Automatic Control 1

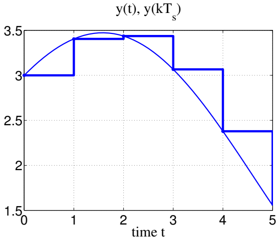
Discrete-time linear systems

Prof. Alberto Bemporad

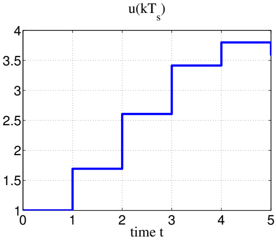
University of Trento

Academic year 2010-2011

Introduction



Sampling of a continuous signal



Discrete-time signal

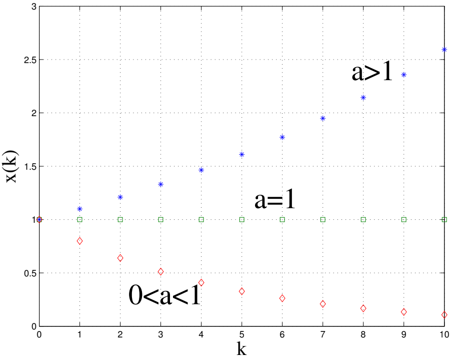
- Discrete-time models describe relationships between *sampled* variables $x(kT_s)$, $u(kT_s)$, $y(kT_s)$, $k = 0, 1, \ldots$
- The value $x(kT_s)$ is kept constant during the *sampling interval* $[kT_s, (k+1)T_s)$
- A discrete-time signal can either represent the *sampling* of a *continuous-time* signal, or be an intrinsically discrete signal
- Discrete-time signals are at the basis of *digital controllers* (as well as of digital filters in signal processing)

Difference equation

- Consider the first order *difference equation* (autonomous system)

$$\begin{cases} x(k+1) &= ax(k) \\ x(0) &= x_0 \end{cases}$$

- The solution is $x(k) = a^k x_0$



Difference equation

- First-order difference equation with input (non-autonomous system)

$$\begin{cases} x(k+1) = ax(k) + bu(k) \\ x(0) = x_0 \end{cases}$$

- The solution has the form

$$x(k) = \underbrace{a^k x_0}_{\text{natural response}} + \underbrace{\sum_{i=0}^{k-1} a^i bu(k-1-i)}_{\text{forced response}}$$

- The *natural response* depends on the initial condition $x(0)$, the *forced response* on the input signal $u(k)$

Example - Wealth of a bank account

- k: year counter
- ρ: interest rate
- $x(k)$: wealth at the beginning of year k
- $u(k)$: money saved at the end of year k
- x_0: initial wealth in bank account

Discrete-time model:

$$\begin{cases} x(k+1) = (1+\rho)x(k) + u(k) \\ x(0) = x_0 \end{cases}$$

x_0	10 k€
$u(k)$	5 k€
ρ	10 %

$$x(k) = (1.1)^k \cdot 10 + \frac{1-(1.1)^k}{1-1.1} 5 = 60(1.1)^k - 50$$

Linear discrete-time system

- Consider the set of n first-order linear difference equations forced by the input $u(k) \in \mathbb{R}$

$$\begin{cases} x_1(k+1) &= a_{11}x_1(k) + \ldots + a_{1n}x_n(k) & +b_1u(k) \\ x_2(k+1) &= a_{21}x_1(k) + \ldots + a_{2n}x_n(k) & +b_2u(k) \\ \vdots & \vdots & \vdots \\ x_n(k+1) &= a_{n1}x_1(k) + \ldots + a_{nn}x_n(k) & +b_nu(k) \\ x_1(0) = x_{10}, & \ldots \quad x_n(0) = x_{n0} \end{cases}$$

- In compact matrix form:

$$\begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ x(0) &= x_0 \end{cases}$$

where $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$.

Linear discrete-time system

- The solution is

$$x(k) = \underbrace{A^k x_0}_{\text{natural response}} + \underbrace{\sum_{i=0}^{k-1} A^i B u(k-1-i)}_{\text{forced response}}$$

- If matrix A is diagonalizable, $A = T\Lambda T^{-1}$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \Rightarrow A^k = T \begin{bmatrix} \lambda_1^k & 0 & \dots & 0 \\ 0 & \lambda_2^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n^k \end{bmatrix} T^{-1}$$

where $T = [v_1 \dots v_n]$ collects n independent eigenvectors.

Modal response

- Assume input $u(k) = 0$, $\forall k \geq 0$
- Assume A is diagonalizable, $A = T\Lambda T^{-1}$
- The state trajectory (natural response) is

$$x(k) = A^k x_0 = T\Lambda^k T^{-1} x_0 = \sum_{i=1}^{n} \alpha_i \lambda_i^k v_i$$

where

- λ_i = eigenvalues of A
- v_i = eigenvectors of A
- α_i = coefficients that depend on the initial condition $x(0)$

$$\alpha = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = T^{-1}x(0),\ T = [v_1 \ldots v_n]$$

- The system modes depend on the eigenvalues of A, as in continuous-time

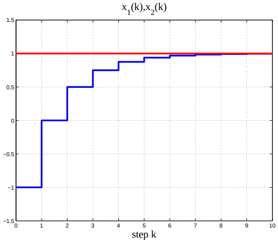
Example

- Consider the linear discrete-time system

$$\begin{cases} x_1(k+1) &= \frac{1}{2}x_1(k)+\frac{1}{2}x_2(k) \\ x_2(k+1) &= x_2(k)+u(k) \\ x_1(0) &= -1 \\ x_2(0) &= 1 \end{cases} \quad \Rightarrow \quad \begin{cases} x(k+1) &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ x(0) &= \begin{bmatrix} -1 \\ 1 \end{bmatrix} \end{cases}$$

- Eigenvalues of A: $\lambda_1 = \frac{1}{2}$, $\lambda_2 = 1$
- Solution:

$$\begin{aligned} x(k) &= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}^k \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \sum_{i=0}^{k-1} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}^i \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k-1-i) \\ &= \begin{bmatrix} \frac{1}{2^k} & 1-\frac{1}{2^k} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} + \sum_{i=0}^{k-1} \begin{bmatrix} 1-\frac{1}{2^i} \\ 1 \end{bmatrix} u(k-1-i) \\ &= \underbrace{\begin{bmatrix} 1-\left(\frac{1}{2}\right)^{k-1} \\ 1 \end{bmatrix}}_{\text{natural response}} + \underbrace{\sum_{i=0}^{k-1} \begin{bmatrix} 1-\frac{1}{2^i} \\ 1 \end{bmatrix} u(k-1-i)}_{\text{forced response}} \end{aligned}$$



simulation for $u(k) \equiv 0$

n^{th}-order difference equation

- Consider the *n^{th}-order difference equation* forced by u

$$a_n y(k-n) + a_{n-1} y(k-n+1) + \cdots + a_1 y(k-1) + y(k) =$$
$$b_n u(k-n) + \cdots + b_1 u(k-1) + b_0 u(k)$$

- Equivalent linear discrete-time system in *canonical state matrix form*

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} (b_n - b_0 a_n) & \dots & (b_1 - b_0 a_1) \end{bmatrix} x(k) + b_0 u(k) \end{cases}$$

- There are infinitely many state-space realizations

MATLAB
`tf2ss`

- n^{th}-order difference equations are very useful for digital filters, digital controllers, and to reconstruct models from data (*system identification*)

Discrete-time linear system

$$\begin{cases} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k)+Du(k) \\ x(0) &= x_0 \end{cases}$$

- Given the initial condition $x(0)$ and the input sequence $u(k)$, $k \in \mathbb{N}$, it is possible to predict the entire sequence of states $x(k)$ and outputs $y(k)$, $\forall k \in \mathbb{N}$
- The state $x(0)$ summarizes all the past history of the system
- The dimension n of the state $x(k) \in \mathbb{R}^n$ is called the *order* of the system
- The system is called *proper* (or *strictly causal*) if $D = 0$
- General multivariable case:

$$\begin{array}{rcl} x(k) & \in & \mathbb{R}^n \\ u(k) & \in & \mathbb{R}^m \\ y(k) & \in & \mathbb{R}^p \end{array} \qquad \begin{array}{rcl} A & \in & \mathbb{R}^{n\times n} \\ B & \in & \mathbb{R}^{n\times m} \\ C & \in & \mathbb{R}^{p\times n} \\ D & \in & \mathbb{R}^{p\times m} \end{array}$$

Example - Student dynamics

- Problem Statement:
  - 3-years undergraduate course
  - percentages of students promoted, repeaters, and dropouts are roughly constant
  - direct enrollment in 2nd and 3rd academic year is not allowed
  - students cannot enrol for more than 3 years
- Notation:

k	Year
$x_i(k)$	Number of students enrolled in year i at year k, $i = 1, 2, 3$
$u(k)$	Number of freshmen at year k
$y(k)$	Number of graduates at year k
α_i	promotion rate during year i, $0 \leq \alpha_i \leq 1$
β_i	failure rate during year i, $0 \leq \beta_i \leq 1$
γ_i	dropout rate during year i, $\gamma_i = 1 - \alpha_i - \beta_i \geq 0$

- 3[rd]-order linear discrete-time system:

$$\begin{cases} x_1(k+1) &= \beta_1 x_1(k) + u(k) \\ x_2(k+1) &= \alpha_1 x_1(k) + \beta_2 x_2(k) \\ x_3(k+1) &= \alpha_2 x_2(k) + \beta_3 x_3(k) \\ y(k) &= \alpha_3 x_3(k) \end{cases}$$

Example - Student dynamics

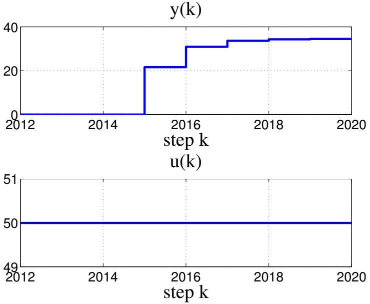
- In matrix form

$$\begin{cases} x(k+1) = \begin{bmatrix} \beta_1 & 0 & 0 \\ \alpha_1 & \beta_2 & 0 \\ 0 & \alpha_2 & \beta_3 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 0 & \alpha_3 \end{bmatrix} x(k) \end{cases}$$

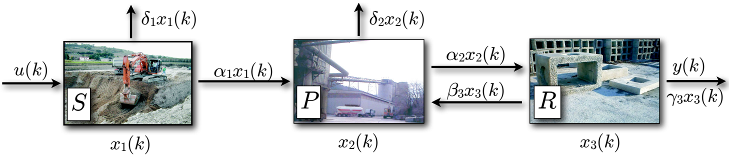
- Simulation

$\alpha_1 = .60$	$\beta_1 = .20$
$\alpha_2 = .80$	$\beta_2 = .15$
$\alpha_3 = .90$	$\beta_3 = .08$

$u(k) \equiv 50,\ k = 2012, \ldots$



Example - Supply chain



- Problem Statement:
  - S purchases the quantity $u(k)$ of raw material at each month k
  - a fraction δ_1 of raw material is discarded, a fraction α_1 is shipped to producer P
  - a fraction α_2 of product is sold by P to retailer R, a fraction δ_2 is discarded
  - retailer R returns a fraction β_3 of defective products every month, and sells a fraction γ_3 to customers
- Mathematical model:

$$\left\{\begin{array}{rcl} x_1(k+1) & = & (1-\alpha_1-\delta_1)x_1(k)+u(k) \\ x_2(k+1) & = & \alpha_1 x_1(k)+(1-\alpha_2-\delta_2)x_2(k)+\beta_3 x_3(k) \\ x_3(k+1) & = & \alpha_2 x_2(k)+(1-\beta_3-\gamma_3)x_3(k) \\ y(k) & = & \gamma_3 x_3(k) \end{array}\right.$$

k	month counter
$x_1(k)$	raw material in S
$x_2(k)$	products in P
$x_3(k)$	products in R
$y(k)$	products sold to customers

Equilibrium

- Consider the discrete-time nonlinear system

$$\begin{cases} x(k+1) & = & f(x(k),u(k)) \\ y(k) & = & g(x(k),u(k)) \end{cases}$$

Definition

A state $x_r \in \mathbb{R}^n$ and an input $u_r \in \mathbb{R}^m$ are an *equilibrium pair* if for initial condition $x(0) = x_r$ and constant input $u(k) \equiv u_r$, $\forall k \in \mathbb{N}$, the state remains constant: $x(k) \equiv x_r$, $\forall k \in \mathbb{N}$

- Equivalent definition: (x_r, u_r) is an equilibrium pair if $f(x_r, u_r) = x_r$
- x_r is called *equilibrium state*, u_r *equilibrium input*
- The definition generalizes to time-varying discrete-time nonlinear systems

Stability

- Consider the nonlinear system

$$\begin{cases} x(k+1) = f(x(k), u_r) \\ y(k) = g(x(k), u_r) \end{cases}$$

and let x_r an equilibrium state, $f(x_r, u_r) = x_r$

Definition

The equilibrium state x_r is *stable* if for each initial conditions $x(0)$ "close enough" to x_r, the corresponding trajectory $x(k)$ remains near x_r for all $k \in \mathbb{N}$ [a]

[a] Analytic definition: $\forall \epsilon > 0\ \exists \delta > 0 : \|x(0) - x_r\| < \delta \Rightarrow \|x(k) - x_r\| < \epsilon,\ \forall k \in \mathbb{N}$

- The equilibrium point x_r is called *asymptotically stable* if it is stable and $x(k) \rightarrow x_r$ for $k \rightarrow \infty$
- Otherwise, the equilibrium point x_r is called *unstable*

Stability of first-order linear systems

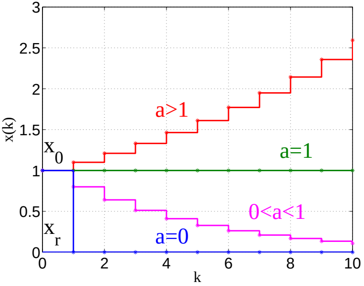
- Consider the first-order linear system

$$x(k+1) = ax(k) + bu(k)$$

- $x_r = 0$, $u_r = 0$ is an equilibrium pair
- For $u(k) \equiv 0$, $\forall k = 0, 1, \ldots$, the solution is

$$x(k) = a^k x_0$$

- The origin $x_r = 0$ is
  - unstable if $|a| > 1$
  - stable if $|a| \leq 1$
  - asymptotically stable if $|a| < 1$



Stability of discrete-time linear systems

Since the natural response of $x(k+1) = Ax(k) + Bu(k)$ is $x(k) = A^k x_0$, the stability properties depend only on A. We can therefore talk about *system stability* of a discrete-time linear system (A,B,C,D)

Theorem:

Let $\lambda_1, \ldots, \lambda_m$, $m \leq n$ be the eigenvalues of $A \in \mathbb{R}^{n \times n}$. The system $x(k+1) = Ax(k) + Bu(k)$ is

- asymptotically stable iff $|\lambda_i| < 1$, $\forall i = 1, \ldots, m$
- (marginally) stable if $|\lambda_i| \leq 1$, $\forall i = 1, \ldots, m$, and the eigenvalues with unit modulus have equal algebraic and geometric multiplicity [a]
- unstable if $\exists\ i$ such that $|\lambda_i| > 1$

[a] Algebraic multiplicity of λ_i = number of coincident roots λ_i of $\det(\lambda I - A)$. Geometric multiplicity of λ_i = number of linearly independent eigenvectors v_i, $Av_i = \lambda_i v_i$

The stability properties of a discrete-time linear system only depend on the **modulus** of the eigenvalues of matrix A

Stability of discrete-time linear systems

Proof:

- The natural response is $x(k) = A^k x_0$
- If matrix A is diagonalizable[1] , $A = T\Lambda T^{-1}$,

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \Rightarrow A^k = T \begin{bmatrix} \lambda_1^k & 0 & \dots & 0 \\ 0 & \lambda_2^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n^k \end{bmatrix} T^{-1}$$

- Take any eigenvalue $\lambda = \rho e^{j\theta}$:

$$|\lambda^k| = \rho^k |e^{jk\theta}| = \rho^k$$

- A is always diagonalizable if algebraic multiplicity - geometric multiplicity

□

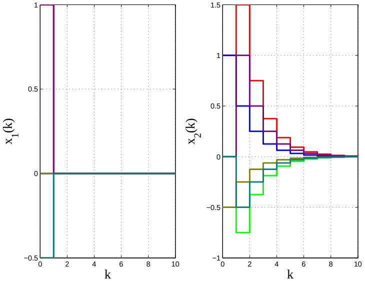
[1] If A is not diagonalizable, it can be transformed to Jordan form. In this case the natural response $x(t)$ contains modes $k^j \lambda^k$, $j = 0, 1, \dots$, alg. multiplicity = geom. multiplicity

Example 1

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 0 \\ 1 & \frac{1}{2} \end{bmatrix} x(k) \\ x(0) = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \end{cases} \Rightarrow \text{eigenvalues of } A\text{: } \left\{0, \frac{1}{2}\right\}$$

solution:

$$\begin{cases} x_1(k) &= 0,\ k = 1, 2, \ldots \\ x_2(k) &= \left(\frac{1}{2}\right)^{k-1} x_{10} + \left(\frac{1}{2}\right)^{k} x_{20},\ k = 1, 2, \ldots \end{cases}$$



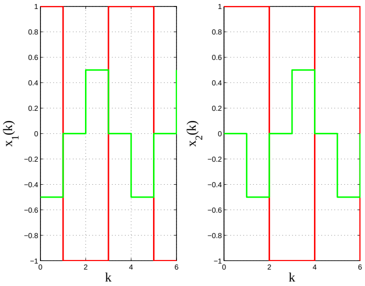
asymptotically stable

Example 2

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} x(k) \\ x(0) = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \end{cases} \Rightarrow \text{eigenvalues of } A\text{: } \{+j, -j\}$$

solution:

$$\begin{cases} x_1(k) = x_{10}\cos\frac{k\pi}{2} + x_{20}\sin\frac{k\pi}{2}, & k = 0, 1, \ldots \\ x_2(k) = x_{10}\sin\frac{k\pi}{2} + x_{20}\cos\frac{k\pi}{2}, & k = 0, 1, \ldots, \end{cases}$$



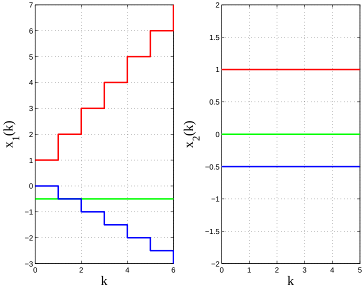
marginally stable

Example 3

$$\begin{cases} x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) \\ x(0) = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \end{cases} \Rightarrow \text{ eigenvalues of } A\text{: } \{1, 1\}$$

$$\begin{cases} x_1(k) &= x_{10} + x_{20}k, \; k = 0, 1, \ldots \\ x_2(k) &= x_{20}, \; k = 0, 1, \ldots \end{cases}$$

Note: A is not diagonalizable !



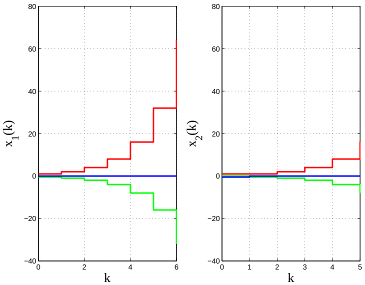
unstable

Example 4

$$\begin{cases} x(k+1) = \begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix} x(k) \\ x(0) = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \end{cases} \Rightarrow \text{ eigenvalues of } A\text{: } \{0, 2\}$$

solution:

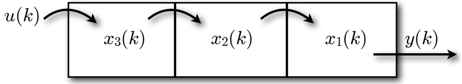
$$\begin{cases} x_1(k) = 2^k x_{10}, \ k = 0, 1, \ldots \\ x_2(k) = 2^{k-1} x_{10}, \ k = 1, 2, \ldots \end{cases}$$



unstable

Zero eigenvalues

- Modes λ_i=0 determine *finite-time* convergence to zero.
- This has no continuous-time counterpart, where instead all converging modes tend to zero in infinite time ($e^{\lambda_i t}$)
- Example: dynamics of a buffer



$$\begin{cases} x_1(k+1) = x_2(k) \\ x_2(k+1) = x_3(k) \\ x_3(k+1) = u(k) \\ y(k) = x_1(k) \end{cases} \Rightarrow \begin{cases} x(k+1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(k) \end{cases}$$

- Natural response: $A^3 x(0) = 0$ for all $x(0) \in \mathbb{R}^3$
- For $u(k) \equiv 0$, the buffer deploys after at most 3 steps !

Summary of stability conditions for linear systems

system		*continuous-time*	*discrete-time*
asympt. stable	$\forall i = 1, \ldots, n$	$\Re(\lambda_i) < 0$	$\lvert\lambda_i\rvert < 1$
unstable	$\exists i$ such that	$\Re(\lambda_i) > 0$	$\lvert\lambda_i\rvert > 1$
stable	$\forall i, \ldots, n$	$\Re(\lambda_i) \leq 0$	$\lvert\lambda_i\rvert \leq 1$
	and $\forall \lambda_i$ such that algebraic = geometric mult.	$\Re(\lambda_i) = 0$	$\lvert\lambda_i\rvert = 1$

Exact sampling

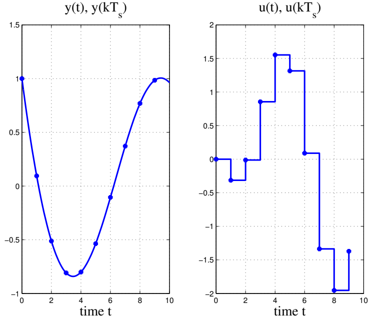
- Consider the continuos-time system

$$\begin{cases} \dot{x}(t) &= Ax(t)+Bu(t) \\ y(t) &= Cx(t)+Du(t) \\ x(0) &= x_0 \end{cases}$$

- We want to characterize the value of $x(t), y(t)$ at the time instants $t=0, T_s, 2T_s, \ldots, kT_s, \ldots$, **under the assumption that the input $u(t)$ is constant during each sampling interval** (*zero-order hold, ZOH*)

$$u(t)=\bar{u}(k),\ kT_s \leq t < (k+1)T_s$$

- $\bar{x}(k) \triangleq x(kT_s)$ and $\bar{y}(k) \triangleq y(kT_s)$ are the state and the output samples at the k^{th} sampling instant, respectively



Exact sampling

- Let us evaluate the response of the continuous-time system between time $t_0 = kT_s$ and $t = (k+1)T_s$ from the initial condition $x(t_0) = x(kT_s)$ using Lagrange formula:

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\sigma)}Bu(\sigma)d\sigma = e^{A((k+1)T_s-kT_s)}x(kT_s) + \int_{kT_s}^{(k+1)T_s} e^{A((k+1)T_s-\sigma)}Bu(\sigma)d\sigma$$

- Since the input $u(t)$ is piecewise constant, $u(\sigma) \equiv \bar{u}(k),\ kT_s \leq \sigma < (k+1)T_s$. By setting $\tau = \sigma - kT_s$ we get

$$x((k+1)T_s) = e^{AT_s}x(kT_s) + \left(\int_0^{T_s} e^{A(T_s-\tau)}d\tau\right)Bu(kT_s)$$

and hence

$$\bar{x}(k+1) = e^{AT_s}\bar{x}(k) + \left(\int_0^{T_s} e^{A(T_s-\tau)}d\tau\right)B\bar{u}(k)$$

which is a linear difference relation between $\bar{x}(k)$ and $\bar{u}(k)$!

Exact sampling

- The discrete-time system

$$\begin{cases} \bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\bar{u}(k) \\ \bar{y}(k) = \bar{C}\bar{x}(k) + \bar{D}\bar{u}(k) \end{cases}$$

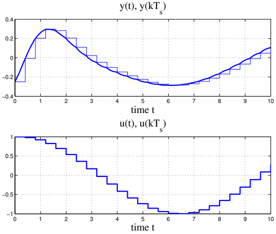
depends on the original continuous-time system through the relations

$$\bar{A} \triangleq e^{AT_s}, \quad \bar{B} \triangleq \left(\int_0^{T_s} e^{A(T_s-\tau)} d\tau\right) B, \quad \bar{C} \triangleq C, \quad \bar{D} \triangleq D$$

- If $u(t)$ is piecewise constant, $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ provides the *exact* evolution of state and output samples at discrete times kT_s

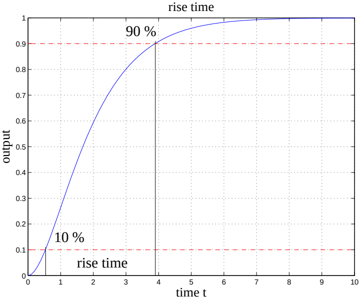
MATLAB

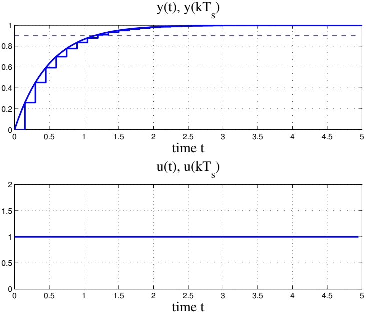
```
sys=ss(A,B,C,D);
sysd=c2d(sys,Ts);
[Ab,Bb,Cb,Db]=ssdata(sysd);
```



Exact sampling

Rule of thumb: $T_s \approx \frac{1}{10}$ of the *rise time* = time to move from 10% to 90% of the steady-state value, for input $u(t) \equiv 1$, $x(0) = 0$

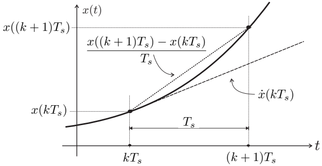




More on the choice of sampling time in the second part of the course ...

Approximate sampling - Euler's method

$$\dot{x}(kT_s) \approx \frac{x((k+1)T_s) - x(kT_s)}{T_s}$$



Leonhard Paul Euler
(1707-1783)

- For nonlinear systems $\dot{x}(t) = f(x(t), u(t))$:

$$\bar{x}(k+1) = \bar{x}(k) + T_s f(\bar{x}(k), \bar{u}(k))$$

- For linear systems $\dot{x}(t) = Ax(t) + Bu(t)$:

$$x((k+1)T_s) = (I + T_s A)x(kT_s) + T_s Bu(kT_s)$$

$$\bar{A} \triangleq I + AT_s, \quad \bar{B} \triangleq T_s B, \quad \bar{C} \triangleq C, \quad \bar{D} \triangleq D$$

- Note that $e^{T_s A} = I + T_s A + \ldots + \frac{T_s^n A^n}{n!} + \ldots$
  Therefore when T_s is small Euler's method and exact sampling are similar

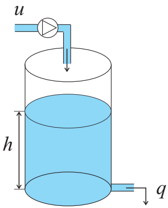
Example - Hydraulic system

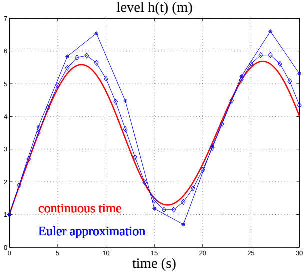
Continuous time:

$$\begin{cases} \frac{d}{dt}h(t) & = & -\frac{a\sqrt{2g}}{A}\sqrt{h(t)}+\frac{1}{A}u(t) \\ q(t) & = & a\sqrt{2g}\sqrt{h(t)} \end{cases}$$

Discrete time:

$$\begin{cases} \bar{h}(k+1) & = & \bar{h}(k)-\frac{T_s a\sqrt{2g}}{A}\sqrt{\bar{h}(k)}+\frac{T_s}{A}\bar{u}(k) \\ \bar{q}(k) & = & a\sqrt{2g}\sqrt{\bar{h}(k)} \end{cases}$$





Tustin's discretization method

- Assume $u(k)$ constant within the sampling interval. Given the linear system $\dot{x} = Ax + Bu$, apply the trapezoidal rule to approximate the integral

$$x(k+1) - x(k) = \int_{kT_s}^{(k+1)T_s} \dot{x}(t)dt = \int_{kT_s}^{(k+1)T_s} (Ax(t) + Bu(t))dt$$

$$\approx \frac{T_s}{2}(Ax(k) + Bu(k) + Ax(k+1) + Bu(k))$$ (trapezoidal rule)

and therefore

$$(I - \frac{T_s}{2}A)x(k+1) = (I + \frac{T_s}{2})x(k) + T_s Bu(k)$$

$$x(k+1) = \left(I - \frac{T_s}{2}A\right)^{-1}\left(I + \frac{T_s}{2}A\right)x(k) + \left(I - \frac{T_s}{2}A\right)^{-1} T_s Bu(k)$$

- Advantage: simpler to compute than exponential matrix, without too much loss of approximation quality

N-steps Euler method

- We can obtain the matrices A, B of the discrete-time linearized model while integrating the nonlinear continuous-time dynamic equations $\dot{x} = f(x,u)$
- *N-steps explicit forward Euler method*: given $x(k)$, $u(k)$, execute the following steps
  1. $x = x(k)$, $A = I$, $B = 0$
  2. for n=1:N do
     - $A \leftarrow (I + \frac{T_s}{N}\frac{\partial f}{\partial x}(x,u(k))A$
     - $B \leftarrow (I + \frac{T_s}{N}\frac{\partial f}{\partial x}(x,u(k))B + \frac{T_s}{N}\frac{\partial f}{\partial u}(x,u(k))A$
     - $x \leftarrow x + \frac{T_s}{N} f(x,u(k))$
  3. end
  4. return $x(k+1) \approx x$ and matrices A, B such that $x(k+1) \approx Ax(k) + Bu(k)$.
- Property: the difference between the state $x(k+1)$ and its approximation x computed by the above iterations satisfies $\|x(k+1) - x)\| = O\left(\frac{T_s}{N}\right)$
- Explicit forward Runge-Kutta 4 method also available

English-Italian Vocabulary

	
discrete-time linear systems	*sistemi lineari a tempo discreto*
sampling interval	*tempo (o intervallo) di campionamento*
difference equation	*equazione alle differenze*
zero-order hold	*mantenitore di ordine zero*
piecewise constant	*costante a tratti*
rise time	*tempo di salita*

Translation is obvious otherwise.