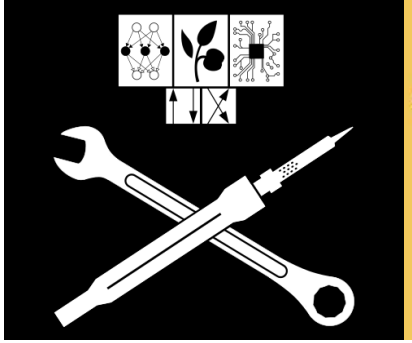
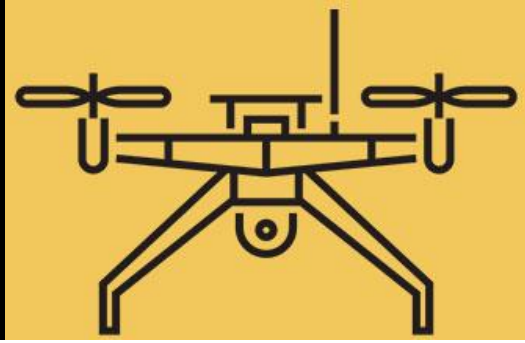




State-space representations

Lecture 2

Jerome Jouffroy, Professor
jerome@sdu.dk

SDU

Today's lecture

- State-space representations (Linear, Nonlinear)
- From continuous-time to discrete-time SS rep.
- From ODEs to SS rep (and back)
- Equilibrium points in a SS context
- Linear approximations of nonlinear systems

State vector and state space

Introduce the vector $\mathbf{x}(t)$ taking values in $\mathbb{R}^n$, ie $\mathbf{x}(t) \in \mathbb{R}^n$

State vector

State space

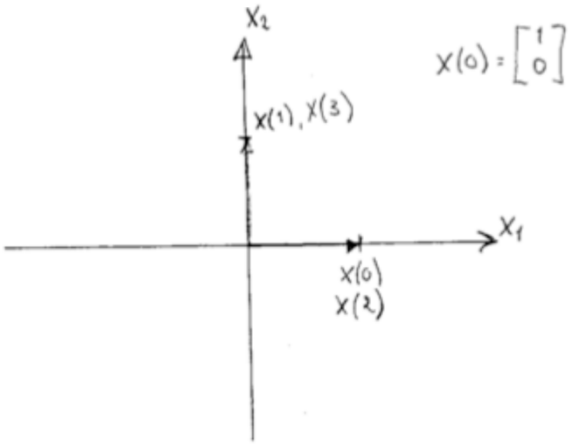
Important property: knowing the state vector at time t_0

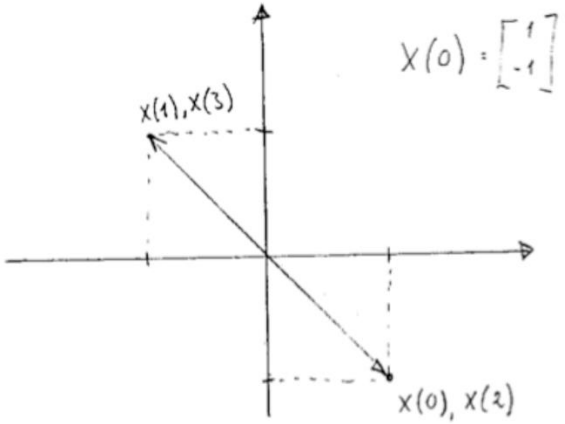
is sufficient to know the state vector for all $t \geq t_0$

$$\begin{cases} x_1(t+1) = x_2(t), & x_1(0) = 1 \\ x_2(t+1) = x_1(t), & x_2(0) = 0 \end{cases}$$

define

$$\mathbf{x}(t) := \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$





SDU

Linear state-space representations (1/3)

A linear time-invariant (LTI) state-space representation is described by

(in continuous-time)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0$$

with

Remark: a SS rep is a first-order form (in CT, an ODE of order 1)

$$\mathbf{x} \in \mathbb{R}^n, \quad \mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{u} \in \mathbb{R}^m, \quad \mathbf{B} \in \mathbb{R}^{n \times m}$$

(in discrete-time)

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \qquad \mathbf{x}(0) = \mathbf{x}_0$$

Linear state-space representations (2/3)

Pharmacokinetics example:

$$\begin{cases} V_1 \dot{c}_1(t) = q\left(c_2(t) - c_1(t)\right) - q_0 c_1(t) + c_0 u(t) \\ V_2 \dot{c}_2(t) = q\left(c_1(t) - c_2(t)\right) \end{cases}$$

(2 coupled linear first-order ODEs)

Rewrite as

$$\begin{cases} \dot{c}_1(t) = \dfrac{q}{V_1}\left(c_2(t) - c_1(t)\right) - \dfrac{q_0}{V_1} c_1(t) + \dfrac{c_0}{V_1} u(t) \\ \dot{c}_2(t) = \dfrac{q}{V_2}\left(c_1(t) - c_2(t)\right) \end{cases}$$

Define the state vector

$$\mathbf{x}(t) := \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix}$$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),$$

which gives

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \underbrace{\begin{bmatrix} -\dfrac{q}{V_1} - \dfrac{q_0}{V_1} & \dfrac{q}{V_1} \\ \dfrac{q}{V_2} & -\dfrac{q}{V_2} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \underbrace{\begin{bmatrix} \dfrac{c_0}{V_1} \\ 0 \end{bmatrix}}_{\mathbf{B}} u(t)$$

SDU

Linear state-space representations (3/3)

Add an output equation to represent outputs of interest:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), & \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases}$$

with

$$\mathbf{y} \in \mathbb{R}^p, \quad \mathbf{C} \in \mathbb{R}^{p \times n}, \quad \mathbf{u} \in \mathbb{R}^m, \quad \mathbf{D} \in \mathbb{R}^{p \times m}$$

Back to the pharmacokinetics example:

we choose to monitor $y(t) = c_2(t) = x_2(t)$

which gives

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t)$$

Nonlinear state-space representations (1/2)

In the nonlinear case, matrices are replaced by vector fields:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \qquad \mathbf{x}(0) = \mathbf{x}_0$$

(in continuous-time)

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t)), \qquad \mathbf{x}(0) = \mathbf{x}_0$$

(in discrete-time)

with $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$

Lorenz example:

$$\begin{cases} \dot{x}_1(t) = -px_1(t) + px_2(t) \\ \dot{x}_2(t) = -x_1(t)x_3(t) - x_2(t) \\ \dot{x}_3(t) = x_1(t)x_2(t) - x_3(t) \end{cases}$$

define

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

so that we have $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$

with $$\mathbf{f}(\mathbf{x}(t)) = \begin{bmatrix} -px_1(t) + px_2(t) \\ -x_1(t)x_3(t) - x_2(t) \\ x_1(t)x_2(t) - x_3(t) \end{bmatrix} = \begin{bmatrix} f_1(x_1(t), x_2(t), x_3(t)) \\ f_2(x_1(t), x_2(t), x_3(t)) \\ f_3(x_1(t), x_2(t), x_3(t)) \end{bmatrix}$$

Nonlinear state-space representations (2/2)

One can also add inputs to the dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

and the output equation

$$\mathbf{y}(t) = \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

Remark: x(t) can be really big and represent whole collections of dynamical systems (swarms)

From continuous to discrete SS rep: the Euler method

Question: how can one simulate a continuous-time system such as $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ on a computer?

We use discretization! → The Euler method: (many other methods)

$$\frac{\mathbf{x}(t+1) - \mathbf{x}(t)}{T_s} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$$

(DT approximation of a CT syst.)

$$\mathbf{x}(t+1) = \mathbf{x}(t) + T_s \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{f}'(\mathbf{x}(t), \mathbf{u}(t))$$

Lorenz example:

$$\begin{cases} \dot{x}_1(t) = -p x_1(t) + p x_2(t) \\ \dot{x}_2(t) = -x_1(t) x_3(t) - x_2(t) \\ \dot{x}_3(t) = x_1(t) x_2(t) - x_3(t) \end{cases}$$

gives

$$\begin{cases} \frac{x_1(t+1) - x_1(t)}{T_s} = -p x_1(t) + p x_2(t) \\ \frac{x_2(t+1) - x_2(t)}{T_s} = -x_1(t) x_3(t) - x_2(t) \\ \frac{x_3(t+1) - x_3(t)}{T_s} = x_1(t) x_2(t) - x_3(t) \end{cases}$$

so that we have

$$\begin{cases} x_1(t+1) = x_1(t) + T_s \left[-p x_1(t) + p x_2(t)\right] \\ x_2(t+1) = x_2(t) + T_s \left[-x_1(t) x_3(t) - x_2(t)\right] \\ x_3(t+1) = x_3(t) + T_s \left[x_1(t) x_2(t) - x_3(t)\right] \end{cases}$$

From n-th order ODEs to state-space representations (1/7)

Question(s): Many ODEs are of higher order than 1, while SS rep are of order 1. Link? Convert ODEs into SS rep?

Start with the MSD example:

$$m\ddot{y}(t) + d\dot{y}(t) + ky(t) = u(t)$$

which is equivalent to $$\ddot{y}(t) = -\frac{d}{m}\dot{y}(t) - \frac{k}{m}y(t) + \frac{1}{m}u(t)$$

Trick: define a state vector of the same dimension as the order the ODE you want to convert

MSD system is of order 2 ➡ $$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} := \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}$$ which gives $$\Rightarrow \dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \end{bmatrix}$$

this state definition gives $$\dot{x}_1(t) = x_2(t) \quad \text{and} \quad \dot{x}_2(t) = -\frac{d}{m}x_2(t) - \frac{k}{m}x_1(t) + \frac{1}{m}u(t)$$

so that we have the SS rep

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{bmatrix}}_{A} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B} u(t) \quad \text{and} \quad y(t) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

(D=0)

From n-th order ODEs to state-space representations (2/7)

General linear case (1/2)

Start with $y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \ldots + a_1\dot{y}(t) + a_0 y(t) = bu(t)$ **(linear ODE of order n)**

Define the state vector of dim n

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{bmatrix} := \begin{bmatrix} y(t) \\ \dot{y}(t) \\ \ddot{y}(t) \\ \vdots \\ y^{(n-1)}(t) \end{bmatrix}$$

gives

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \\ \vdots \\ y^{(n)}(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ x_3(t) \\ \vdots \\ y^{(n)}(t) = \dot{x}_n(t) \end{bmatrix}$$

such that $y^{(n)}(t) = -a_0 y(t) - a_1\dot{y}(t) - \ldots - a_{n-1}y^{(n-1)}(t) + b.u(t)$

is replaced with

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = x_3(t) \\ \vdots \\ \dot{x}_n(t) = -a_0 x_1(t) - a_1 x_2(t) - \ldots - a_{n-1}x_n(t) + b.u(t) \end{cases}$$

From n-th order ODEs to state-space representations (3/7)

General linear case (2/2)

In vectorial form, the previous system reads

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix}}_{\mathbf{B}} u(t)$$

Output equation:

start with the state definition

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{bmatrix} := \begin{bmatrix} y(t) \\ \dot{y}(t) \\ \ddot{y}(t) \\ \vdots \\ y^{(n-1)}(t) \end{bmatrix}$$

which gives $y(t) = x_1(t)$

So that we have $\mathbf{C} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}$, $\mathbf{D} = 0$

From n-th order ODEs to state-space representations (4/7)

The state is just a definition. What if we define the state differently?

start with the ODE $\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = bu(t)$ (1)

and define $x_1(t) := \frac{1}{b}y(t)$ and $x_2(t) := \frac{d}{dt}\left(\frac{1}{b}y(t)\right) = \frac{1}{b}\dot{y}(t)$

which gives $\dot{x}_1(t) = x_2(t)$ and $\dot{x}_2(t) = \frac{1}{b}\ddot{y}(t)$

Rewrite ODE (1) as $\frac{1}{b}\ddot{y}(t) = -\frac{a_1}{b}\dot{y}(t) - \frac{a_0}{b}y(t) + u(t)$

which gives $\dot{x}_2(t) = -a_1 x_2(t) - a_0 x_1(t) + u(t)$

→ we have the SS rep

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -a_0 & -a_1\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u(t)$$

$$y(t) = \begin{bmatrix}b & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}$$

(compare with

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -a_0 & -a_1\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ b\end{bmatrix}u(t)$$

$$\mathbf{x}(t) := \begin{bmatrix}y(t)\\ \dot{y}(t)\end{bmatrix} \rightarrow y(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}$$

)

From n-th order ODEs to state-space representations (5/7)

With derivatives of the input:

$$y^{(n)}(t) + a_{n-1}y^{(n-1)}(t) + \ldots + a_1\dot{y}(t) + a_0 y(t)$$
$$= b_{n-1}u^{(n-1)}(t) + b_{n-2}u^{(n-2)}(t) + \cdots + b_1\dot{u}(t) + b_0 u(t)$$

A possible state-space representation is given by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \ldots & -a_{n-1} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_{n-1} \end{bmatrix} \mathbf{x}(t)$$

(Remark: the definition of x(t) is a bit more involved than without derivatives of the input)

From n-th order ODEs to state-space representations (6/7)

Discrete-time systems

$$y(t+n) + a_{n-1}y(t+n-1) + \ldots + a_1 y(t+1) + a_0 y(t) = bu(t)$$

(linear difference equation of order n)

define the state as

$$\mathbf{x}(t) := \begin{bmatrix} y(t) \\ y(t+1) \\ y(t+2) \\ \vdots \\ y(t+(n-1)) \end{bmatrix}$$

gives

$$\mathbf{x}(t+1) = \begin{bmatrix} y(t+1) \\ y(t+2) \\ y(t+3) \\ \vdots \\ y(t+n) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ x_3(t) \\ x_4(t) \\ \vdots \\ -a_0 x_1(t) - a_1 x_2(t) \ldots - a_{n-1} x_n(t) \end{bmatrix}$$

so that we have

$$\mathbf{x}(t+1) = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \ldots & -a_{n-1} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ b \end{bmatrix} u(t)$$

$$y = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} \mathbf{x}(t)$$

SDU

From n-th order ODEs to state-space representations (7/7)

Nonlinear systems: Remark: not all nonlinear ODEs can be transformed into (nonlinear). Example:

$$\ddot{y}(t) = y(t)\dot{u}^2(t)$$

BUT: a lot of nonlinear ODEs can be transformed using the same tricks as for linear ODEs

Pendulum example:

$$ml^2\ddot{\theta}(t) + mgl\sin\theta(t) = u(t), \quad \text{with} \quad y(t) = \theta(t)$$

define

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} := \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}$$

and use $\ddot{\theta}(t) = -\frac{g}{l}\sin\theta(t) + \frac{1}{ml^2}u(t)$

so that we get

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)) \quad \text{with} \quad \mathbf{f}(\mathbf{x}(t), u(t)) = \begin{bmatrix} x_2(t) \\ -\frac{g}{l}\sin x_1(t) + \frac{1}{ml^2}u(t) \end{bmatrix}$$

Equilibrium points in state-space representations (1/5)

Question: What is an equilibrium point in the state-space context?

Definition: An equilibrium point or equilibrium, noted $\mathbf{x}^*$, is a point in the state-space, for which, if the state $\mathbf{x}(t)$ reaches $\mathbf{x}^*$, it will stay there forever (in a state of equilibrium). □

Consequence of "stay there forever": $\frac{d}{dt}\mathbf{x}^* = 0$ (in continuous-time)

For nonlinear systems described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$ whose solution is $\mathbf{x}(t)$

x(t) stays on x* means x(t)=x*, so that x* fulfils

$$0 = \mathbf{f}(\mathbf{x}^*)$$

ie x* is the solution of this nonlinear algebraic equation

Equilibrium points in state-space representations (2/5)

Pendulum example

Recall the pendulum SS rep $\dot{\mathbf{x}}(t) = \begin{bmatrix} x_2(t) \\ -\frac{g}{l}\sin x_1(t) \end{bmatrix}$

To find the equilibrium points, solve

$$\begin{bmatrix} x_2^* \\ -\frac{g}{l}\sin x_1^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

so that we have

$$\mathbf{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

(pendulum down)

$$\mathbf{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} \pi \\ 0 \end{bmatrix}$$

(pendulum up)

There can be several equilibrium points, possibly many

Equilibrium points in state-space representations (3/5)

Pendulum with a torque input

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} x_2(t) \\ -\frac{g}{l}\sin x_1(t) + \frac{1}{ml^2}u(t) \end{bmatrix}$$

solving

$$\begin{bmatrix} x_2^* \\ -\frac{g}{l}\sin x_1^* + \frac{1}{ml^2}u^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

gives $x_2^* = 0$ and $0 = -\frac{g}{l}\sin x_1^* + \frac{1}{ml^2}u^*$

so that we have the equilibrium points $\mathbf{x}^* = \begin{bmatrix} x_1^* \\ 0 \end{bmatrix}$ where x_1^* satisfies $u^* = mgl\sin x_1^*$

Physical interpretation: the position at equilibrium can be anything as long as the corresponding torque is applied to the pendulum/robot arm

Equilibrium points in state-space representations (4/5)

General case with inputs

Consider the nonlinear state-space representation

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

The equilibrium points and their associated control input vector $\mathbf{x}^*,\ \mathbf{u}^*$ can be computed by solving the algebraic equation

$$0 = \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*)$$

Equilibrium points in state-space representations (5/5)

Discrete-time systems

For systems represented by

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t))$$

The equilibrium points are the solutions of the nonlinear algebraic equation

$$\mathbf{x}^* = \mathbf{f}(\mathbf{x}^*)$$

For systems with inputs represented by

$$\mathbf{x}(t+1) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

The equilibrium points and their associated inputs are the solutions of the nonlinear algebraic equations

$$\mathbf{x}^* = \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*)$$

Linear approximations of nonlinear systems (1/5)

Main idea: Locally and around an equilibrium point, a nonlinear system can be approximated by a linear one → Very useful for linear control of nonlinear systems

Start with $$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}\left(\mathbf{x}(t)\right)$$

and define small variations around $\mathbf{x}^*$: $$\delta\mathbf{x}(t) := \mathbf{x}(t) - \mathbf{x}^*$$

so that we have $$\frac{d}{dt}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right) = \mathbf{f}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right)$$

→ Taylor series approximation

$$\mathbf{f}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right) \approx \mathbf{f}(\mathbf{x}^*) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})\right|_{\mathbf{x}=\mathbf{x}^*} .\delta\mathbf{x}(t)$$

(vector field)

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

→ (Jacobian)

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \ldots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \ldots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \ldots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \ldots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix}$$

Linear approximations of nonlinear systems (2/5)

$$\frac{d}{dt}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right) = \mathbf{f}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right)$$

(NL system)

$$\mathbf{f}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right) \approx \mathbf{f}(\mathbf{x}^*) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})\right|_{\mathbf{x}=\mathbf{x}^*} .\delta\mathbf{x}(t)$$

(Taylor series approx.)

$$\frac{d}{dt}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right) \approx \mathbf{f}(\mathbf{x}^*) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})\right|_{\mathbf{x}=\mathbf{x}^*} .\delta\mathbf{x}(t)$$

$\frac{d}{dt}\mathbf{x}^*$ =0, $\mathbf{f}(\mathbf{x}^*)$ =0

BUT:

$$\frac{d}{dt}\mathbf{x}^* = 0 \qquad \mathbf{f}(\mathbf{x}^*) = 0$$

$$\frac{d}{dt}\left(\delta\mathbf{x}(t)\right) = \left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})\right|_{\mathbf{x}=\mathbf{x}^*}\right]\delta\mathbf{x}(t)$$

so that we have the linear SS rep.

$$\delta\dot{\mathbf{x}}(t) = \mathbf{A}\delta\mathbf{x}$$

with

$$\mathbf{A} = \left[\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})\right|_{\mathbf{x}=\mathbf{x}^*}\right]$$

SDU

(linear approximation of a NL system around a specific equilibrium point)

Linear approximations of nonlinear systems (3/5)

Pendulum example

Start again with $\dot{\mathbf{x}} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 \end{bmatrix}$

vector field → $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 \end{bmatrix}$

Jacobian → $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos x_1 & 0 \end{bmatrix}$

Linear approximation around. $\mathbf{x}^* = [0,0]^T$?

→ Evaluate the Jacobian around eq. point: $\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})\right|_{\mathbf{x}=\mathbf{x}^*=[0,0]^T} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix}$

$$\delta\dot{\mathbf{x}} = \mathbf{A}\delta\mathbf{x} \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix}$$

Linear approximations of nonlinear systems (4/5)

With inputs

Consider now NL system with input

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

which we want to approximate around $\mathbf{x}^*$ and $\mathbf{u}^*$

$$\delta\mathbf{x}(t) := \mathbf{x}(t) - \mathbf{x}^*$$
$$\delta\mathbf{u}(t) := \mathbf{u}(t) - \mathbf{u}^*$$

NL system $\frac{d}{dt}\left(\mathbf{x}^* + \delta\mathbf{x}(t)\right) = \mathbf{f}\left(\mathbf{x}^* + \delta\mathbf{x}(t), \mathbf{u}^* + \delta\mathbf{u}(t)\right)$

leads to (Taylor series)

$$\frac{d}{dt}\delta\mathbf{x}(t) = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u})\Big|_{\mathbf{x}=\mathbf{x}^*, \mathbf{u}=\mathbf{u}^*}\right]\delta\mathbf{x}(t) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}, \mathbf{u})\Big|_{\mathbf{x}=\mathbf{x}^*, \mathbf{u}=\mathbf{u}^*}\right]\delta\mathbf{u}(t)$$

so that we have the linear approx. $\delta\dot{\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{u}$

with $\mathbf{A} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u})\Big|_{\mathbf{x}=\mathbf{x}^*, \mathbf{u}=\mathbf{u}^*}\right]$ and $\mathbf{B} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}, \mathbf{u})\Big|_{\mathbf{x}=\mathbf{x}^*, \mathbf{u}=\mathbf{u}^*}\right]$

Linear approximations of nonlinear systems (5/5)

Adding a torque input to the pendulum

we would like to approximate $\dot{\mathbf{x}}(t) = \begin{bmatrix} x_2(t) \\ -\frac{g}{l}\sin x_1(t) + \frac{1}{ml^2}u(t) \end{bmatrix}$ around $\mathbf{x}^* = \begin{bmatrix} \pi/4 \\ 0 \end{bmatrix}$

compute the Jacobians

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, u) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos x_1 & 0 \end{bmatrix} \quad \text{and} \quad \frac{\partial \mathbf{f}}{\partial u}(\mathbf{x}, u) = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}$$

evaluation around x*,u*

$$\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})\right|_{\mathbf{x}=\mathbf{x}^*, u=u^*} = \begin{bmatrix} 0 & 1 \\ -\frac{\sqrt{2}}{2}\frac{g}{l} & 0 \end{bmatrix}$$

$$\delta\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{\sqrt{2}}{2}\frac{g}{l} & 0 \end{bmatrix}}_{A} \delta\mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}}_{B} \delta u$$

SDU