Lecture 6: Sampling of Linear Systems

[IFAC PB Ch. 1, Ch. 2, and Ch. 3 (to pg 23)]

- Effects of Sampling
- Sampling a Continuous-Time State-Space Model
- Difference Equations
- State-Space Models in Discrete Time

Textbook

The main text material for this part of the course is:

IFAC Professional Brief: Computer Control: An Overview (Educational Version 2011), Wittenmark, Åström, Årzen (IFAC PB)
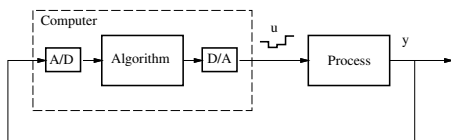
- Summary of the digital control parts of Åström and Wittenmark: Computer Controlled Systems
- Some new material

Chapter 10 and 11 are not part of this course (but can be useful in other courses)

Chapter 8, 13 and 14 partly overlap with RTCS.

Sampling

AD-converter acts as sampler

DA-converter acts as a hold device

Normally, zero-order-hold is used $\Rightarrow$ piecewise constant control signals

Hold Devices

Zero-Order Hold (ZOH) almost always.
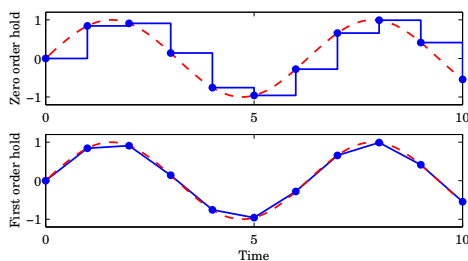
First-Order Hold (FOH):

- Signal between the conversions is a linear extrapolation

$$f(t) = f(kh) + \frac{t - kh}{h}(f(kh + h) - f(kh)) \quad kh \leq t < kh + h$$

- Advantages:
  - Often better reconstruction
  - Output continuous
- Disadvantages:
  - $f(kh + h)$ must be available at time kh
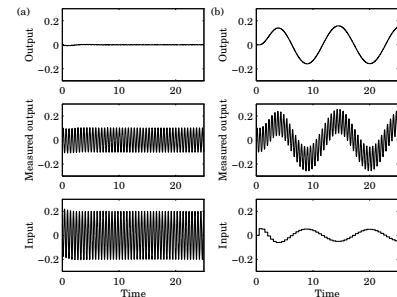 - Not supported by standard DA-converters

Hold Devices

In IFAC PB there are quite a lot of results presented for the first-order hold case. These are not part of this course.
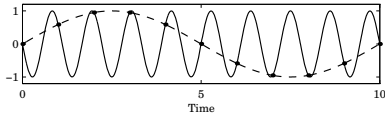
Dynamic Effects of Sampling

Continuous time control   Discrete time control

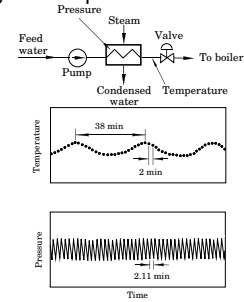Sampling with high frequency measurement noise may create new frequencies!

Aliasing

$\omega_N = \omega_s/2 =$ Nyquist frequency, ($\omega_s =$ sampling freq.)

Frequencies above the Nyquist frequency are folded and appear as low-frequency signals.
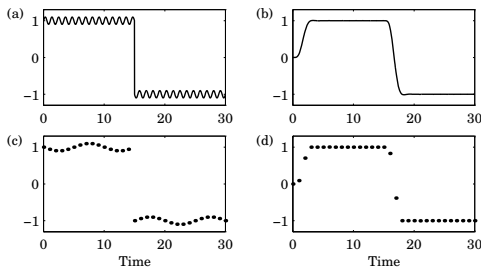
Aliasing – Example

Feed water heating in a ship boiler

Analog low-pass filter to remove high-frequency measurement noise
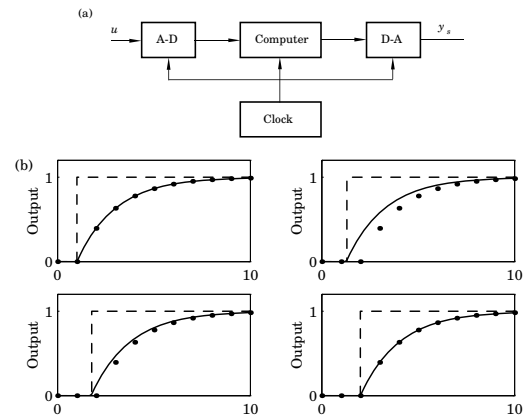
Example – Prefiltering

$\omega_d = 0.9$, $\omega_N = 0.5$, $\omega_{alias} = 0.1$
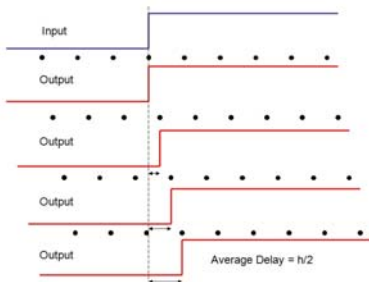
6th order Bessel with $\omega_B = 0.25$
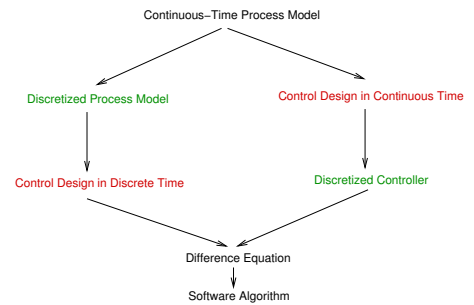
Time dependence in sampled-data systems

Sample and Hold Approximation

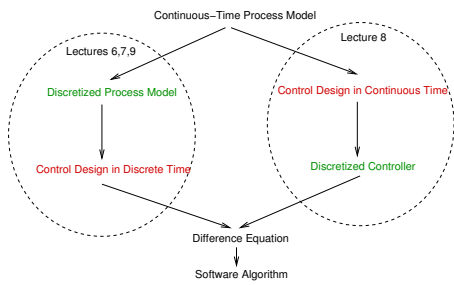A sampler in direct combination with a ZOH device gives an average delay of $h/2$
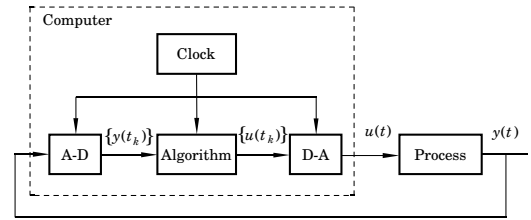
Design Approaches

Design Approaches

Continuous–Time Process Model

Lectures 6,7,9

Lecture 8

Discretized Process Model

Control Design in Continuous Time

Control Design in Discrete Time

Discretized Controller

Difference Equation

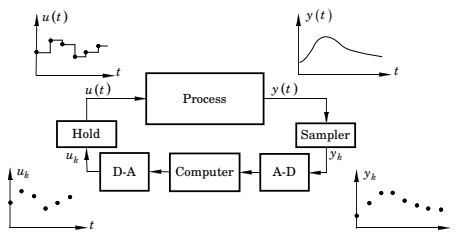Software Algorithm

---

Sampled Control Theory

The basic idea: Look at the sampling instances only

- the stroboscopic model
- look upon the process from the computer's point of view

Sampled Control Theory

- System theory analogous to continuous time linear systems
- Better performance can be achieved
- Problems with intersample behaviour

Disk Drive Example

Control of the arm of a disk drive

$$G(s) = \frac{k}{Js^2}$$

Continuous time controller

$$U(s) = \frac{bK}{a}U_c(s) - K\frac{s+b}{s+a}Y(s)$$

Discrete time controller (continuous time design + discretization)
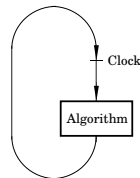
$$u(t_k) = K(\frac{b}{a}u_c(t_k) - y(t_k) + x(t_k))$$
$$x(t_k + h) = x(t_k) + h\left((a-b)y(t_k) - ax(t_k)\right)$$
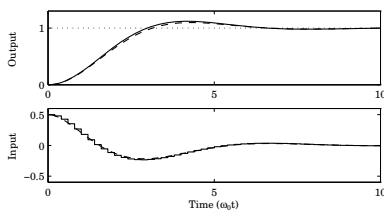

Disk Drive Example

```
y: = adin(in2)
u:=K*(b/a*uc-y+x)
dout(u)
x:=x+h*((a-b)*y-a*x)
```
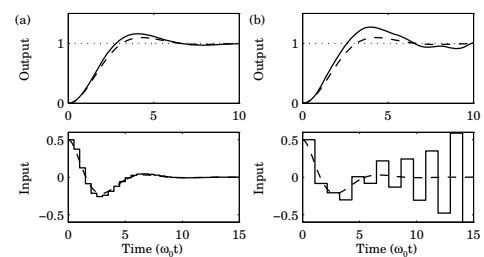

Sampling period $h = 0.2/\omega_0$

Increased sampling period
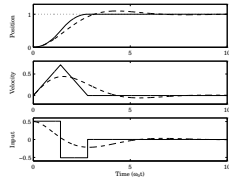
a) $h = 0.5/\omega_0$ b) $h = 1.08/\omega_0$

Better performance?

Dead-beat control $h = 1.4/\omega_0$

$$u(t_k) = t_0 u_c(t_k) + t_1 u_c(t_{k-1}) - s_0 y(t_k) - s_1 y(t_{k-1}) - r_1 u(t_{k-1})$$
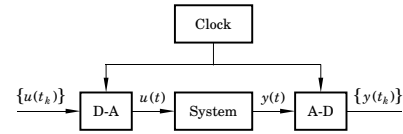

However, long sampling periods also have problems

- open loop between the samples
- disturbance and reference changes that occur between samples will remain undetected until the next sample

Sampling of Systems

Look at the system from the point of view of the computer

Zero-order-hold sampling of a system

- Let the inputs be piecewise constant
- Look at the sampling points only

Solving the continuous-time system equation

In the basic control course you learned how to calculate the solution to the state-space system equation.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

is given by

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau$$

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)} Bu(\tau) d\tau + Du(t)$$

Use this to derive a discretized model!!

Sampling a continuous-time system

System description

$$\frac{dx}{dt} = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Solve the system equation under the assumption that u is piecewise constant

$$\begin{aligned} x(t) &= e^{A(t-t_k)}x(t_k) + \int_{t_k}^t e^{A(t-s')} Bu(s') ds' \\ &= e^{A(t-t_k)}x(t_k) + \int_{t_k}^t e^{A(t-s')} ds' \, Bu(t_k) \ \ (u \text{ const.}) \\ &= e^{A(t-t_k)}x(t_k) + \int_0^{t-t_k} e^{As} ds \, Bu(t_k) \ \ (\text{variable change } s = t - s') \\ &= \Phi(t, t_k)x(t_k) + \Gamma(t, t_k)u(t_k) \end{aligned}$$

The General Case

$$x(t_{k+1}) = \Phi(t_{k+1}, t_k)x(t_k) + \Gamma(t_{k+1}, t_k)u(t_k)$$
$$y(t_k) = Cx(t_k) + Du(t_k)$$

where

$$\Phi(t_{k+1}, t_k) = e^{A(t_{k+1}-t_k)}$$
$$\Gamma(t_{k+1}, t_k) = \int_0^{t_{k+1}-t_k} e^{As} ds \ B$$

No assumption about periodic sampling

Periodic sampling

Assume periodic sampling, i.e. $t_k = k \cdot h$, then

$$x(kh + h) = \Phi x(kh) + \Gamma u(kh)$$
$$y(kh) = Cx(kh) + Du(kh)$$

where

$$\Phi = e^{Ah}$$
$$\Gamma = \int_0^h e^{As} ds \ B$$

NOTE: Time-invariant linear system!

No approximations

Example: Sampling of double integrator

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$
$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} x$$

Hence

$$\Phi = e^{Ah} = I + Ah + A^2 h^2/2 + \cdots$$
$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & h \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$$

$$\Gamma = \int_0^h \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} ds = \int_0^h \begin{pmatrix} s \\ 1 \end{pmatrix} ds = \begin{pmatrix} \frac{h^2}{2} \\ h \end{pmatrix}$$

Calculating the Matrix Exponential

Pen and paper for small systems (see Exercises)

Matlab for large systems

```
>> syms h

>> a = [0 1; 0 0];

>> phi = expm(a*h)

ans =

[ 1, h]
[ 0, 1]
```

Calculating the Matrix Exponential

One can show that

$$\begin{pmatrix} \Phi(h) & \Gamma(h) \\ 0 & I \end{pmatrix} = \exp\left(\begin{pmatrix} A & B \\ 0 & 0 \end{pmatrix} h \right)$$
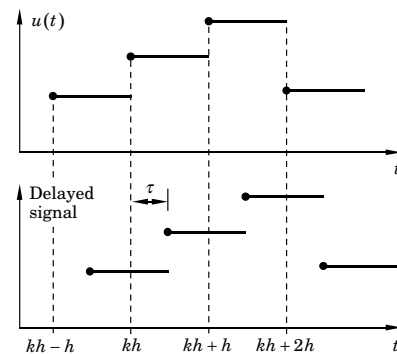
Simultaneous calculation of Φ and Γ

```
>> syms h
>> a = [0 1;0 0];
>> b = [0 1];
>> expm([a b;zeros(1,size(a)) 0]*h)

ans =

[   1,   h, 1/2*h^2]
[   0,   1,       h]
[   0,   0,       1]
```

Sampling of system with time delay

Sampling of system with time delay

Input delay $0 < \tau < h$

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t - \tau)$$

$$x(kh + h) - \Phi x(kh) = \int_{kh}^{kh+h} e^{A(kh+h-s')} B u(s' - \tau) ds'$$

$$= \underbrace{\int_{kh}^{kh+\tau} e^{A(kh+h-s')} ds' B}_{\Gamma_1} u(kh - h) + \underbrace{\int_{kh+\tau}^{kh+h} e^{A(kh+h-s')} ds' B}_{\Gamma_0} u(kh)$$

$$= \underbrace{e^{A(h-\tau)} \int_0^\tau e^{As} ds B}_{\Gamma_1} u(kh - h) + \underbrace{\int_0^{h-\tau} e^{As} ds B}_{\Gamma_0} u(kh)$$

$$x(kh + h) = \Phi x(kh) + \Gamma_1 u(kh - h) + \Gamma_0 u(kh)$$

Introduce a new state $z(kh) = u(kh - h)$

State-space form

$$\begin{pmatrix} x(kh + h) \\ z(kh + h) \end{pmatrix} = \begin{pmatrix} \Phi & \Gamma_1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x(kh) \\ z(kh) \end{pmatrix} + \begin{pmatrix} \Gamma_0 \\ I \end{pmatrix} u(kh)$$

This approach can be extended also for $\tau > h$

Additional states corresponding to delayed input signals are added

Similar techniques can also be used to handle output delays and delays that are internal in the plant.

In continuous-time delays mean infinite-dimensional systems. In discrete-time the sampled system is a finite-dimensional system $\Rightarrow$ easier to handle

Example – Double integrator with delay τ

$$\Phi = e^{Ah} = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$$

$$\Gamma_1 = e^{A(h-\tau)} \int_0^\tau e^{As}\, ds\, B = \begin{pmatrix} 1 & h-\tau \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tau^2/2 \\ \tau \end{pmatrix} = \begin{pmatrix} h\tau - \tau^2/2 \\ \tau \end{pmatrix}$$

$$\Gamma_0 = \int_0^{h-\tau} e^{As}\, ds\, B = \begin{pmatrix} (h-\tau)^2/2 \\ h-\tau \end{pmatrix}$$

$$x(kh+h) = \Phi x(kh) + \Gamma_1 u(kh-h) + \Gamma_0 u(kh)$$

Explicit solution of system equations

$$x(1) = \Phi x(0) + \Gamma u(0)$$
$$x(2) = \Phi x(1) + \Gamma u(1)$$
$$\quad = \Phi^2 x(0) + \Phi\Gamma u(0) + \Gamma u(1)$$
$$\vdots$$
$$x(k) = \Phi^k x(0) + \sum_{j=0}^{k-1} \Phi^{k-j-1}\Gamma u(j)$$
$$y(k) = C\Phi^k x(0) + \sum_{j=0}^{k-1} C\Phi^{k-j-1}\Gamma u(j) + Du(k)$$

Two parts, one depending on the initial condition $x(0)$ and one that is a weighted sum of the inputs over the interval $[0, k-1]$

Stability

The matrix Φ can, if it has distinct eigenvalues, be written on the form

$$\Phi = U \begin{bmatrix} \lambda_1 & & * \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} U^{-1}. \quad \text{Hence } \Phi^k = U \begin{bmatrix} \lambda_1{}^k & & * \\ & \ddots & \\ 0 & & \lambda_n{}^k \end{bmatrix} U^{-1}.$$

The diagonal elements are the eigenvalues of Φ.

Φ^k decays exponentially if and only if $|\lambda_k| < 1$ for all k, i.e. if the eigenvalues of Φ are inside the unit circle.

This is the stability condition for discrete-time systems

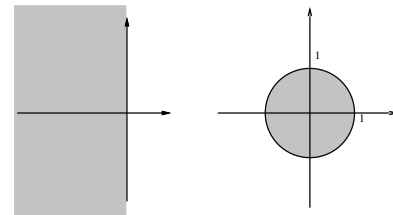Eigenvalues obtained from the characteristic equation

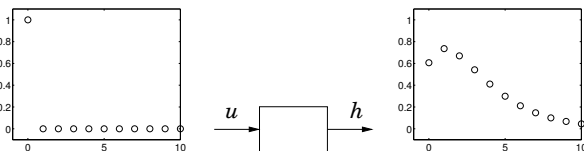$$\det(\lambda I - \Phi) = 0$$

Stability regions

In continuous time the stability region is the complex left half plane, i.e., the system is asymptotically stable if all the poles are strictly in the left half plane.

In discrete time the stability region is the unit circle.

Pulse response

$$x(1) = \Gamma \qquad x(2) = \Phi\Gamma \qquad x(3) = \Phi^2\Gamma \qquad \dots$$
$$h(1) = C\Gamma \qquad h(2) = C\Phi\Gamma \qquad y(3) = C\Phi^2\Gamma \qquad \dots$$

$$h(0) = D \qquad h(k) = C\Phi^{k-1}\Gamma \qquad k = 1, 2, 3, \dots$$

(Continuous-time (assuming $D = 0$): $h(t) = Ce^{At}B \quad t \geq 0$)

Convolution

Swedish: Faltning

Continuous time:

$$(h * u)(t) = \int_0^t h(t-s)u(s)\,ds \qquad t \geq 0$$

Discrete time:

$$(h * u)(k) = \sum_{j=0}^k h(k-j)u(j) \qquad k = 0, 1, \dots$$

Solution to the System Equation

The solution to the system equation

$$y(k) = C\Phi^k x(0) + \sum_{j=0}^{k-1} C\Phi^{k-j-1}\Gamma u(j) + Du(k)$$

can be written in terms of the pulse response

$$y(k) = C\Phi^k x(0) + (h * u)(k)$$

Two parts, one that depends on the initial conditions and one that is a convolution between the pulse response and the input signal

Difference equation

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) \quad = \quad b_1 u(k-1) + \cdots + b_n u(k-n)$$

Differential equation

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n y \quad = \quad b_1 \frac{d^{n-1} u}{dt^{n-1}} + \cdots + b_n u$$

From difference equation to state space

Start with $b_1 = 1$ and $b_2 = \cdots = b_n = 0$. Put $k \to k+1$, and $y(k) = z(k)$:

$$z(k+1) + a_1 z(k) + \cdots + a_n z(k-n+1) = u(k)$$
$$x(k) = [\, z(k) \quad z(k-1) \quad \dots \quad z(k-n+1) \,]^T$$

gives

$$x(k+1) = \begin{bmatrix} z(k+1) \\ z(k) \\ \vdots \\ z(k-n+2) \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_n \\ 1 & 0 & \dots & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k)$$
$$z(k) = [\, 1 \quad 0 \quad \dots \quad 0 \,] x(k)$$

The general case: Controllable Canonical Form

Let

$$y(k) = b_1 z(k) + b_2 z(k-1) + \cdots + b_n z(k-n)$$

Then (think superposition!)

$$x(k+1) = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_n \\ 1 & 0 & \dots & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k)$$
$$y(k) = [\, b_1 \quad b_2 \quad \dots \quad b_n \,] x(k)$$

and

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) \quad = \quad b_1 u(k-1) + \cdots + b_n u(k-n)$$

By choosing different states, different state-space models can be derived which all describe the same difference equation.

A realization is minimal if the number of states is equal to n.

In the *direct form* the states are selected as the old values of y together with the old values of u – non-minimal.

Change of coordinates

The coordinate transformation $z = Tx$ gives

$$z(k+1) = Tx(k+1)$$
$$= T(\Phi x(k) + \Gamma u(k))$$
$$= T\Phi T^{-1} z(k) + T\Gamma u(k)$$

$$y(k) = CT^{-1} z(k) + Du(k)$$

The eigenvalues of $T\Phi T^{-1}$ are the same as for Φ.

(Proof:

$$\det(\lambda I - T\Phi T^{-1}) = \det(\lambda TT^{-1} - T\Phi T^{-1}) = \det T \det(\lambda I - \Phi) \det T^{-1}$$
$$= \det(\lambda I - \Phi)$$

)

In Matlab

```
>> a = [0 1;0 0];
>> b = [0;1];
>> c = [1 0];
>> d = 0;
>> contSystem = ss(a,b,c,d);
>> h = 0.1;
>> discSystem = c2d(contSystem,h);
>> impulse(discSystem);
>> step(discSystem);
```

43