① DYNAMIC SYSTEMS AND OPTIMIZATION

1

* DYNAMIC SYSTEMS
  - Continuous/discrete time
  - Continuous/discrete state (Infinite/Finite dimens)
  - Continuous/discrete controls
  - Time variant/Time invariant
  - Linear/Nonlinear
  - Controlled/Uncontrolled
  - Stable/unstable
  - Deterministic/Stochastic

CONTINUOUS TIME

- Initial Value Problems
- Linear time-Invariant sys
- Zero-order hold solution
- Solution map of linear time invariant

DISCRETE TIME *

- Forward simulation
- Linear time invariant system

---

* OPTIMISATION
  - Finite/Infinite dimensional
  - Continuous/Infinite dimensional Optimization
  - Continuous/Integer Optimisation
  - Convex/Nonconvex Optimisation

→ DYNAMIC SYSTEMS (1a)

Consider a continuous-time dynamical system $\boxed{\dot{x} = f(x,u)}$

TIME DEPENDENCE

STATE VARIABLES

CONTROL VARIABLES

→ LINEAR TIME INVARIANT SYSTEMS

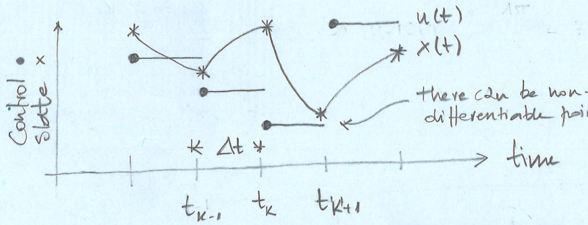
$\dot{x} = Ax + Bu$

WE WANT TO CONVERT THE SYSTEM FROM CONTINUOUS- TO DISCRETE-TIME

→ We first introduce the concept of 'ZERO-ORDER HOLD'

→ WE HAVE A CONTINUOUS TIME SYSTEM AND WE NEED TO APPLY CONTROL ACTIONS IN A DIGITAL FASHION

→ "WE HAVE A CONTROL ACTION, WE HOLD IT UNTIL THE NEW CONTROL ACTION IS AVAILABLE"



$\boxed{\text{PIECEWISE CONSTANT CONTROL } u(t)}$

The state will evolve in continuous-time, driven by its own dynamics and the constant control ⟶ $x(t)$

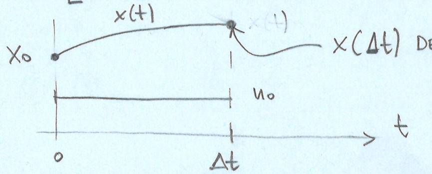
WE BUILD A MAP BETWEEN THE STATE AT TIME t_k AND t_{k+1}

$x(t)$ $x(t+1)$

We consider a dynamical system $f(x,u) = \dot{x}$

- AT TIME $t=0$ WE HAVE $x(0) = x_0$

→ IN $[0, \Delta t]$, WE HAVE THAT $\dot{x} = f(x, u_0)$, FOR SOME CONSTANT CONTROL u_0



$x(\Delta t)$ DEPENDS ONLY ON x_0 AND u_0 (the rest is system dynamics)

$$\underbrace{x(\Delta t \mid x_0, u_0)}_{\text{SOLUTION MAP}}$$

DO1

For nonlinear systems the solution map need often be approximated by a numerical routine

Derivatives of dynamic systems are needed for optimal control

THEY ARE OFTEN DENOTED AS 'SENSITIVITIES'

$$\boxed{\text{In general: for LTI is} \quad f_d(x_k, u_k) = A_d x_k + B_d u_k}$$

CONTINUOUS- AND DISCRETE-TIME DYNAMICAL SYSTEM

$t \in \mathbb{R}_+$, $k \in \mathbb{N}_+$

CONTINUOUS- AND DISCRETE-STATE DYNAMICAL SYSTEM

$x \in \mathbb{X} \subseteq \mathbb{R}^N$

$x \in \mathbb{X}$ FINITE AND COUNTABLE

CONTINUOUS- AND DISCRETE-CONTROL SETS

TIME-VARIANT AND TIME-INVARIANT DYNAMICAL SYSTEMS

LINEAR- AND NONLINEAR DYNAMICAL SYSTEMS

CONTROLLED AND UNCONTROLLED DYNAMICAL SYSTEMS

STABLE AND UNSTABLE DYNAMICAL SYSTEMS

DETERMINISTIC AND STOCHASTIC DYNAMICAL SYSTEMS

DO

Many systems of interest are described in the form DIFFERENTIAL EQUATIONS

→ Numerical simulations are based on a discretization of the time interval of interest

→ This generates discrete-time systems

A CONTROLLED DYNAMIC SYSTEM IN CONTINUOS TIME CAN BE DESCRIBED BY AN ORDINARY DIFFERENTIAL EQUATION ON THE INTERVAL $[0,T]$

$$\dot{x} = f(x(t), u(t), t) \quad , \quad t \in [0,T] \qquad \begin{cases} x \in \mathbb{R}^{N_x} \\ u \in \mathbb{R}^{N_u} \end{cases}$$

$$x(t=0) = x_0$$

$x(t)$ IS THE STATE VARIABLE

$u(t)$ IS THE CONTROL INPUT

FUNCTION f MAPS FROM STATE VARS, CONTROLS AND TIME ONTO THE RATE OF CHANGE OF THE STATE VARS

$$f : \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \times [0,T] \longrightarrow \mathbb{R}^{N_x}$$

— // — // —

A special class of continuous-time dynamical system (LINEAR and TIME INVARIANT)

$$\dot{x} = Ax + Bu$$

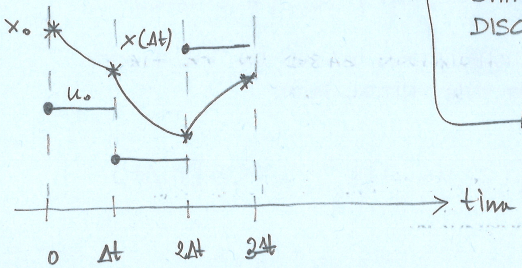
$$\begin{cases} A \in \mathbb{R}^{N_x} \times \mathbb{R}^{N_x} \\ B \in \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \end{cases} \longrightarrow f(x(t), u(t), t) = Ax(t) + Bu(t)$$

WITH THIS NOTION, WE DEFINE A DISCRETE-TIME SYSTEM

$$f_d(x_0,u_0) = x^+ \triangleq \underbrace{x(\Delta t|x_0,u_0)} \qquad \text{(Def.)}$$

IMPORTANT **

IF WE KNOW A PIECEWISE-CONSTANT CONTROL, WE CAN COMPUTE THE STATE TRAJECTORY OVER TIME, AT DISCRETE TIME INTERVALS



- IT CAN BE COMPUTED ANALYTICALLY (FOR SOME SYSTEMS)
- IT CAN/MUST BE APPROXIMATED NUMERICALLY (FOR MOST SYSTEMS)

— SOLUTION MAP FOR LINEAR-TIME-INVARIANT SYSTEMS

For linear time-invariant systems, the solution map has an analytical form for continuous time

$$x(t|x_0,u_0) = \underbrace{e^{A t}}_{\text{MATRIX}} x_0 + \int_0^{t} \underbrace{e^{A(t-\tau)}}_{\text{MATRIX}} B \underbrace{u(\tau)} d\tau$$

FIXED, FOR FIXED SAMPLING TIME

CONSTANT IN $[0,\Delta t]$ (CAN BE MOVED OUTSIDE THE INTEGRAL) $\Rightarrow u_0$

→ WITH SAMPLING TIME Δt

$$x(\Delta t|x_0,u_0) = \underbrace{A_d} x_0 + \underbrace{B_d} u_0$$

$0 \to \Delta t$ gives us

$$A_d = e^{A\Delta t}$$

$0 \to \Delta t$ gives us

$$B_d = \int_0^{\Delta t} e^{A(\Delta t-\tau)} B \, d\tau$$

If $f(x,u)$ is differentiable, then also the solution map is differentiable

D02

Euler integration is a simple (and crude) way to generate an approximation for $x(t|x_0, u_0 = \text{constant})$ for $t \in [0,\Delta t]$

→ IT PERFORMS A LINEAR EXTRAPOLATION BASED ON THE FIRST DERIVATIVE $\dot{x} = f(x,u)$ AT THE INITIAL POINT

$$\tilde{x}(t|x_0,u_{const}) = x_0 + f(x_0,u_{const})\,t \qquad t \in [0,\Delta t]$$

1. WE DIVIDE THE INTERVAL $[0,\Delta t]$ INTO M SUBINTERVALS, EACH OF LENGTH $h = \Delta t/M$
2. WE PERFORM M SUCH LINEAR EXTRAPOLATION STEPS CONSECUTIVELY STARTING AT $\tilde{x}_0 = x_0$

$$\tilde{x}_{j+1} = \tilde{x}_j + h f(\tilde{x}_j, u_{const}) \qquad (j=0,\dots,M-1)$$

→ Euler's method is stable (ERROR PROPAGATION IS BOUNDED)

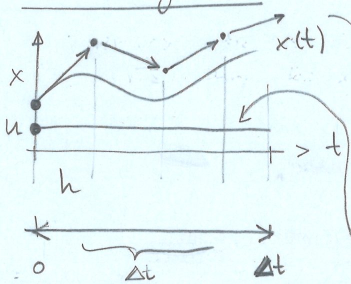
NUMERICAL INTEGRATION OF DYNAMICAL SYSTEMS

→ SAME AS SIMULATION (in time)
→ SOLUTION (typically numerical) OF ODEs

$$\dot{x} = f(x, u_{const}) \quad , \quad t \in [0, \Delta t]$$

SAMPLING TIME FOR THE CONTROLLER

Euler integration



Constant control eg

→ Approximated trajectory — it gets better as the step size h gets smaller

$$h = \frac{\Delta t}{M}$$

FOR EXAMPLE, THE SAMPLING TIME OF THE CONTROLLER

N° OF STEPS MUST GET LARGER

x_0, u_0 given as "INPUTS"
for $i = 0 \rightarrow M-1$
$$x_{i+1} = x_i + \underbrace{h \underbrace{f(x_i, u_0)}}$$
end

"OUTPUT"
$x_M = x(\Delta t)$

f NEEDS TO BE EVALUATED M TIMES

THE EULER INTEGRATION STEP

CONVERGENCE:

$$\| x_M - x(\Delta t) \| = O(h)$$

(MUST BE CHECKED)

Runge-Kutta integration (RK4, of order four)

LIKE EULER'S METHOD IT ALSO GENERATES A SEQUENCE OF VALUES $\tilde{x}_J$ WITH $j = 0, \ldots, M-1$ AND $\tilde{x}_0 = x_0$ (J IS LIKE i ABOVE)

AT $\tilde{x}_J$, USING THE CONSTANT INPUT u_{const}, ONE STEP OF RK4 COMPUTES

- $k_1 = f(\tilde{x}_J, u_{const})$
- $k_2 = f(\tilde{x}_J + h/2\, k_1, u_{const})$
- $k_3 = f(\tilde{x}_J + h/2\, k_2, u_{const})$
- $k_4 = f(\tilde{x}_J + h\, k_3, u_{const})$

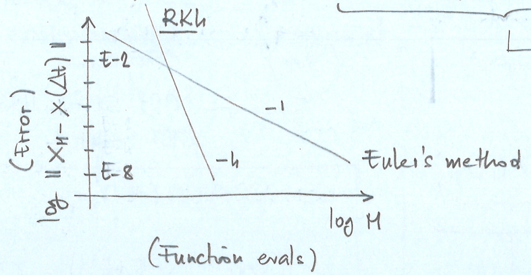
$$\xrightarrow{\text{then}} \tilde{x}_{J+1} = \tilde{x}_J + h/6\,(k_1 + 2k_2 + 2k_3 + k_4)$$

4-TIME MORE EXPENSIVE

D03

- RK4 HAS A BETTER EFFICIENCY THAN EULER'S
- EFFICIENCY HERE MEANS $\underbrace{\text{ACCURACY vs FUNCTION EVALS}}$

→ MOST IMPORTANT FEATURE OF A NUMERICAL METHOD



→ THIS IS THE TYPICAL ACCURACY RANGE IN PRACTICAL APPLICATIONS

DISCRETE-TIME SYSTEM (As before but slightly more formally)

Typically originate from the discretization of continuous-time sys.

Let assume that the discretization size is fixed
- → THERE IS NO PHYSICAL TIME t ⇝ $(x(t))$
- → WE USE INDEXES, $k \in \mathbb{N}$ ⇝ (x_k)

$\Big\}\; x \in \mathbb{R}^{N_x}$

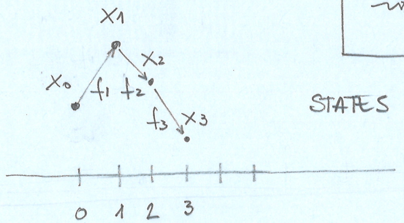
THE DISCRETE-TIME SYSTEM

$$x_{k+1} = f_k(x_k, u_k)$$

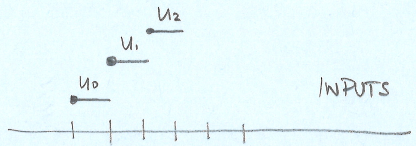
for $k = 0, 1, \dots, N-1$

SAYS WHETHER THE SYSTEM IS TIME-INVARIANT

THIS IS A SEQUENCE OF STATES AT SPECIFIC POINTS IN TIME
⇝ there is nothing in between



STATES (nothing in between)



INPUTS (piecewise constant)

If we know the initial value x_0 and we know the sequence of controls $u_0, u_1, \dots, u_{N-1}$, then we can determine all the other states ——— FORWARD SIMULATION

Def. (FORWARD SIMULATION)

$$f_{sim} : \mathbb{R}^{\underbrace{N_x}\, + \,\underbrace{N N_u}} \longrightarrow \mathbb{R}^{\underbrace{N_x (N+1)}}$$

Associates a trajectory (discretized) to an initial state and a sequence of inputs u

N_x → from all the states

$N N_u$ → from all the controls

$N_x(N+1)$ → to all the states of the trajectory

THE STATES ARE GENERATED ONE AT A TIME . . . RECURSIVELY

$$(x_0, u_0, u_1, \dots, u_{N-1}) \longrightarrow (x_0, x_1, \dots, x_N)$$

— // — ~ — // —

LINEAR TIME-INVARIANT SYSTEMS

$$f_{sim}(x_0, u_0, u_1, \dots, u_{N-1}) = \begin{cases} x_0 \\ Ax_0 + Bu_0 = x_1 \\ Ax_1 + Bu_1 = A(Ax_0 + Bu_0) + Bu_1 \\ \qquad\qquad = A^2 x_0 + ABu_0 + Bu_1 = x_2 \\ Ax_2 + Bu_2 = A^3 x_0 + A^2 B u_0 + ABu_1 + Bu_2 \\ \vdots \\ \underbrace{Ax_{N-1} + Bu_{N-1}}_{} = A^N x_0 + A^{N-1} Bu_0 + A^{N-2} Bu_1 \\ \qquad + \dots + Bu_{N-1} \end{cases}$$

WE HAVE THAT :

$$x_{k+1} = Ax_k + Bu_k$$

A: $N_x \times N_x$

B: $N_x \times N_u$

$$x_N = \underbrace{\begin{bmatrix} A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\text{CONTROLLABILITY MATRIX :-)}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

STABILITY FOR DISCRETE-TIME SYSTEM :

- All eigenvalues of A must be in the unit circle in the complex plane
- Asymptotically if the modulus is strictly smaller than 1 DOS

AFFINE SYSTEMS and LINEARISATION ALONG TRAJECTORIES

This is a generalization of linear systems

→ AFFINE TIME VARIANT SYSTEMS

$$x_{k+1} = A_k x_k + B_k u_k + c_k \quad , \quad \text{with } k = 0, 1, \_\_\_ , N-1$$

They are often the result of the linearization of nonlinear dynamic systems along a given reference trajectory

CONSIDER A NONLINEAR DYNAMIC SYSTEM AND SOME GIVEN REFERENCE $\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{N-1}$ AS WELL AS A SET OF CONTROLS $\tilde{u}_0, \tilde{u}_1, \dots, \tilde{u}_{N-1}$

→ the Taylor expansion of each function f_k at the reference value $(\tilde{x}_k, \tilde{u}_k)$

$$(x_{k+1} - \tilde{x}_{k+1}) \approx \left.\frac{\partial f_k}{\partial x}\right|_{(\tilde{x}_k, \tilde{u}_k)} (x_k - \tilde{x}_k) + \left.\frac{\partial f_k}{\partial u}\right|_{(\tilde{x}_k, \tilde{u}_k)} (u_k - \tilde{u}_k) + \left[f(\tilde{x}_k, \tilde{u}_k) - \tilde{x}_{k+1} \right]$$

EVEN FOR TIME-INVARIANT NONLINEAR DYNAMICS, THE LINEARISED DYNAMICS BECOME TIME-VARIANT BECAUSE OF THE DIFFERENT LINEARIZATION POINTS ON THE REFERENCE TRAJECTORY

IT IS IMPORTANT TO NOTE THAT THE FORWARD SIMULATION MAP OF AN AFFINE SYSTEM IS AGAIN AN AFFINE FUNCTION OF THE INITIAL VALUE AND THE CONTROLS

OPTIMIZATION (General idea) (1b)

MATHEMATICAL OPTIMIZATION REFERS TO THE PROBLEM OF FINDING THE BEST, THE OPTIMAL VALUE AMONG A SET OF POSSIBLE DECISIONS

→ SOME SOLUTION CANDIDATES ARE <u>FEASIBLE</u> (OTHERS ARE NOT)

→ IT IS ASSUMED THAT THE <u>FEASIBILITY</u> OF A SOLUTION CANDIDATE CAN BE CHECKED BY EVALUATION OF SOME <u>CONSTRAINT FUNCTION</u>

There exist a number of problem classes, we give a brief overview together with the basic concepts associated to it.

The main ingridients

- DECISION VARIABLES x (SETS OF SOLUTION CANDIDATES)
- OBJECTIVE FUNCTION $f(x)$ (CRITERION TO MAXIMISE OR MINIMISE)
- CONSTRAINTS (VALID SOLUTION SETS)

THE FIRST TWO CATEGORIES OF OPTIMIZATION PROBLEMS

$\boxed{\text{– FINITE / INFINITE DIMENSIONAL}}$

– REFERRED TO THE DECISION VARIABLES

* FINITE CASE $x \in \mathbb{R}^N$ N IS FINITE

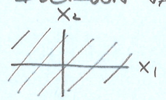
* INFINITE $x \in$ function space

→ SEMI-INFINITE OPTIMISATION { DECISION VARS ARE FINITE; CONSTRAINTS ARE INFINITE

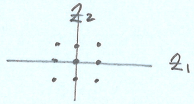
CONTINUOUS / INTEGER

— REFERRED TO THE TYPE OF DECISION VARIABLES

$x \in \mathbb{R}^N$, CONTINOUS



$x \in \mathbb{Z}^N$, INTEGER



* CONTINUOUS

$$\min_{x \in \mathbb{R}^N} f(x) \quad \text{(objective function)}$$

(Decision variables)

$$\text{s.t.} \quad g(x) = 0 \quad \text{(Equality constraints)}$$

$$h(x) \leq 0 \quad \text{(Inequality constraints)}$$

* INTEGER

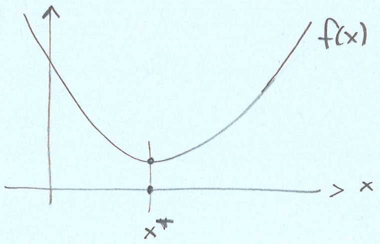
$$\left.\begin{aligned} &\min_{z \in \mathbb{Z}^N} f(z) \\ &\text{s.t.} \quad g(z) = 0 \\ &\qquad\; h(z) \leq 0 \end{aligned}\right\} \text{SAME TERMS AS IN THE CONTINUOUS CASE}$$

* MIXED INTEGER

$$\left.\begin{aligned} &\min_{\substack{x \in \mathbb{R}^N \\ z \in \mathbb{Z}^N}} f(x,z) \\ &\text{s.t.} \quad g(x,z) = 0 \\ &\qquad\; h(x,z) \leq 0 \end{aligned}\right\} \text{HARDEST OF THE THREE}$$

CONVEX / NON-CONVEX

CONVEX FEASIBLE SET + CONVEX OBJECTIVE FUNCTION



IF x^* IS A LOCAL MINIMUM,
THEN x^* IS A GLOBAL MINIMUM

EXAMPLES OF CONVEX OPTIMIZATION
- LINEAR PROGRAMMING
- QUADRATIC PROGRAMMING
- SECOND-ORDER CONE PROGRAMMING
- SEMI-DEFINITE PROGRAMMING
- QUADRATICALLY CONSTRAINED QUADRATIC PROGRAMMING

EXAMPLES OF NON-CONVEX OPTIMIZATION ☺
- NONLINEAR PROGRAMMING

Differentiable functions
f, g, h