instance, consider a continuous-time SISO dynamic system represented by the transfer function $\text{sys}(s) = N(s)/D(s)$, where $s = j\omega$ and $N(s)$ and $D(s)$ are called the numerator and denominator polynomials, respectively. The `tf` model object can represent SISO or MIMO transfer functions in continuous time or discrete time.

You can create a transfer function model object either by specifying its coefficients directly, or by converting a model of another type (such as a state-space model `ss`) to transfer-function form. For more information, see [Transfer Functions](#).

You can also use `tf` to create generalized state-space ([genss](#)) models or uncertain state-space ([uss](#) (Robust Control Toolbox)) models.

Creation

Syntax

```
sys = tf(numerator,denominator)
sys = tf(numerator,denominator,ts)
sys = tf(numerator,denominator,ltiSys)
sys = tf(m)
sys = tf(__,Name,Value)
sys = tf(ltiSys)
sys = tf(ltiSys,component)
s = tf('s')
z = tf('z',ts)
```

El diagrama de polos y ceros se crea con el comando:

zpk

Zero-pole-gain model

- Obtener la función de transferencia del circuito de la figura 1 y graficar el diagrama de polos y ceros.
- Si la entrada es una señal paso de la forma $v_i = 10 \cdot 1(t)$, obtener la respuesta (para estado cero) grafica.
- Repetir para una entrada $v_i = 2 \sin t$

NOTAS

1. <https://www.mathworks.com/help/symbolic/dsolve.html>
2. <https://www.mathworks.com/help/matlab/ref/fplot.html>
3. <https://www.mathworks.com/help/control/ref/tf.html>
4. <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=Syst emModeling>
5. <https://www.mathworks.com/help/control/linear-system-modeling.html>
6. Derek Rowell. State-Space Representation of LTI Systems
7. Experimentos: grupos de dos personas máximo.
8. COPIA DETECTADA DURANTE LA CALIFICACIÓN SE SANCIONARÁ CON LA ANULACIÓN DEL EXPERIMENTO Y LA CORRESPONDIENTE SANCIÓN ESTABLECIDA EN EL REGLAMENTO.
9. Revisión con el monitor: semana 2 – 30 enero a febrero 3.
10. Entrega reporte: semana 3. 10 febrero antes de las 23.59 por la plataforma BS. Después de esta hora no se recibe. (ver guía reportes ver 2)

REVISIONES

Revisión 1	Febrero 2013	CCB
Revisión 2	Agosto 2014	CCB
Revisión 3	Febrero 2015	CCB
Revisión 4	Febrero 2016	CCB
Revisión 5	Julio 2016	CCB
Revisión 6	Enero 2017	CCB
Revisión 7	Julio 2017	CCB
Revisión 8	Julio 2022	CCB/CCS