Control theory: state-space approach for linear systems

Introduction to state-space representation

2 janvier 2022

State-space representation

In this course, systems (physical, biological, chemical,... processes) are described by equations having the following form :

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t)(+Du(t)) \end{array} \right. \quad (1)$$

with the assumption that time t is continuous (that is taking values in $\mathbb{R}$).

State-space representation

In this course, systems (physical, biological, chemical,... processes) are described by equations having the following form :

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases} \qquad (1)$$

with the assumption that time t is continuous (that is taking values in $\mathbb{R}$).

- Vector $u(t)$ is the *input*. Its value can be fixed arbitrarily for all t.

State-space representation

In this course, systems (physical, biological, chemical,... processes) are described by equations having the following form :

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases} \tag{1}$$

with the assumption that time t is continuous (that is taking values in $\mathbb{R}$).

- Vector $u(t)$ is the *input*. Its value can be fixed arbitrarily for all t.
- Vector $y(t)$ is the *output* which can be measured.

State-space representation

In this course, systems (physical, biological, chemical,... processes) are described by equations having the following form :

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases} \tag{1}$$

with the assumption that time t is continuous (that is taking values in $\mathbb{R}$).

- Vector $u(t)$ is the *input*. Its value can be fixed arbitrarily for all t.
- Vector $y(t)$ is the *output* which can be measured.
- Vector $x(t)$ is called the *state* of the system. It's a kind of "memory" for the system, since it gathers enough pieces of information needed to predict the future behavior of the system (knowing the input $u(t)$).

State-space representation

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases} \tag{2}$$

- The first equation in (2) is called *evolution equation* or *state equation*. This is a (set of) differential equation(s) which expresses the "trend" of state $x(t)$ knowing its value at time-instant t and the applied input $u(t)$.

State-space representation

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases} \quad (2)$$

- The first equation in (2) is called *evolution equation* or *state equation*. This is a (set of) differential equation(s) which expresses the "trend" of state $x(t)$ knowing its value at time-instant t and the applied input $u(t)$.

State-space representation

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases} \tag{2}$$

- The first equation in (2) is called *evolution equation* or *state equation*. This is a (set of) differential equation(s) which expresses the "trend" of state $x(t)$ knowing its value at time-instant t and the applied input $u(t)$.
- The second equation in (2) is called *output equation*. It makes it possible to compute the output vector $y(t)$ knowing the state value at t. Unlike the evolution equation, it is not a differential equation.

State-space representation

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \right. \tag{2}$$

- The first equation in (2) is called *evolution equation* or *state equation*. This is a (set of) differential equation(s) which expresses the "trend" of state $x(t)$ knowing its value at time-instant t and the applied input $u(t)$.
- The second equation in (2) is called *output equation*. It makes it possible to compute the output vector $y(t)$ knowing the state value at t. Unlike the evolution equation, it is not a differential equation.
- The set of equations (2) constitute the *state-space representation*.

If the system is studied by means of a computer, then the input and the output cannot be examined continuously, but only at discrete time-instants (synchronized with the processor clock). It is then necessary to consider that time takes its values k in $\mathbb{Z}$.

If the system is studied by means of a computer, then the input and the output cannot be examined continuously, but only at discrete time-instants (synchronized with the processor clock). It is then necessary to consider that time takes its values k in $\mathbb{Z}$. We then use a *discrete-time state-space representation* involving recurrence equations :

$$\left\{ \begin{array}{rcl} x(k+1) & = & Ax(k) + Bu(k) \\ y(k) & = & Cx(k) \end{array} \right. \tag{3}$$

Several general remarks on the state-space representation :

- systems are studied in the time-domain,

Several general remarks on the state-space representation :

- systems are studied in the time-domain,
- manipulations involve matrix algebra,

Several general remarks on the state-space representation :

- systems are studied in the time-domain,
- manipulations involve matrix algebra,
- the formalism makes it possible to naturally extend the results to multiple-input multiple-output (MIMO) systems.

State variables

First-order linear system described by a differential equation :

$$\dot{y}(t) - ay(t) = bu(t), y(0^+) = y_0.$$

State variables

First-order linear system described by a differential equation :

$$\dot{y}(t) - ay(t) = bu(t), y(0^+) = y_0.$$

The solution to this equation is

$$y(t) = y(0^+) \cdot e^{at} + \int_0^t b \cdot e^{a(t-x)} \cdot u(x)dx$$

It makes it visible that we need to know the whole past of the input (before t) to be able to evaluate the current output value (at t).

State variables

On the contrary, using state-space representation,

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases}$$

the output at t of the same linear system can be directly deduced from the knowledge of the state at t.

State variables

On the contrary, using state-space representation,

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases}$$

the output at t of the same linear system can be directly deduced from the knowledge of the state at t.
In a way, state vector $x(t)$ provide a complete knowledge of the past functioning of the system (of what happened in the past).

Example : container handling gantry crane

The choice of state-variables is the most tricky step to derive a state-space representation.

One possible reasoning is to select a set of variables describing the "stocks of energy" into the system as state variables.

This way, we can guess that the mass-speed (rendering the kinetic energy) and the angle θ (rendering the potential energy) are good state variables for this system.

We can then choose

$$x = \begin{pmatrix} v_m \\ \theta \end{pmatrix}$$

Example : container handling gantry crane

From the equations

$$\frac{d\theta}{dt} = -\frac{1}{L}v_m + \frac{1}{L}v_c, \qquad \frac{dv_m}{dt}(= a_m) = g\theta,$$

with

$$x = \begin{pmatrix} v_m \\ \theta \end{pmatrix}, \quad u = v_c, \quad y = v_m,$$

we deduce a state-space representation for the system :

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 0 & g \\ -\frac{1}{L} & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u(t) \\ y(t) & = & \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{array} \right. .$$

Remarks

As mentioned before, the variables describing the "stocks of energy" into the system are good candidates for state variables.

An alternative reasoning is "find a set of variables such that the knowledge of their values at t is sufficient to predict the future of the output".

The minimal number of variables needed for its state corresponds to the *order* of the system.

Equivalent state-space representations

Let us consider a continuous-time state-space representation for a system. Let P be a square invertible (non-singular) matrix and $\chi(t) = P^{-1}x(t)$, $A' = P^{-1}AP$, $B' = P^{-1}B$, $C' = CP$.

Equivalent state-space representations

Let us consider a continuous-time state-space representation for a system. Let P be a square invertible (non-singular) matrix and $\chi(t) = P^{-1}x(t)$, $A' = P^{-1}AP$, $B' = P^{-1}B$, $C' = CP$. We then have

$$\left\{ \begin{array}{rcl} \dot{\chi}(t) & = & A'\chi(t) + B'u(t) \\ y(t) & = & C'\chi(t) \end{array} \right. \tag{4}$$

If we replace χ, A', B' and C' by their expression, we get

$$\left\{ \begin{array}{rcl} P^{-1}\dot{x}(t) & = & P^{-1}APP^{-1}x(t) + P^{-1}Bu(t) \\ y(t) & = & CPP^{-1}x(t) \end{array} \right.$$

$$\Longleftrightarrow \left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \right.$$

Equivalent state-space representations

Let us consider a continuous-time state-space representation for a system. Let P be a square invertible (non-singular) matrix and $\chi(t) = P^{-1}x(t)$, $A' = P^{-1}AP$, $B' = P^{-1}B$, $C' = CP$. We then have

$$\left\{ \begin{array}{lcl} \dot{\chi}(t) & = & A'\chi(t) + B'u(t) \\ y(t) & = & C'\chi(t) \end{array} \right. \quad (4)$$

These last equations have an identical form to the original state-space representation. In other words, we can get that way a second state-space representation for the same system.

Equivalent state-space representations

- With a similar reasoning, it is possible to get equivalent state-representations for discrete-time systems.

Equivalent state-space representations

- With a similar reasoning, it is possible to get equivalent state-representations for discrete-time systems.
- A system admits as many equivalent state-representations as there exist different square invertible matrices P.

Equivalent state-space representations

- With a similar reasoning, it is possible to get equivalent state-representations for discrete-time systems.
- A system admits as many equivalent state-representations as there exist different square invertible matrices P.
- Vector $\chi(t)$ is another valid state vector for the system. Except if P is equal to the identity matrix, the state variables have then a different "physical meaning".