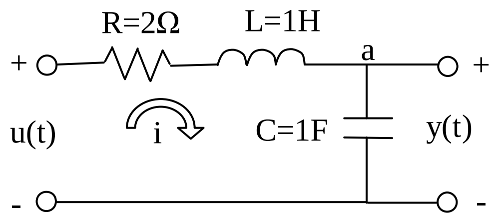
Use State Variable to represent circuits

Let's start with a simple example:



NOTE: in this chapter, we will use u(t) (not to be confused with the step function) to denote the input and use x(t) to denote the state.

1. Define the states:
   Similar to the discrete-case, we define the states based on the memory storage elements. For passive circuit, the memory storage elements are the capacitors and inductors:
   $$i_C = C\frac{dv_C}{dt} \quad and \quad v_L = L\frac{di_L}{dt}$$
   As the knowledge capacitor voltage and inductor current allows us to infer the capacitor current and the inductor voltage by taking derivatives, we define state variables to be the capacitor voltage and inductor current.
   $$x_1 = v_c(t)$$
   $$x_2 = i_L(t)$$
   Note that there is one state variable for each memory storage element.

2. Derive the "State" and "Output" equations – express the derivatives of the states and the output in terms of the current state and the input ONLY.

   KCL at node a: $\frac{dx_1}{dt} = \frac{1}{C}x_2$

   KVL: $\frac{dx_2}{dt} = -\frac{1}{L}x_1 - \frac{R}{L}x_2 + \frac{1}{L}u$

   These two equations can be more compactly written in matrix form:
   $$\begin{pmatrix}\dot{x}_1 \\ \dot{x}_2\end{pmatrix} = \begin{pmatrix}0 & 1/C \\ -1/L & -R/L\end{pmatrix}\begin{pmatrix}x_1 \\ x_2\end{pmatrix} + \begin{pmatrix}0 \\ 1/L\end{pmatrix}u$$
   The above matrix equation is called the State Equation because it relates the change (1st-order derivative) of the state to the current state and input.

   We can also relate the output to the state variables as follows:

   Output Equation: $y(t) = \begin{pmatrix}1 & 0\end{pmatrix}\begin{pmatrix}x_1 \\ x_2\end{pmatrix}$

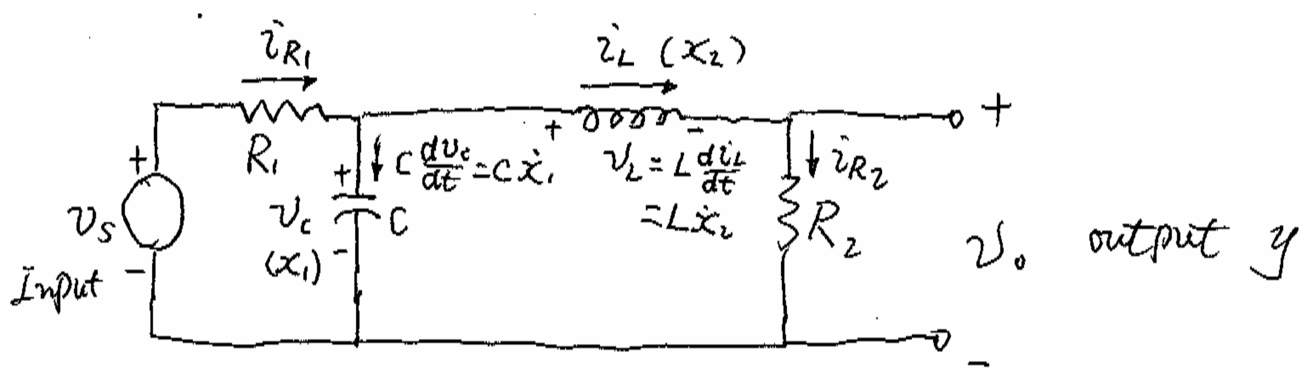
Using this approach, we can write the state-variable representation for any circuit. In general, the state and output equations are always in the following form:

$$\boxed{\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}\end{aligned}}$$

In our previous example, we have

$A = \begin{pmatrix} 0 & 1/C \\ -1/L & -R/L \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ 1/L \end{pmatrix}$, $C = \begin{pmatrix} 1 & 0 \end{pmatrix}$, $D = 0$

Be very careful about the dimensions of each matrix. Let's do another example:



Step 1: Define the state variable: v_c as x_1, i_L as x_2

Step 2: Express the derivatives $\dot{x}_1$, $\dot{x}_2$ and output y in terms of x_1, x_2 and input v_s.

$$y = R_2 i_{R_2} = R_2 i_L = R_2 x_2 = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$ Output Equation

$$\dot{x}_1 = \frac{1}{C} i_C = \frac{1}{C}(i_{R_1} - i_L) = \frac{1}{C}\left(\frac{v_s - x_1}{R_1} - x_2\right) = \frac{-1}{R_1 C} x_1 - \frac{1}{C} x_2 + \frac{1}{R_1 C} v_s \qquad (1)$$

$$\dot{x}_2 = \frac{1}{L} v_L = \frac{1}{L}(v_c - y) = \frac{1}{L}(x_1 - R_2 x_2) = \frac{1}{L} x_1 - \frac{R_2}{L} x_2 \qquad (2)$$

Combining (1) and (2):

$$\Rightarrow \begin{bmatrix} \dot{x}_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{CR_1} & -\dfrac{1}{C} \\ \dfrac{1}{L} & -\dfrac{R_2}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \dfrac{1}{CR_1} \\ 0 \end{bmatrix} v_s$$

State Equation

To summarize:

For electrical network: select i_L and v_C as state variables.

Step 1: Select each i_L and v_c as state variables

Step 2: For each i_L, write a KVL ($\dot{i}_L = \frac{di_L}{dt}$ will be included)

For each v_c, write a KCL ($\dot{v}_C$ will be included)

Step 3: Other KCL and KVL, and element relation to eliminate "other" variables (other than states (i_L, v_c) and sources (input).

=>state equation.

Step 4 : Output equation

State Equation from Transfer Functions

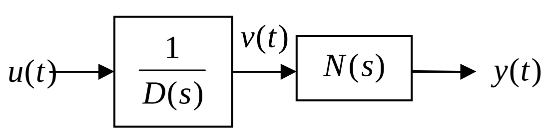
State Equation => Tell how to realize and simulate (system realization) the systems

Given $H(s) = \frac{N(s)}{D(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0}$ (Assume m<n $\Rightarrow$ BIBO)

Find
$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$
(u --- scalar, y--- scalar)

Such that $C(sI - A)^{-1} B + D$ (the transfer function expression derived directly from the state equations) equals the given transfer function.

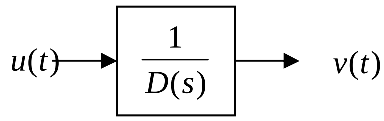
Given a system $H(s) = \frac{N(s)}{D(s)}$, we can implement this with the following block diagram:



Assume the denominator polynomial is an n[th]-degree polynomial, define our state vector as a n-dimensional vector follows:

$$\begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ \frac{d}{dt} v(t) \\ \vdots \\ \frac{d^{n-2}}{dt^{n-2}} v(t) \\ \frac{d^{n-1}}{dt^{n-1}} v(t) \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ ? & ? & ? & ? & ? & ? \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{pmatrix}$$

Note that our definition already provide almost the complete form of a state equation, except that we don't have an expression for $\dot{x}_n(t)$.
That has to come from the system. In the first part of our system:



Or:

$$\begin{aligned} &U(s) = D(s)V(s) \\ &= s^n V(s) + a_{n-1}s^{n-1}V(s) + \ldots + a_1 sV(s) + a_0 V(s) \end{aligned}$$

Applying inverse Laplace Transform:

$$\begin{aligned} &u(t) = \frac{d^n}{dt^n}v(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}v(t) + \ldots + a_1\frac{d}{dt}v(t) + a_0 v(t) \\ &= \dot{x}_n(t) + a_{n-1}x_n(t) + \ldots + a_1 x_2(t) + a_0 x_1(t) \\ &\dot{x}_n(t) = u(t) - a_{n-1}x_n(t) - \ldots - a_1 x_2(t) - a_0 x_1(t) \end{aligned}$$

Now plug it back to our system:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ -a_0 & -a_1 & -a_2 & -a_3 & \ldots & -a_{n-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} u(t)$$

Let $N(s) = b\ s^m + b_1 s^{m-1} + \ldots + b_{m-1}s + b_m$. The output equation can be written as

$$\begin{aligned} &Y(s) = N(s)V(s) \\ &= b_m s^m V(s) + b_{m-1}s^{m-1}V(s) + \ldots + b_1 sV(s) + b_0 V(s) \end{aligned}$$

In time domain:

$$\begin{aligned} &y(t) = b_m\frac{d^m}{dt^m}v(t) + b_{m-1}\frac{d^{m-1}}{dt^{m-1}}v(t) + \ldots + b_1\frac{d}{dt}v(t) + b_0 v(t) \\ &= b_m x_{m+1}(t) + b_{m-1}x_m(t) + \ldots + b_1 x_2(t) + b_0 x_1(t) \\ &= \begin{pmatrix} b_0 & b_1 & \cdots & b_m \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m+1} \end{pmatrix} \end{aligned}$$

Example/ $H(s)=\dfrac{4s^2+3s-1}{6s^3+7s^2+s+5}$

First, we normalize H(s) so that the leading coefficient of the denominator polynomial is 1: $H(s)=\dfrac{\frac{2}{3}s^2+\frac{1}{2}s-\frac{1}{6}}{s^3+\frac{7}{6}s^2+\frac{1}{6}s+\frac{5}{6}}$

By inspection, we can just write out the state equations:

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{pmatrix} = \begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -\frac{5}{6} & -\frac{1}{6} & -\frac{7}{6}\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}+\begin{pmatrix}0\\ 0\\ 1\end{pmatrix}u$$

$$y = \begin{pmatrix}-\frac{1}{6} & \frac{1}{2} & \frac{2}{3}\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}$$

<u>Where do we go from here?</u>

After obtaining the state variable representation, it will allow us to

1. Analyze its stability, controllability and observability
   a. The “eigenvalues” of our state matrix A are precisely the poles of the linear system.
   b. A system is called controllable if, regardless of its initial state, we can drive its state to any value within finite time by using a suitable input. A system is controllable if and only if the following matrix is invertible:
   $$\left[B \mid AB \mid AB^2 \mid \cdots \mid AB^{n-1}\right]$$
   c. A system is called observable if we can determine the initial state given any output. A system is observable if and only if the following matrix is invertible:
   $$\begin{bmatrix}C\\ CA\\ CA^2\\ \vdots\\ CA^{n-1}\end{bmatrix}$$

2. Simulate it with a computer by discretization

To simulate $\begin{matrix}\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} = \mathbf{Cx} + \mathbf{Du}\end{matrix}$ we can follow the procedure below:

1. At $t = 0$, $\mathbf{x}(t) = \mathbf{0}$ (or can be any initial state)
2. Compute dynamics: $\dot{\mathbf{x}}(t) = \mathbf{Ax}(t) + \mathbf{Bu}(t)$
3. Compute output: $\mathbf{y}(t) = \mathbf{Cx}(t) + \mathbf{Du}(t)$
4. Compute new state by Taylor Series Expansion: $\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t$
5. $t = t + \Delta t$
6. Go to step 2.

This implementation assumes the input is constant and the state vector is linear within the sample interval $[t, t + \Delta t)$. While the first assumption is not too unreasonable, we can improve upon the second one by combining step 2 and 4 together to compute

$$\mathbf{x}(t + \Delta t) = e^{\mathbf{A}\Delta t}\mathbf{x}(t) + (e^{\mathbf{A}\Delta t} - \mathbf{I})\mathbf{A}^{-1}\mathbf{B}u(t)$$

This involves a matrix exponential which can be computed by using Taylor Series:

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2 t^2 + \frac{1}{3!}\mathbf{A}^3 t^3 + ...$$

But I am afraid we are running out of time. For further reference on the use of State-Variable Representation, please consult the excellent text “Linear System Theory and Design” by C.T. Chen.