Aalto University

Dynamical models and numerical simulations

CHEM-E7225 (was E7195), 2024

Francesco Corona (¬_¬)

Chemical and Metallurgical Engineering
School of Chemical Engineering

Dynamical models

Dynamical models and numerical simulations

Dynamical models

We focus on deterministic differential equation models of dynamical systems, in time

- All numerical simulation methods executed on a computer discretise time

We highlight some relevant properties of continuos-time systems

- How to convert them to discrete-time systems

Continuous-time systems are often described by ordinary differential equations (ODE)

- ⇝ Other common forms of ODEs (delayed ODE)
- ⇝ Differential-algebraic equations (DAE)
- ⇝ Partial differential equations (PDE)

Continuous-time models (cont.)

We describe controlled dynamical systems in continuous-time with a first-order ODE

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

Nonlinear time-varying systems

$$u(t) \longrightarrow \boxed{\begin{aligned}\dot{x}(t) &= f\left(t, x(t), u(t)|\theta_x\right) \\ y(t) &= g\left(t, x(t), u(t)|\theta_y\right)\end{aligned}} \longrightarrow y(t)$$

- ⇝ $x(t) \in \mathcal{R}^{N_x}$
- ⇝ $u(t) \in \mathcal{R}^{N_u}$
- ⇝ $\theta_x \in \mathcal{R}^{N_{\theta_x}}$
- ⇝ $t \in \mathcal{R}$
- ⇝ $y(t) \in \mathcal{R}^{N_y}$
- ⇝ $\theta_y \in \mathcal{R}^{N_{\theta_y}}$

Function f is a general map from time t, state $x(t)$, controls $u(t)$ and parameters θ_x

- $f : [0, T] \times \mathcal{R}^{N_x} \times \mathcal{R}^{N_u} \mapsto \mathcal{R}^{N_x}$, to the rate of change of the state
- Because t is an explicit argument, function f is time-varying

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{N_x}(t) \end{bmatrix} = \begin{bmatrix} f_1\left(x_1(t), x_2(t), \ldots, x_{N_x}(t), u_1(t), u_2(t), \ldots, u_{N_u}(t), t|\theta_x\right) \\ f_2\left(x_1(t), x_2(t), \ldots, x_{N_x}(t), u_1(t), u_2(t), \ldots, u_{N_u}(t), t|\theta_x\right) \\ \vdots \\ f_{N_x}\left(x_1(t), x_2(t), \ldots, x_{N_x}(t), u_1(t), u_2(t), \ldots, u_{N_u}(t), t|\theta_x\right) \end{bmatrix}$$

Continuous-time models (cont.)

$$\underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{N_x}(t) \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} f_1\left(x_1(t), x_2(t), \dots, x_{N_x}(t), u_1(t), u_2(t), \dots, u_{N_u}(t), t | \theta_x\right) \\ f_2\left(x_1(t), x_2(t), \dots, x_{N_x}(t), u_1(t), u_2(t), \dots, u_{N_u}(t), t | \theta_x\right) \\ \vdots \\ f_{N_x}\left(x_1(t), x_2(t), \dots, x_{N_x}(t), u_1(t), u_2(t), \dots, u_{N_u}(t), t | \theta_x\right) \end{bmatrix}}_{f(x(t), u(t), t | \theta_x)}$$

We are interested in the conditions under which the differential equation has a solution

- Given a fixed initial value $x(0)$ for the state, and controls $u(t)$ with $t \in [0, T]$

The dependence of f on the the controls $u(t)$ is equivalent to another time-dependence

$$\begin{aligned} \dot{x}(t) &= f\left(x(t), u(t), t | \theta_x\right) \\ &:= \overline{f}\left(x(t), t | \overline{\theta}_x\right) \end{aligned}$$

A time-varying uncontrolled (autonomous, or time-homogeneous) differential equation

Continuous-time models (cont.)

$$\dot{x}(t) = f\left(x(t), t|\overline{\theta}_x\right)$$

An **initial value problem** (**IVP**) consists of a differential equation (ODE) and a restriction

- At $t = 0$, we constrain $x(t)$ to be some fixed value $x(0) = x_0$

A solution to the initial value problem on the open interval $[0, t)$ that contains the origin $t = 0$ is the differentiable function $x(\cdot)$ with $x(0) = x_0$ and $\dot{x}(t) = \overline{f}\left(x(t), t|\overline{\theta}_x\right)$

The solution to the IVP is equivalent to the solution to an integral equation,

$$x(t) = x_0 + \int_0^t f\left(x(\tau), \tau|\overline{\theta}_x\right) d\tau$$

Continuous-time models (cont.)

For notational simplicity, we leave away the dependence of function f on controls $u(t)$

- We can keep them fixed in time, together with the other parameters θ_x
- (The initial condition, $x(t=0)=x_0$, is also fixed)

Then, we have the uncontrolled dynamical system

$$\begin{aligned}\dot{x}(t) &= f\left(t, x(t) | \theta_x\right), \quad t \in [0, T] \\ x(0) &= x_0\end{aligned}$$

The solution,

$$x(t) = x_0 + \int_0^t f\left(x(\tau), \tau | \theta_x\right) d\tau$$

Existence and uniqueness of the solution to the IVP are implied by the properties of f

- Existence is guaranteed by the continuity of f with respect to $x(t)$ and t
- For continuous-time systems, existence is not a granted property

Continuous-time models (cont.)

Theorem

Existence and uniqueness

Let $f : [t_{\text{ini}}, t_{\text{fin}}] \times \mathcal{R}^{N_x} \rightarrow \mathcal{R}^{N_x}$ be some continuous function in $x(t)$ and t

Consider the initial value problem with initial value

$$\begin{aligned} \dot{x}(t) &= f\left(t, x(t)|\theta_x\right), \quad t \in [t_{\text{ini}}, t_{\text{fin}}] \\ x(t_{\text{ini}}) &= x_0 \end{aligned}$$

The IVP has a solution $x : [t_{\text{ini}}, t_{\text{fin}}] \rightarrow \mathcal{R}^{N_x}$ and that solution is the unique solution to the IVP problem if and only if function f is Lipschitz continuous with respect to $x(t)$

That is, there exists a constant value $L \in (0, \infty)$ such that for any pair $(x(t), x'(t))$,

$$\|f\left(x(t), t|\theta_x\right) - f\left(x'(t), t|\theta_x\right)\| \leq L\|x(t) - x'(t)\|, \quad \forall t \in [t_{\text{ini}}, t_{\text{fin}}]$$

Or, equivalently, for any pair $(x(t), x'(t))$

$$\frac{\|f\left(x(t), t|\theta_x\right) - f\left(x'(t), t|\theta_x\right)\|}{\|x(t) - x'(t)\|} \leq L, \quad \forall t \in [t_{\text{ini}}, t_{\text{fin}}]$$

Continuous-time models (cont.)

$$\frac{\| f\left(x(t), t|\theta_x\right) - f\left(x'(t), t|\theta_x\right)\|}{\|x(t) - x'(t)\|} \leq L, \quad \forall t \in [t_{\text{ini}}, t_{\text{fin}}]$$

Lipschitz continuity of f with respect to $x(t)$ is a property that is difficult to determine

- It is difficult to determine a global (over the time-interval) Lipschitz constant L

A simpler property to be verified is the differentiability of f with respect to $x(t)$

Because every function f which is differentiable with respect to $x(t)$ is locally Lipschitz continuous, we define the condition for local existence and uniqueness of the solution

Continuous-time models (cont.)

Theorem

Local existence and uniqueness

Let $f : [t_{\text{ini}}, t_{\text{fin}}] \times \mathcal{R}^{N_x} \rightarrow \mathcal{R}^{N_x}$ be some continuous function in $x(t)$ and t

Consider the initial value problem with initial value

$$\begin{aligned} \dot{x}(t) &= f\left(t, x(t) | \theta_x\right), \quad t \in [t_{\text{ini}}, t_{\text{fin}}] \\ x(t_{\text{ini}}) &= x_0 \end{aligned}$$

If f is continuously differentiable with respect to $x(t)$ for all $t \in [t_{\text{ini}}, t'_{\text{fin}}]$, there exists a non-empty interval $[t_{\text{ini}}, t'_{\text{fin}}]$ with $t'_{\text{fin}} \in (t_{\text{ini}}, t_{\text{fin}}]$ where the IVP has a unique solution

Continuous-time models (cont.)

Example

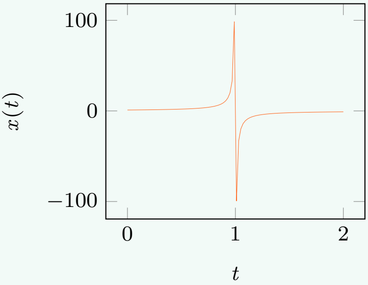
Consider the initial value problem

$$\dot{x}(t) = x^2(t), \quad t \in [0, 2]$$
$$x(0) = 1$$

The explicit closed-form solution

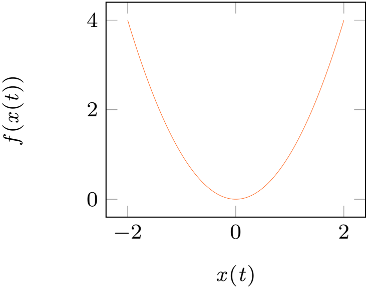
$$x(t) = \frac{1}{1-t}$$

$x(t)$ is only defined for $t \in [0, 1)$



Over the shorter interval $[0, T']$ with $T' < 1$, the solution exists and it is also unique

Continuous-time models (cont.)



Function $f\left(x(t)\right) = x^2(t)$ is not a globally Lipschitz continuous function

$$\frac{\| f\left(x^{\clubsuit}(t)\right) - f\left(x^{\spadesuit}(t)\right) \|}{\| x^{\clubsuit}(t) - x^{\spadesuit}(t) \|} \quad \not\leq \quad L$$

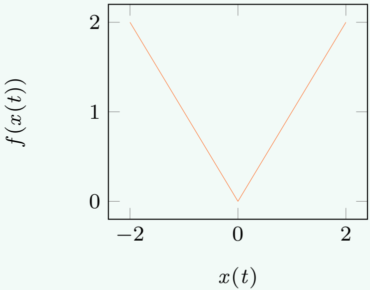
There is no single L that satisfies the inequality for all pairs $\left(x^{\clubsuit}(t), x^{\spadesuit}(t)\right)$

Function $x^2(t)$ is continuously differentiable with respect to $x(t)$, thus locally Lipschitz

□

Continuous-time models (cont.)

Example



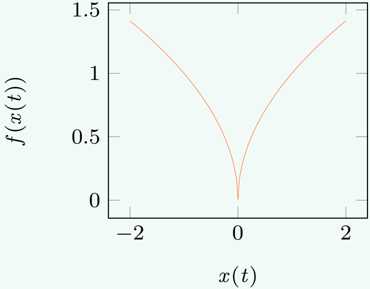
Is function $f\left(x(t)\right) = |x(t)|$ a globally Lipschitz continuous function?

$$\frac{\|f\left(x^{\clubsuit}(t)\right) - f\left(x^{\spadesuit}(t)\right)\|}{\|x^{\clubsuit}(t) - x^{\spadesuit}(t)\|} \leq L \quad (?)$$

If not, is it at least locally Lipschitz?

Continuous-time models (cont.)

Example



Is function $f(x(t)) = |x(t)|^{1/2}$ globally Lipschitz continuous?

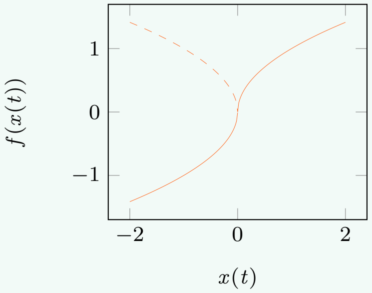
$$\frac{\|f(x^{\clubsuit}(t)) - f(x^{\spadesuit}(t))\|}{\|x^{\clubsuit}(t) - x^{\spadesuit}(t)\|} \leq L \quad (?)$$

If not, is it at least locally Lipschitz?

□

Continuous-time models (cont.)

Example



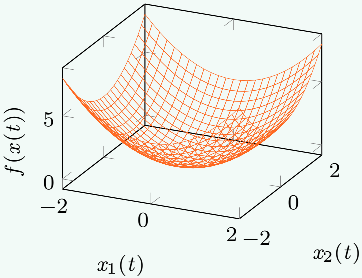
Is function $f\left(x(t)\right) = \text{sign}(x)|x(t)|^{1/2}$ globally Lipschitz continuous?

$$\frac{\|f\left(x^{\clubsuit}(t)\right) - f\left(x^{\spadesuit}(t)\right)\|}{\|x^{\clubsuit}(t) - x^{\spadesuit}(t)\|} \leq L \quad (?)$$

If not, is it at least locally Lipschitz?

Continuous-time models (cont.)

Example



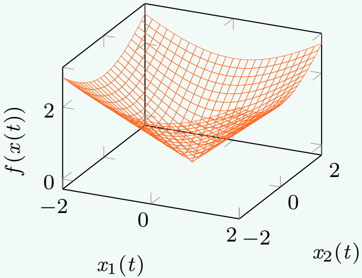
Is $f\left(x(t)\right) = \|x(t)\|_2^2$ a globally Lipschitz continuous function?

$$\frac{\|f\left(x^{\clubsuit}(t)\right) - f\left(x^{\spadesuit}(t)\right)\|}{\|x^{\clubsuit}(t) - x^{\spadesuit}(t)\|} \leq L \quad (?)$$

If not, is it at least locally Lipschitz?

Continuous-time models (cont.)

Example



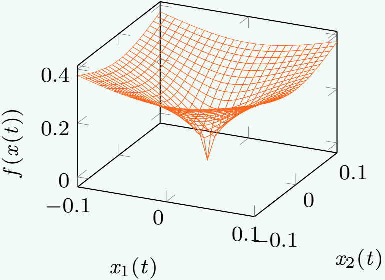
Is $f(x(t)) = \|x(t)\|_2$ a globally Lipschitz continuous function?

$$\frac{\|f\left(x^{\clubsuit}(t)\right) - f\left(x^{\spadesuit}(t)\right)\|}{\|x^{\clubsuit}(t) - x^{\spadesuit}(t)\|} \leq L \quad (?)$$

If not, is it at least locally Lipschitz?

Continuous-time models (cont.)

Example



Is $f\left(x(t)\right) = \|x(t)\|_2^{1/2}$ a globally Lipschitz continuous function?

$$\frac{\|f\left(x^{\clubsuit}(t)\right) - f\left(x^{\spadesuit}(t)\right)\|}{\|x^{\clubsuit}(t) - x^{\spadesuit}(t)\|} \leq L \quad (?)$$

If not, is it at least locally Lipschitz?

□

Continuous-time models (cont.)

Conditions for global and local existence, and uniqueness of the solution of an IVP are extended to systems with finitely many discontinuities of function f with respect to t

- The solution must be defined separately on each of the continuous subintervals
- At the discontinuity time-points, the derivative is not (strongly) defined

Continuity of the state trajectory is used to enforce the transition between subintervals

- (The end-state of one interval need be the initial state for the next one)

Continuous-time models (cont.)

Steady-state, **stationary**, **equilibrium**, or **fixed points**

- Values of x (fixed θ_x and u) such that $f\left(x(t)|\theta_x\right) = 0$

$$
\begin{aligned}
\frac{dx(t)}{dt} &= f\left(x(t)|\theta_x\right) \\
&= 0
\end{aligned}
$$

Stability

Consider the time evolution of a (set of) variable(s) of system originally at steady-state

- At some point in time, the system is perturbed, some change occurs

⇝ The system will respond to the perturbation, move away from SS

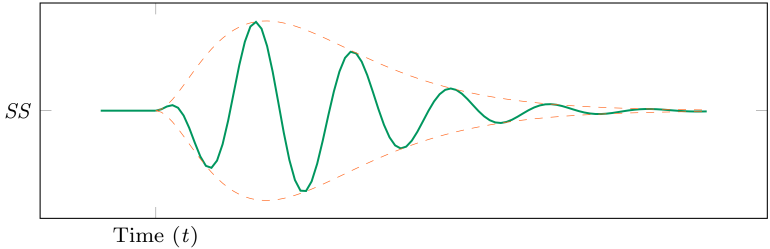
A system is stable if its variable(s) return autonomously to their steady-state value(s)

⇝ A *stable system* is also said to be a self-regulating process

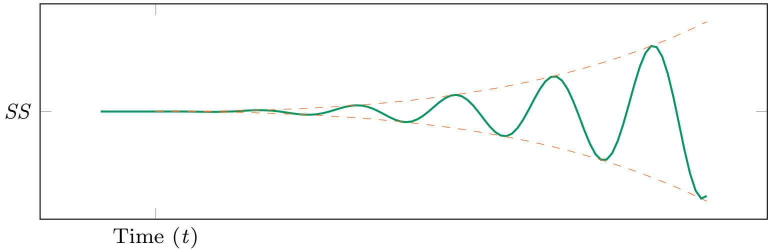
- A stable system would not need a controller, in general
- (If the steady-state condition is the desired state)
- (And, if we have an infinite amount of time)

Continuous-time models (cont.)

Stable



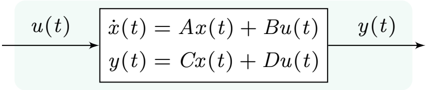
Unstable



Continuous-time models | LTIs

A very important class of dynamical system are linear time-invariant systems, or LTIs

Linear time-invariant systems, LTI



- $t \in \mathcal{R}$
- $x(t) \in \mathcal{R}^{N_x}$
- $u(t) \in \mathcal{R}^{N_u}$
- $A \in \mathcal{R}^{N_x \times N_x}$
- $B \in \mathcal{R}^{N_x \times N_u}$
- $\{A, B\} = \theta_x \in \mathcal{R}^{(N_x \times N_x) + (N_x \times N_u)}$

- $y(t) \in \mathcal{R}^{N_y}$
- $C \in \mathcal{R}^{N_y \times N_x}$
- $D \in \mathcal{R}^{N_y \times N_u}$
- $\{C, D\} = \theta_y \in \mathcal{R}^{(N_y \times N_x) + (N_y \times N_u)}$

Linear time-invariant systems $f = Ax + Bu$ are Lipschitz continuous with respect to x

- The global Lipschitz constant $L = \|A\|$

Continuous-time models | LTIs (cont.)

The solution to the analysis, for $t \geq t_{\text{ini}}$, an initial state $x(t_{\text{ini}})$ and an input $u(t \geq t_{\text{ini}})$

$$x(t) = e^{A(t-t_{\text{ini}})}x(t_{\text{ini}}) + \int_{t_{\text{ini}}}^{t} e^{A(t-\tau)}Bu(\tau)\text{d}\tau$$

$$y(t) = \underbrace{Ce^{A(t-t_{\text{ini}})}x(t_{\text{ini}}) + C\int_{t_{\text{ini}}}^{t} e^{A(t-\tau)}Bu(\tau)\text{d}\tau}_{Cx(t)} + Du(t)$$

The solution is known as the **Lagrange formula**

$\rightsquigarrow$ Based on the **state transition matrix**, e^{At}

Continuous-time models | LTIs (cont.)

Definition

Controllability of linear time-invariant systems

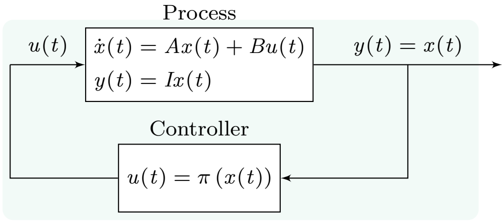
Consider a linear and time-invariant system (A, B), with $x(t) \in \mathcal{R}^{N_x}$ and $u(t) \in \mathcal{R}^{N_u}$

$$x(t) = Ax(t) + Bu(t)$$

The system is said to be **controllable**, if and only if it is possible to transfer the state of the system from any initial value $x_0 = x(0)$ to any other final value $x_f = x(t_f)$

- ..., only by manipulating the input $u(t)$
- ..., in some finite time $t_f \geq 0$

The final state x_f is called the **zero-state** or the **target-state**



Continuous-time models | LTIs (cont.)

Definition

Controllability gramian

Consider the linear and time-invariant system (A, B), with $x(t) \in \mathcal{R}^{N_x}$ and $u(t) \in \mathcal{R}^{N_u}$

$$x(t) = Ax(t) + Bu(t)$$

The system's **controllability gramian** is a $(N_x \times N_x)$ matrix, real and symmetric

$$W_c(t) = \int_0^t e^{A\tau} B B^T e^{A^T \tau} \mathrm{d}\tau$$

Theorem

Controllability test (I)

Consider the linear and time-invariant system (A, B), with $x(t) \in \mathcal{R}^{N_x}$ and $u(t) \in \mathcal{R}^{N_u}$

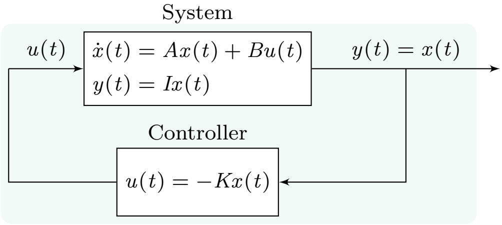
$$x(t) = Ax(t) + Bu(t)$$

Let $W_c(t) = \int_0^t e^{A\tau} B B^T e^{A^T \tau} \mathrm{d}\tau$ be the controllability gramian of the system

- The system is controllable iff $W_c(t)$ is non-singular, for all $t > 0$

State feedback (cont.))

We have system $\dot{x}(t) = Ax(t) + Bu(t)$, we can perfectly measure its state $x(t) = y(t)$



We design controllers that define an optimal control action $u(t)$, given the state $x(t)$

$$\rightsquigarrow \quad u(t) = -Kx(t)$$

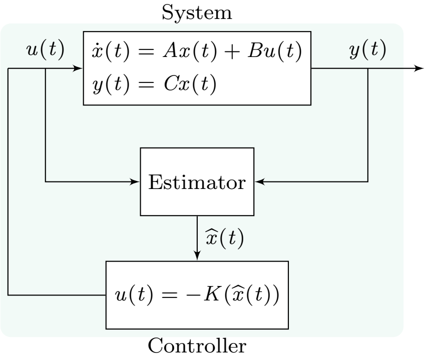
Linear-quadratic regulators (LQR) are model-based controllers

$$K = \left(B' Q_f B + R\right)^{-1} B' Q_f A$$

State estimation (cont.))

When we cannot measure the state, $x(t) \neq y(t)$, we design a device capable to estimate it from measurable quantities (data) and knowledge about the dynamics (a model)

The device that approximates the system's state is a **state observer**, or **estimator**



Were the state estimate $\widehat{x}(t)$ accurate, we could use it with the optimal controller $(-K)$

Continuous-time models | LTIs (cont.)

Definition

Observability of linear-time-invariant systems

Consider a linear and time-invariant system (A, C) with $x(t) \in \mathcal{R}^{N_x}$ and $u(t) \in \mathcal{R}^{N_y}$

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

The system is said to be **observable** if and only if it is possible to determine its state $x(t)$ from the force-free response of its measurements over a finite time $(t_f < \infty)$

- ..., from any arbitrary initial state $x(t_0)$

Continuous-time models | LTIs (cont.)

Definition

Observability gramian

Consider the linear and time-invariant system (A, C), with $x(t) \in \mathcal{R}^{N_x}$ and $y(t) \in \mathcal{R}^{N_y}$

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

The system's **observability gramian** is a $(N_x \times N_x)$ matrix, real and symmetric

$$W_o(t) = \int_0^t e^{A^T\tau} C^T C e^{A\tau} \mathrm{d}\tau$$

Continuous-time models | LTIs (cont.)

Theorem

Observability test (I)

Consider the linear and time-invariant system (A, C), with $x(t) \in \mathcal{R}^{N_x}$ and $y(t) \in \mathcal{R}^{N_y}$

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

Let $W_o(t) = \int_0^t e^{A^T\tau} C^T C e^{A\tau} \mathrm{d}\tau$ be the observability gramian of the system

- The system is observable iff $W_o(t)$ is non-singular, for all $t > 0$

Continuous-time models | LTIs (cont.)

Proof (Sufficient condition)

From the second Lagrange equation, we have the force-free evolution of the output

$$y(\tau) = Ce^{A\tau}x(0)$$

We left-multiply the equation by $e^{A^T\tau}$, then we integrate between 0 and some t_f

$$\begin{aligned}\int_0^{t_f} e^{A^T\tau}y(\tau)\mathrm{d}\tau &= \int_0^{t_f} e^{A^T\tau}Ce^{A\tau}x(0)\mathrm{d}\tau \\ &= W_o(t_f)x(0)\end{aligned}$$

Thus, we have

$$x(0) = W_o^{-1}(tf)\int_0^{t_f} e^{A^T\tau}Cy(\tau)\mathrm{d}\tau$$

The initial state is given as a function of the inverse of the observability gramian $W_o(tf)$ and the integral $\int_0^{t_f} e^{A^T\tau}Ce^{A\tau}y(\tau)\mathrm{d}\tau$ which can be computed from measurements $y(\tau)$

- The observability gramian need be non-singular for the inverse to exist

■

Continuous-time models | LTIs (cont.)

Definition

Luenberger observer

Consider a linear and time-invariant system, $x(t) \in \mathcal{R}^{N_x}$, $u(t) \in \mathcal{R}^{N_u}$, and $y(t) \in \mathcal{R}^{N_y}$

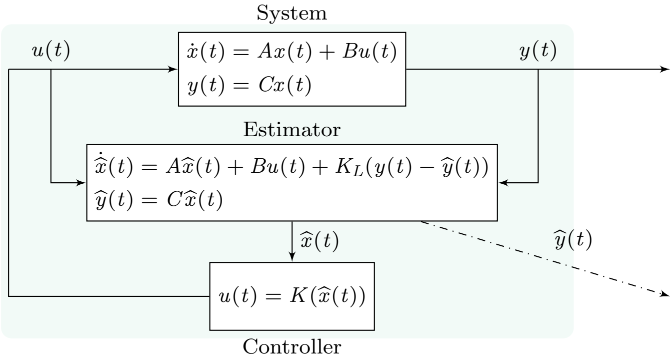
$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases},$$

The linear and time-invariant dynamical system

$$\begin{cases} \dot{\widehat{x}}(t) = A\widehat{x}(t) + Bu(t) + K_L\left(y(t) - \widehat{y}(t)\right) \\ \widehat{y}(t) = C\widehat{x}(t) \end{cases},$$

with $\widehat{x} \in \mathcal{R}^{N_x}$, $\widehat{y}(t) \in \mathcal{R}^{N_y}$ is a Luenberger observer of the system iff $K_L \in \mathcal{R}^{N_x \times N_y}$ is any matrix such that the eigenvalues of matrix $A - K_L C$ all have a negative real part

Continuous-time models | LTIs (cont.)



Luenberger observers are asymptotic state observers that are also model-based

- Kalman filters are stochastic counterpart, linear-quadratic estimators

Continuous-time models | DAEs

A class of system models combine differential states $x(t)$ and algebraic states $z(t)$

- The derivative of function $z(t)$ is not expressed explicitly in the model
- $z(t)$ is determined implicitly by an algebraic (set of) equation(s), h

(Time-invariant) Differential algebraic systems, DAE

$u(t) \rightarrow$

$$
\begin{aligned}
\dot{x}(t) &= f\left(x(t), u(t), z(t) | \theta_x\right) \\
0 &= h\left(x(t), u(t), z(t) | \theta_z\right) \\
y(t) &= g\left(x(t), z(t), u(t) | \theta_y\right)
\end{aligned}
$$

$\rightarrow y(t)$

- $\rightsquigarrow$ $x(t) \in \mathcal{R}^{N_x}$
- $\rightsquigarrow$ $u(t) \in \mathcal{R}^{N_u}$
- $\rightsquigarrow$ $z(t) \in \mathcal{R}^{N_z}$
- $\rightsquigarrow$ $\theta_x \in \mathcal{R}^{N_{\theta_x}}$
- $\rightsquigarrow$ $\theta_z \in \mathcal{R}^{N_{\theta_z}}$
- $\rightsquigarrow$ $t \in \mathcal{R}$

- $\rightsquigarrow$ $y(t) \in \mathcal{R}^{N_y}$
- $\rightsquigarrow$ $\theta_y \in \mathcal{R}^{N_{\theta_y}}$

The algebraic equations cannot be solved independently of the differential equations

Continuous-time models | DAE (cont.)

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \\ \vdots \\ \dot{x_{N_x}}(t) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} f_1\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_x\right) \\ f_2\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_x\right) \\ \vdots \\ f_{N_x}\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_x\right) \\ h_1\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_z\right) \\ h_2\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_z\right) \\ \vdots \\ h_{N_z}\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_z\right) \end{bmatrix}$$

Uniqueness of a numerical solution requires non-singularity of the Jacobian of h wrt z

$$\det\left(\frac{\partial h\left(x(t), u(t), z(t)\right)}{\partial z}\right) \neq 0$$

These specific differential algebraic equations are known as index-one DAE

Continuous-time models | DAE (cont.)

Function $h : \mathcal{R}^{N_x} \times \mathcal{R}^{N_u} \times \mathcal{R}^{N_z} \rightarrow \mathcal{R}^{N_z}$,

$$h\left(x(t), u(t), z(t) | \theta_x\right) = \begin{bmatrix} h_1\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_z\right) \\ h_2\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_z\right) \\ \vdots \\ h_{N_z}\left(x_1(t), \dots, x_{N_x}(t), u_1(t), \dots, u_{N_u}(t), z_1(t), \dots, z_{N_z}(t) | \theta_z\right) \end{bmatrix}$$

The Jacobian of h with respect to the algebraic state variables z

$$\frac{\partial h\left(x(t), u(t), z(t)\right)}{\partial z} = \underbrace{\begin{bmatrix} \left[\partial h_1(x,u,z)/\partial z_1 \right. & \cdots & \partial h_1(x,u,z)/\partial z_{n_z} & \cdots & \left. \partial h_1(x,u,z)/\partial z_{N_z}\right] \\ \left[\partial h_2(x,u,z)/\partial z_1 \right. & \cdots & \partial h_2(x,u,z)/\partial z_{n_z} & \cdots & \left. \partial h_2(x,u,z)/\partial z_{N_z}\right] \\ & & \vdots & & \\ \left[\partial h_{n_z}(x,u,z)/\partial z_1 \right. & \cdots & \partial h_{n_z}(x,u,z)/\partial z_{n_z} & \cdots & \left. \partial h_{n_z}(x,u,z)/\partial z_{N_z}\right] \\ & & \vdots & & \\ \left[\partial h_{N_z}(x,u,z)/\partial z_1 \right. & \cdots & \partial h_{N_z}(x,u,z)/\partial z_{n_z} & \cdots & \left. \partial h_{N_z}(x,u,z)/\partial z_{N_z}\right] \end{bmatrix}(t)}_{N_z \times N_z}$$

Continuous-time models | DAE (cont.)

Any index-one differential-algebraic equation can be differentiated with respect to time

- This allows for a practical numerical solution using ODE integrators

Because we have that $h\left(x(t), z(t)\right) = 0$, we also have

$$\frac{dh\left(x(t), z(t)\right)}{dt} = 0$$

For the total derivative of the algebraic equations, we have

$$\begin{aligned}\frac{dh\left(x(t), z(t)\right)}{dt} &= \frac{\partial h\left(x(t), z(t)\right)}{\partial z} \underbrace{\frac{dz(t)}{dt}}_{\dot{z}(t)} + \frac{\partial h\left(x(t), z(t)\right)}{\partial x} \underbrace{\frac{dx(t)}{dt}}_{f(x(t),z(t))} \\ &= 0\end{aligned}$$

Using the non-singularity of the Jacobian with respect to z, we have

$$\dot{z}(t) = -\underbrace{\left(\frac{\partial h\left(x(t), z(t)\right)}{\partial z}\right)^{-1}} \frac{\partial h\left(x(t), z(t)\right)}{\partial x} f\left(x(t), z(t)\right)$$

□

Continuous-time models (cont.)

A differential model describes the microscopic (in time) behaviour of process $(x(t))_{t\geq 0}$

- That is, the motion of the state in an infinitesimal time period

Consider a tiny time interval Δt, then $f\left(x(t)\right)$ is approximately constant over $[0, \Delta t]$

$$\begin{aligned} x(\Delta t) &= x_0 + \int_0^{\Delta t} f\left(x(t)\right) dt \\ &\approx x_0 + f\left(x_0\right) \int_0^{\Delta t} dt \\ &= x_0 + f\left(x_0\right) [t]_0^{\Delta t} \\ &= x_0 + f\left(x_0\right) \Delta t \end{aligned}$$

More generally, the discretisation of infinitesimal dynamics over intervals $[t, t+\Delta t]$

$$\begin{aligned} x(t+\Delta t) &= x(t) + \int_t^{t+\Delta t} f\left(x(\tau)\right) d\tau \\ &\approx x(t) + f\left(x(t)\right) \Delta t \end{aligned}$$

Equivalently, we have

$$\underbrace{x(t+\Delta t) - x(t)}_{\Delta x(t)} \approx f\left(x(t)\right) \Delta t$$

Continuous-time models (cont.)

$$x(t+\Delta t) \approx x(t) + f\left(x(t)\right)\Delta t$$

To approximate the evolution of process $(x(t))_{t=0}^{T}$, we divide the interval in K pieces

- For simplicity, we would typically let the size of each piece be $\Delta t = \dfrac{T-0}{K}$
- We apply the discretisation scheme on each piece, from x_0 at $t=0$

$$\begin{aligned}
x(1\Delta t) &= x(0) + f\left(x(0)\right)\Delta t \\
x(2\Delta t) &= x(1\Delta t) + f\left(x(1\Delta t)\right)\Delta t \\
\cdots &= \cdots \\
x(k\Delta t) &= x((k-1)\Delta t) + f\left(x((k-1)\Delta t)\right)\Delta t \\
\cdots &= \cdots
\end{aligned}$$

$$x(\underbrace{(K-1)\Delta t}_{T-\Delta t}) = x(\underbrace{(K-1)\Delta t}_{T-2\Delta t}) + f\left(x(\underbrace{(K-1)\Delta t}_{T-2\Delta t})\right)\Delta t$$

$$x(\underbrace{K\Delta t}_{T}) = x(\underbrace{(K-1)\Delta t}_{T-\Delta t}) + f\left(x(\underbrace{(K-1)\Delta t}_{T-\Delta t})\right)\Delta t$$

Continuous-time models (cont.)

$$x(t) = x_0 + \int_0^t f\left(x(\tau), u(\tau)\right) d\tau$$

Consider a tiny time interval Δt, then $f\left(x(t), u(t)\right)$ is approximately constant in $[0, \Delta t]$

$$\begin{aligned} x(\Delta t) &= x_0 + \int_0^{\Delta t} f\left(x(t), u(t)\right) dt \\ &\approx x_0 + f\left(x_0, u_0\right) \int_0^{\Delta t} dt \\ &= x_0 + f\left(x_0, u_0\right) \Delta t \end{aligned}$$

The discretisation of infinitesimal dynamics over intervals $[t, t + \Delta t]$

$$\begin{aligned} x(t + \Delta t) &= x(t) + \int_t^{t+\Delta t} f\left(x(\tau), u(\tau)\right) d\tau \\ &\approx x(t) + f\left(x(t), u(t)\right) \Delta t \end{aligned}$$

After we divide the interval in K pieces, the approximation of the evolution of $\left(x(t)\right)_{t=0}^{T}$

$$x(k\Delta t) = x((k-1)\Delta t) + f\left(x((k-1)\Delta t), u((k-1)\Delta t)\right) \Delta t \quad (k = 1, \dots K)$$

□

Continuous-time models (cont.)

The inputs are generated by a computer and implemented as piecewise constant signals

Zero-order hold controls

That is, the input $u(t)$ is kept constant between two equally spaced times, t_k and t_{k+1}

- We define the times when the control is applied as **sampling times**
- We let the sampling times be $\{t_k = k\Delta t\}_{k=0}^{K}$
- Δt denotes the (common) duration

The sampling interval Δt need not be the same one we used for approximating $(x(t))$

$$\{x(t_k = k\Delta t)\}_{k=0}^{K}$$

Zero-order holding is the operation of keeping a signal constant for $t \in [t_k, t_{k+1})$

Continuous-time models (cont.)

Suppose that $\dot{x}(t) = f\left(x(t), u(t)|\theta_x\right)$ is differentiable and that the inputs are piecewise constant with fixed values $u(t) = u_k$ with $u_k \in \mathcal{R}^{N_u}$ over each interval, for $t \in [t_k, t_{k+1})$

We can treat the transition from state $x(t_k)$ to $x(t_{k+1})$ as a discrete-time system

- The time in which the system evolves takes values only on a time grid

$$0 \cdots t_1 \cdots t_2 \cdots\cdots t_{k-1} \cdots \underbrace{t_k \cdots t_{k+1}}_{\Delta t} \cdots\cdots t_{K-1} \cdots t_K$$

In each interval $(t_k, t_{k+1}]$, the solution to the individual IVP exists and it is unique

- With initial value $x(t_k) = x_{\text{init}}$

Continuous-time models (cont.)

We consider the initial value problem, $x(0) = x_{\text{ini}}$ and constant control $u(t) = u_{\text{const}}$

$$\begin{aligned} \dot{x}(t) &= f\left(x(t), u_{\text{const}} | \theta_x\right), \quad t \in [0, \Delta t] \\ x(0) &= x_{\text{ini}} \end{aligned}$$

The unique solution $x : [0, \Delta t] \mapsto \mathcal{R}^{N_x}$ to the IVP with x_{init} and u_{const} is a function

- The arguments are: 1) the initial state x_{ini} and 2) the constant control u_{const}

The solution is the state trajectory over the short interval $[0, \Delta t]$

$$x(t \mid x_{\text{ini}}, u_{\text{const}}; \theta_x), \quad t \in [0, \Delta t]$$

The map from pair $(x_{\text{init}}, u_{\text{const}})$ to process $(x(t))_0^{\Delta t}$ is denoted as the **solution map**

The final value $x(t = \Delta t | x_{\text{init}}, u_{\text{const}}, \theta_x)$ of this short trajectory is important

- $x(\Delta t)$ defines the initial state of the next initial value problem

$$\begin{aligned} \dot{x}(t) &= f\left(x(t), u_{\text{const}} | \theta_x\right), \quad t \in [\Delta t, 2\Delta t] \\ x(\Delta t) &= x_{\text{ini}} \end{aligned}$$

Continuous-time models (cont.)

We define the **transition function** which returns that final value $x(\Delta t|x_{\text{ini}}, u_{\text{const}}; \theta_x)$

$$f_{\Delta t} : \mathcal{R}^{N_x} \times \mathcal{R}^{N_u} \rightarrow \mathcal{R}^{N_x}$$

The transition function returns the state $x(\Delta t|x_{\text{ini}}, u_{\text{const}}; \theta_x)$, given x_{ini} and u_{const}

$$x(\Delta t|x_{\text{ini}}, u_{\text{const}}; \theta_x) = f_{\Delta t}\left(x_{\text{ini}}, u_{\text{const}}|\theta_x\right)$$

$f_{\Delta t}$ is used to define a discrete-time system whose evolution describes the state at $\{t_k\}$

$$x(t_{k+1}) = f_{\Delta t}\left(x(t_k), u_k|\theta_x\right) \qquad (k = 0, 1, \dots K)$$

When we discuss general dynamical system, we will often refer to discrete-time systems

- The transition function $f_{\Delta t}$ may be only available implicitly
- (Often, we will define it as a computer routine/function)

Continuous-time models (cont.)

For linear and time-invariant dynamical systems $\dot{x}(t) = Ax(t) + Bu(t)$ with $x(0) = x_{\text{init}}$ and constant input u_{const}, the solution map $x(t|x_{\text{ini}}, u_{\text{ini}}, \theta_x)$ is explicitly known

$$x(t|x_{\text{ini}}, u_{\text{ini}}, \theta_x) = \underbrace{e^{At}x_{\text{ini}} + \int_0^t e^{A(t-\tau)}Bu_{\text{const}}\,d\tau}_{f_{\Delta t}(x_{\text{ini}}, u_{\text{const}}|\theta_x)}$$

$$= \underbrace{e^{At}x_{\text{ini}} + \left(\int_0^t e^{A(t-\tau)}\,d\tau\right)Bu_{\text{const}}}_{f_{\Delta t}(x_{\text{ini}}, u_{\text{const}}|\theta_x)}$$

The corresponding discrete-time system with sampling time Δt is linear time-invariant

$$x(t_{k+1}) = \underbrace{A_{\Delta t}x(t_k) + B_{\Delta t}u_k}_{f_{\Delta t}(x(t_k), u_k|\theta_x)}, \qquad (k = 0, 1, \dots, K-1)$$

$\rightsquigarrow$ $A_{\Delta t} = e^{A\Delta t}$ and $B_{\Delta t} = \left(\int_0^{\Delta t} e^{A(\Delta t - \tau)}\,d\tau\right)B$

Since Δt is fixed, also $A_{\Delta t}$ and $B_{\Delta t}$ are fixed (their elements are not function of time)

- LTI continuous-time system (A, B) maps to LTI discrete-time system $(A_{\Delta t}, B_{\Delta t})$

□

Discrete-time models

We describe a controlled dynamical system in discrete-time with a difference equation

$$x_{k+1} = f_k\left(x_k, u_k | \theta_x\right), \quad k \in \mathcal{N}_{0 \rightsquigarrow K-1}$$

- $\rightsquigarrow$ $K+1$ state vectors, $x_0, x_1, \dots, x_k, \dots, x_K \in \mathcal{R}^{N_x}$
- $\rightsquigarrow$ K input vectors, $u_0, u_1, \dots, u_k, \dots, u_{K-1} \in \mathcal{R}^{N_u}$
- $\rightsquigarrow$ Some time-horizon of length K
- $\rightsquigarrow$ Parameter vector $\theta_x \in \mathcal{R}^{N_{\theta_x}}$
- $\rightsquigarrow$ (Time-varying dynamics)

Given the initial state x_0 and all the controls $u_0, u_1, \dots, u_{K-1}$, we could recursively call the functions $f_k\left(x_k, u_k | \theta_x\right)$ and sequentially obtain all the other states $x_1, x_2, \dots, x_K$

- This recursion is known as **forward simulation** of the system dynamics

Discrete-time models (cont.)

Definition

Forward simulation

The forward simulation of the system dynamics is formally defined as a function

- The argument are x_0 and the collection $u_0, u_1, \dots, u_{K-1}$
- The image is the collection $x_0, x_1, \dots, x_K$

That is, we have

$$\begin{aligned} f_{\text{sim}} &: \mathcal{R}^{N_x + (K \times N_u)} \rightarrow \mathcal{R}^{(K+1)N_x} \\ &: (x_0, u_0, u_1, \dots, u_{K-1}) \mapsto (x_0, x_1, \dots, x_K) \end{aligned}$$

Function f_{sim} is defined by the recursive solution of the problem

$$x_{k+1} = f_k\left(x_k, u_k | \theta_x\right) \qquad (\text{for all } k \in \mathcal{N}_{0 \rightsquigarrow K-1})$$

Discrete-time models | LTI

Linear time-invariant systems, LTI

$$x_{k+1} = Ax_k + Bu_k, \quad k \in \mathcal{N}_{0 \rightsquigarrow K-1}$$

- $x_0, x_1, \dots, x-K, \dots, x_K \in \mathcal{R}^{N_x}$
- $u_0, u_1, \dots, u_k, \dots, u_{K-1} \in \mathcal{R}^{N_u}$
- $A \in \mathcal{R}^{N_x \times N_x}$
- $B \in \mathcal{R}^{N_x \times N_u}$
- $\{A, B\} = \theta_x \in \mathcal{R}^{(N_x \times N_x)+(N_x \times N_u)}$

The forward simulation map of linear time-invariant systems with horizon of length K

$$f_{\text{sim}}(x_0, u_0, \dots, u_{K-1}) = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix}$$

$$= \begin{bmatrix} x_0 \\ Ax_0 + Bu_0 \\ A^2 x_0 + ABu_0 + Bu_1 \\ \vdots \\ A^K x_0 + \sum_{k=0}^{K-1} A^{K-1-k} Bu_k \end{bmatrix}$$

Discrete-time models | LTI (cont.)

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix} = \underbrace{\begin{bmatrix} x_0 \\ Ax_0 + Bu_0 \\ A^2 x_0 + ABu_0 + Bu_1 \\ \vdots \\ A^K x_0 + \sum_{k=0}^{K-1} A^{K-1-k} Bu_k \end{bmatrix}}_{f_{\text{sim}}(x_0, u_0, \dots, u_{K-1})}$$

Consider the terminal value x_K after K steps from x_0 and subjected to $u_0 \rightsquigarrow u_{K-1}$,

$$x_K = A^K x_0 + \underbrace{\begin{bmatrix} A^{K-1}B & A^{K-2}B & \cdots & B \end{bmatrix}}_{\mathcal{C}_K} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{K-1} \end{bmatrix}$$

Matrix $\mathcal{C}_K$ is the discrete-time controllability matrix of the linear time-invariant system

- The discrete-time version because based on the discrete pair (A, B)

Discrete-time models | Affine

Affine time-varying systems are an important generalisation of the plain LTI model

Affine time-varying systems

$$x_{k+1} = A_k x_k + B_k u_k + c_k, \quad k \in \mathcal{N}_{0 \rightsquigarrow K-1}$$

- $x_0, x_1, \dots, x_k, \dots, x_K \in \mathcal{R}^{N_x}$
- $u_0, u_1, \dots, u_k, \dots, u_{K-1} \in \mathcal{R}^{N_u}$
- $A_0, A_1, \dots, A_k, \dots, A_K \in \mathcal{R}^{N_x \times N_x}$
- $B_0, B_1, \dots, B_k, \dots, B_K \in \mathcal{R}^{N_x \times N_u}$
- $\{A_k, B_k\} = \theta_x \in \mathcal{R}^{(N_x \times N_x)+(N_x \times N_u)}$

Affine time-varying systems arise from trajectory linearisations of nonlinear models

$$x_{k+1} = f_k \left(x_k, u_k | \theta_x \right)$$

- Linearisation of nonlinear (and time-varying) dynamics around point $(\overline{x}_k, \overline{u}_k)$
- We assume the that point $(\overline{x}_k, \overline{u}_k)$ is a term in a trajectory $\{(x_k, u_k)\}$
- (For example, $\{\overline{x}_0, \overline{x}_1, \dots, \overline{x}_K\}$ and $\{\overline{u}_0, \overline{u}_1, \dots, \overline{u}_{K-1}\}$)

$$\dot{x}(t) = f_f\left(x(t), u(t)|\theta_x\right)$$

In continuous-time, we would approximate (nonlinear and time-varying) dynamics f^t with a first-order Taylor's expansion around the point $(\overline{x}(t), \overline{u}(t))$ along the trajectory

After defining the deviation variables $x'(t) = x(t) - \overline{x}(t)$ and $u'(t) = u(t) - \overline{u}(t)$,

$$\underbrace{\begin{bmatrix} x'_1(t) \\ \vdots \\ x'_{N_x}(t) \end{bmatrix}}_{\dot{x}'(t)} = \underbrace{\begin{bmatrix} \dfrac{\partial f_1^t}{\partial x_1} & \cdots & \dfrac{\partial f_1^t}{\partial x_{N_x}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_{N_x}^t}{\partial x_1} & \cdots & \dfrac{\partial f_{N_x}^t}{\partial x_{N_x}} \end{bmatrix}_{(\overline{x}(t),\overline{u}(t))}}_{A^t} \underbrace{\begin{bmatrix} x'_1(t) \\ \vdots \\ x'_{N_x}(t) \end{bmatrix}}_{x'(t)} + \underbrace{\begin{bmatrix} \dfrac{\partial f_1^t}{\partial u_1} & \cdots & \dfrac{\partial f_1^t}{\partial u_{N_u}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_{N_x}^t}{\partial u_1} & \cdots & \dfrac{\partial f_{N_x}^t}{\partial u_{N_u}} \end{bmatrix}_{(\overline{x}(t),\overline{u}(t))}}_{B^t} \underbrace{\begin{bmatrix} u'_1(t) \\ \vdots \\ u'_{N_u}(t) \end{bmatrix}}_{u'(t)}$$

$$+ \underbrace{\begin{bmatrix} f_1^t \\ f_2^t \\ \vdots \\ f_{N_x}^t \end{bmatrix}_{(\overline{x}(t),\overline{u}(t))}}_{c^t}$$

- A^t is the Jacobian of f^t with respect to x, at $(\overline{x}(t), \overline{u}(t))$
- B^t is the Jacobian of f^t with respect to u, at $(\overline{x}(t), \overline{u}(t))$
- c^t is f^t evaluated at $(\overline{x}(t), \overline{u}(t))$

Discrete-time models | Affine (cont.)

The affine continuous-time approximation expressed in terms of deviation variables,

$$\underbrace{\begin{bmatrix} x_1'(t) \\ \vdots \\ x_{N_x}'(t) \end{bmatrix}}_{\dot{x}'(t)} = \underbrace{\begin{bmatrix} a_{1,1}^t & \cdots & a_{1,N_x}^t \\ \vdots & \ddots & \vdots \\ a_{N_x,1}^t & \cdots & a_{N_x,N_x}^t \end{bmatrix}}_{(N_x \times N_x)} \underbrace{\begin{bmatrix} x_1'(t) \\ \vdots \\ x_{N_x}'(t) \end{bmatrix}}_{(N_x \times 1)} + \underbrace{\begin{bmatrix} b_{1,1}^t & \cdots & b_{1,N_u}^t \\ \vdots & \ddots & \vdots \\ b_{N_x,1}^t & \cdots & b_{N_x,N_u}^t \end{bmatrix}}_{(N_x \times N_u)} \underbrace{\begin{bmatrix} u_1'(t) \\ \vdots \\ u_{N_u}'(t) \end{bmatrix}}_{(N_u \times 1)}$$

$$+ \underbrace{\begin{bmatrix} c_1^t \\ c_2^t \\ \vdots \\ c_{N_x}^t \end{bmatrix}}_{N_x \times 1}$$

Discrete-time models | Affine (cont.)

$$x_{k+1} = f_k\left(x_k, u_k | \theta\right)$$

Similarly, we can approximate nonlinear and time-varying dynamics in discrete-time

We have the affine time-varying system,

$$\underbrace{x_{k+1} - \overline{x}_{k+1}}_{x'_{k+1}} = f_k\left(x_k, u_k\right) - \overline{x}_{k+1}$$

$$\approx \underbrace{\frac{\partial f}{\partial x}\Big|_{(\overline{x}_k, \overline{u}_k)}}_{A_k \in \mathcal{R}^{N_x \times N_x}} \underbrace{\left(x_k - \overline{x}_k\right)}_{x'_k} + \underbrace{\frac{\partial f}{\partial u}\Big|_{(\overline{x}_k, \overline{u}_k)}}_{B_k \in \mathcal{R}^{N_x \times N_u}} \underbrace{\left(u_k - \overline{u}_k\right)}_{u'_k} + \underbrace{f_k\left(\overline{x}_k, \overline{u}_k\right) - \overline{x}_{k+1}}_{c_k \in \mathcal{R}^{N_x \times 1}}$$

The forward simulation map of affine time-varying systems, for a horizon of length K

$$x_K = \left(A_{K-1} \cdots A_0\right) x_0 + \sum_{k=0}^{K-1} \left(\prod_{j=k+1}^{K-1} A_j\right) \left(B_k u_k + c_k\right)$$

□

Numerical simulations

Dynamical models and numerical simulations

Numerical simulations

The design/deployment of optimal controllers depends on the availability of efficient/ accurate numerical simulation tools that build discretisations of continuous dynamics

We know that the IVP $\dot{x}(t) = f\left(x(t), u(t)|\theta_x\right)$ with $x(0) = x_0$ has a unique solution when f is Lipschitz continuous with respect to $x(t)$ and continuous with respect to t

$\rightsquigarrow$ A solution exists on the interval $[0, T]$, even if time $T > 0$ is arbitrary small

Numerical simulation methods compute approximate solutions to some well-posed IVP

- (Well-posedness is in the sense of the existence/uniqueness theorem)

For practical reasons, numerical simulation methods can be categorised in two groups

- Single-step methods and multi-step methods

Typically, each group is then divided into two main subgroups

- Explicit methods and implicit methods

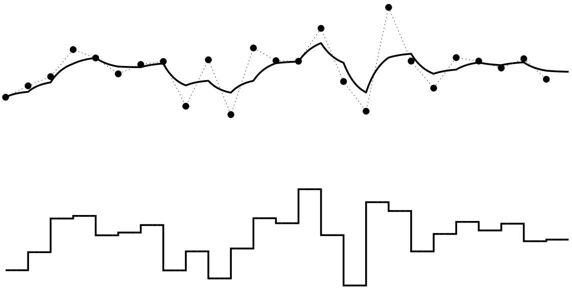
Numerical simulations (cont.)

The idea of a numerical simulation method is to compute an approximation to a solution map $x(t|x_{\text{ini}}, u_{\text{const}}; \theta_x)$ for $t \in [0, T]$, the computation is known as an **integrator**

⇝ Remember, the function from pair $(x_{\text{ini}}, u_{\text{const}})$ to process $\{x(t)\}_0^T$

An intuitive way to compute an approximation for $x(t|x_{\text{init}}, u_{\text{const}}; \theta_x)$ when $t \in [0, T]$

- Perform a linear extrapolation, based on the time derivative of $x(t)$
- From the initial point x_{init}, under constant controls u_{const}
- (The time-derivative is the $\dot{x}(t) = f\left(x(t), u(t)|\theta_x\right)$)



Numerical simulations | Explicit Euler

The approach is an explicit Euler integration step, a good approximation if T is tiny

$$x(t|x_{\text{init}}, u_{\text{const}}; \theta_x) \approx \underbrace{x(0|x_{\text{init}}, u_{\text{const}}; \theta_x)}_{x_{\text{ini}}} + \underbrace{f\,(x_{\text{init}}, u_{\text{const}}|\theta_x)\,(t-0)}_{tf(x_{\text{init}}, u_{\text{const}}|\theta_x)} \quad t \in [0, T]$$
$$= \widehat{x}(t|x_{\text{ini}}, u_{\text{const}}; \theta_x)$$

The error of the explicit Euler integration step is of order T^2, it grows as T^2 grows

- Or informally, the approximation error is small if T is very small
- The error is directly related to the truncation in the expansion

Numerical simulations | Explicit Euler (cont.)

The practical implementation of the explicit **explicit Euler integration method**

We consider a now longer interval with $t \in [0, T]$ and we divide it in K subintervals

$$0 \cdots 1 \cdots 2 \cdots \cdots (k-1) \cdots \underbrace{k \cdots (k+1)}_{\Delta t} \cdots \cdots (K-1) \cdots K$$

- Typically, we set each subinterval to have the same time-length

$$\Delta t = \frac{T}{K}$$

- We denote the K time points $\{t_k\}$ as nodes in the time grid

---

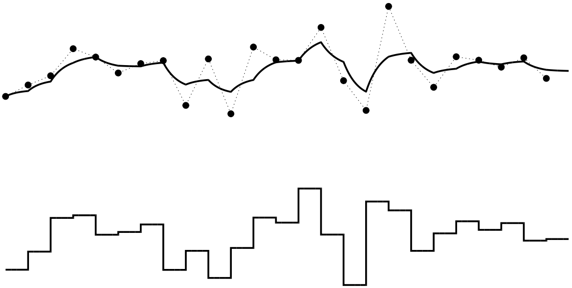
Starting from $\widehat{x}_0 = x_{\text{init}}$, we then perform K sequential linear extrapolation steps

$$\widehat{x}_{k+1} = \widehat{x}_k + f\left(\widehat{x}_k, u_{\text{const}} | \theta_x\right) \Delta t, \quad k = 0, 1, \ldots, K-1$$

For notational simplicity, we set the indexing for k to start from zero

- This allows us to start the sequence with $\widehat{x}_0 = x_{\text{ini}}$

Numerical simulations | Explict Euler (cont.)



Sequentially, the individual integration steps

- ⇝ $k = 0$

$$\widehat{x}_1 = \widehat{x}_0 + f\left(\widehat{x}_0, u_{\text{const}} | \theta_x\right) \Delta t$$

- ⇝ $k = 1$

$$\widehat{x}_2 = \widehat{x}_1 + f\left(\widehat{x}_1, u_{\text{const}} | \theta_x\right) \Delta t$$

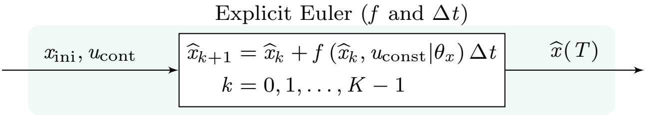
- ⇝ $\cdots$

$$\cdots$$

- ⇝ $k = K - 1$

$$\widehat{x}_K = \widehat{x}_{K-1} + f\left(\widehat{x}_{K-1}, u_{\text{const}} | \theta_x\right) \Delta t$$

Numerical simulations | Explicit Euler (cont.)



To compute the approximation $\widehat{x}_{k+1}$ at node $k+1$, an explicit Euler integration only requires information related to node k, specifically the numerical approximation $\widehat{x}_k$

- (The method is presented assuming that the dynamics are time-invariant)

The local (at k) approximation error gets smaller with the 'length' of the subintervals

- Using smaller (more) subintervals would lead to more accurate approximations

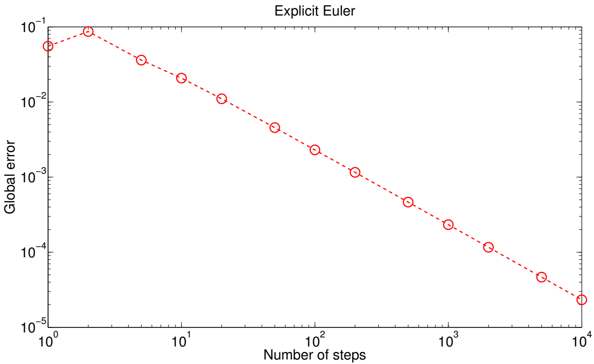
The Euler method is stable as the propagation of local errors is bounded by a constant

$$\underbrace{\left\| \widehat{x}(T | x_{\text{init}}, u_{\text{const}}, \theta_x) - x(T | x_{\text{init}}, u_{\text{const}}, \theta_x) \right\|}_{\text{Accumulated approximation error}}$$

Numerical simulations | Explicit Euler (cont.)

The consistency error of each subinterval is of order $(\Delta t)^2$ and there are $\dfrac{T}{\Delta t}$ subintervals

- The global, accumulated, error at the final time has order $(\Delta t)^2 \dfrac{T}{\Delta t} = T\Delta t$



The error function is linear in the number of function evaluations, slope equal to one

Numerical simulations | Explicit Euler (cont.)

This would suggest running integration procedures with many small-sized subintervals

- ⇝ The scheme requires the evaluation of function $f\left(x_{\text{ini}}, u_{\text{const}} | \theta_x\right)$ at each step
- ⇝ Good approximations with many steps require many function evaluations

(Other methods can achieve the desired accuracy levels with lower computational cost)

□

Numerical simulations | Explicit Runge-Kutta

The **order-4 Runge-Kutta integration method, RK4** generates a sequence of values $\widehat{x}_k$, by evaluating (and store) function f four times at each node k, from $\widehat{x}_0 = x_{\text{init}}$

From approximation $\widehat{x}_k$ and with constant input u_{const}, at each node k we have

$$\begin{aligned}
\kappa_1 &= f\left(\widehat{x}_k, u_{\text{const}} | \theta_x\right) \\
\kappa_2 &= f\left(\widehat{x}_k + \frac{\Delta t}{2}\kappa_1, u_{\text{const}} | \theta_x\right) \\
\kappa_3 &= f\left(\widehat{x}_k + \frac{\Delta t}{2}\kappa_2, u_{\text{const}} | \theta_x\right) \\
\kappa_4 &= f\left(\widehat{x}_k + \Delta t \kappa_3, u_{\text{const}} | \theta_x\right)
\end{aligned}$$

Each function evaluation is explicit and performed around the approximation point $\widehat{x}_k$

- The evaluations are stored as $\kappa_i \in \mathcal{R}^{N_x}$, $i \in \{1, 2, 3, 4\}$

The evaluations are then combined to construct the next approximation $\widehat{x}_{k+1}$ point

$$\widehat{x}_{k+1} = \widehat{x}_k + \frac{h}{6}\left(\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4\right), \quad k = 0, 1, \ldots, K-1$$

Numerical simulations | Explicit Runge-Kutta (cont.)

The solution map obtained by using an explicit Runge-Kutta method of order-4, RK4

Explicit Runge-Kutta (f and Δt)

$x_{\text{ini}}, u_{\text{cont}}$ →

$$\widehat{x}_{k+1} = \widehat{x}_k + \frac{h}{6}(\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4)$$
$$k = 0, 1, \ldots, K-1$$

→ $\widehat{x}(T)$

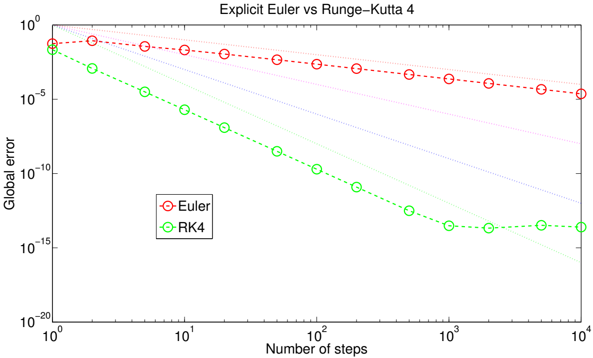
It can be understood as a continuous and differentiable nonlinear function

- The maximum order of differentiability depends on function f

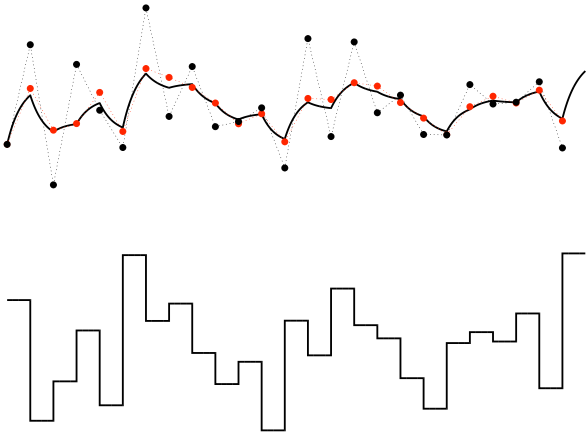
Numerical simulations | Explicit Runge-Kutta (cont.)

One step of the RK4 method is as expensive as four Euler steps, though more accurate

- The accumulated approximation error has order $T(\Delta t)^4$



Numerical simulations | Explicit Runge-Kutta (cont.)



Numerical simulations (cont.)

Summarising, consider a numerical simulation scheme over some time interval $[t_0, t_f]$

- The subintervals have a length $\Delta t = (t_0 - t_f)/K$

$$t_0 \cdots t_1 \cdots t_2 \cdots\cdots t_{k-1} \cdots \underbrace{t_k \cdots t_{k+1}}_{\Delta t} \cdots\cdots t_{K-1} \cdots t_K$$

- The nodes are indexed as $k = 0, 1, \dots, K$
- The position of the nodes

$$t_k := t_0 + k\Delta t, \quad k = 0, 1, \dots, K$$

The solution is approximated at nodes t_k by the values

$$\widehat{x}_k \approx x(t_k | x(t_0), u_{\text{const}}; \theta_x) \qquad (k = 0, 1, \dots, K)$$

Convergence

We define the order-p convergence of a method as worst-case local approximation error

$$\max_{k=0,\dots,K} \|\widehat{x}_k - x(t_k)\| = \mathcal{O}\left((\Delta t)^p\right)$$

As $K \to \infty$, we expect that $\widehat{x}_k$ gets closer to $x(t_k)$