Sampling of Linear Systems

Real-Time Systems, Lecture 6

Anton Cervin

2 February 2016

Lund University, Department of Automatic Control

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

2

Textbook

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

Wittenmark, Åström, Årzén: *IFAC Professional Brief: Computer Control: An Overview*, (Educational Version 2016) ("IFAC PB")
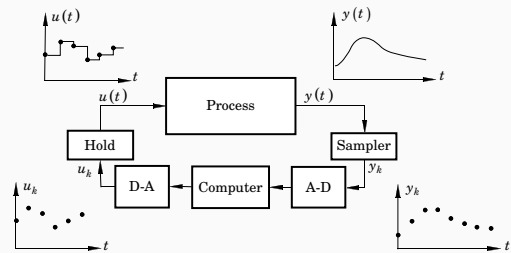
- Summary of the digital control parts of Åström and Wittenmark: *Computer Controlled Systems* (1997)
- Some new material

Chapters 10 and 11 are not part of this course (but can be useful in other courses, e.g., Predictive Control)

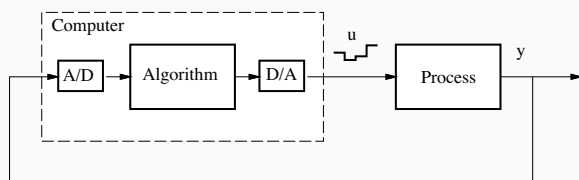Chapters 7, 13 and 14 partly overlap with RTCS.
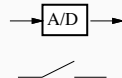
3

Sampled Control Theory

- System theory analogous to continuous-time linear systems
- Better control performance can be achieved (compared to discretization of continuous-time design)
- Problems with aliasing, intersample behaviour

4

Sampling

AD-converter acts as sampler

Regular/periodic sampling:

- Constant sampling interval h
- Sampling instants: $t_k = kh$

5

Hold Devices

Zero-Order Hold (ZOH) almost always used. DA-converter acts as hold device $\Rightarrow$ piecewise constant control signals
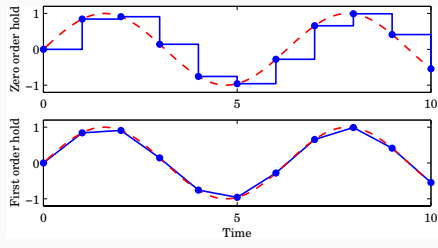
First-Order Hold (FOH):

- 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:
  - Better reconstruction
  - Continuous output signal
- Disadvantages:
  - $f(kh + h)$ must be available at time kh
  - More involved controller design
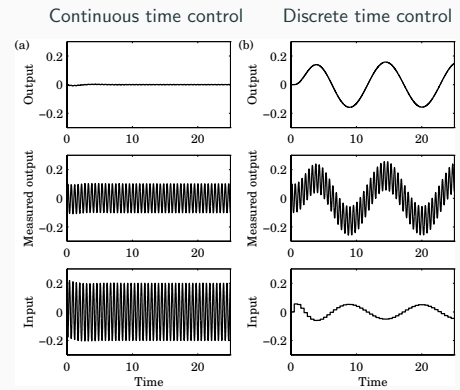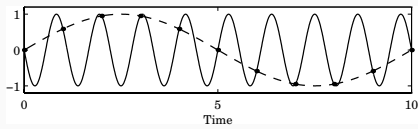 - Not supported by standard DA-converters

6

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 of high-frequency measurement noise may create new frequencies!

Aliasing

- Sampling frequency [rad/s]: $\omega_s = 2\pi/h$
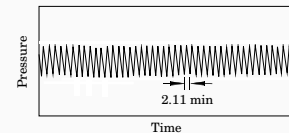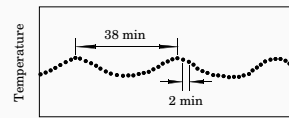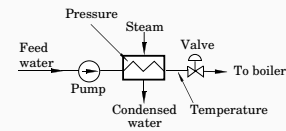- Nyquist frequency [rad/s]: $\omega_N = \omega_s/2$

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

Calculation of "fundamental alias" for an original frequency ω_1:

$$\omega = |(\omega_1 + \omega_N) \bmod (\omega_s) - \omega_N|$$

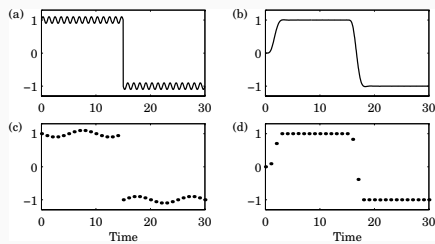## Aliasing – Real World Example

Feed water heating in a ship boiler

Prefiltering

Analog low-pass filter needed to remove high-frequency measurement noise before sampling
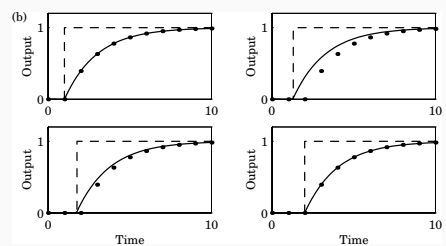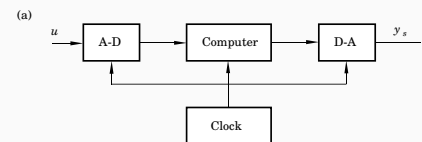
Example:

(a), (c): $f_1 = 0.9$ Hz, $f_N = 0.5$ Hz $\Rightarrow f_{alias} = 0.1$ Hz

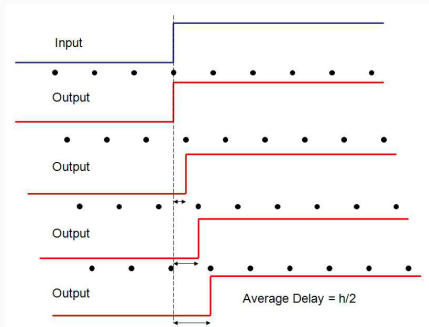(b), (d): 6th order Bessel prefilter with bandwidth $f_B = 0.25$ Hz

More on aliasing in Lecture 11.

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 for Computer Control
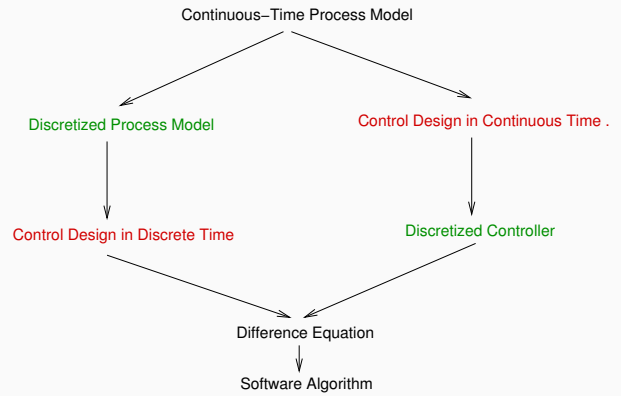
Design Approaches for Computer Control
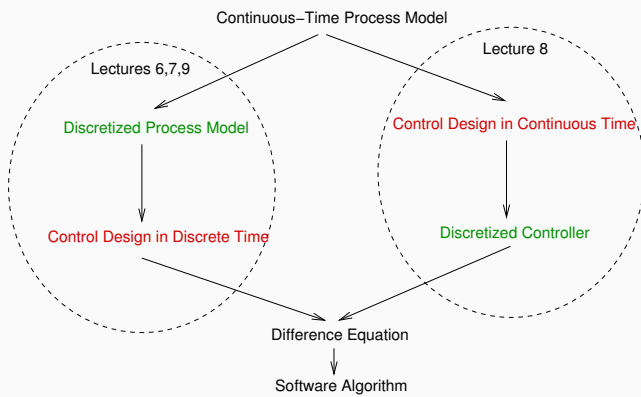
Sampled Control Theory
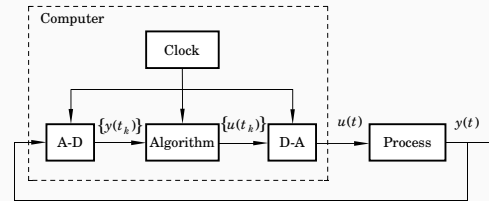
Basic idea: Look at the sampling instances only

- Stroboscopic model
- Look upon the process from the computer's point of view

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)

$$u(t_k) = K\left(\frac{b}{a}u_c(t_k) - y(t_k) + x(t_k)\right)$$

$$x(t_{k+1}) = x(t_k) + h\Big((a-b)y(t_k) - ax(t_k)\Big)$$

(Continuous-time poles placed according to $P(s) = s^3 + 2\omega_0 s^2 + 2\omega_0^2 s + \omega_0^3$)

Disk Drive Example

```
uc := adin(1)
y  := adin(2)
u  := K*(b/a*uc-y+x)
daout(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?

Deadbeat control, $h = 1.4/\omega_0$

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

Better Performance?

Deadbeat: The output reaches the reference value after n samples (n = model order)

No counterpart in continuous time

However, long sampling periods also have problems

- Open loop between samples
- Sensitive to model errors
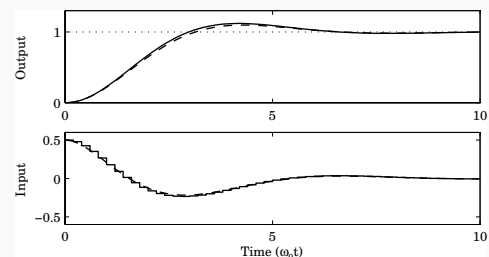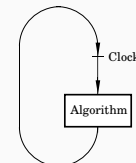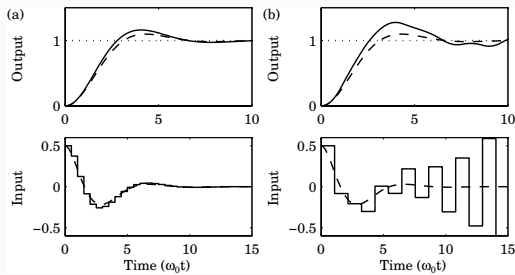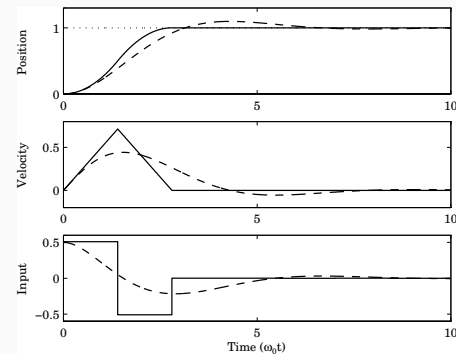- Disturbance and reference changes that occur between samples will remain undetected until the next sample

Sampling of Linear Systems

Look at the system from the point of view of the computer
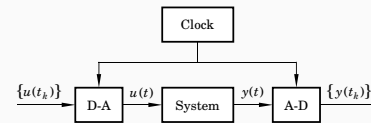
Zero-order-hold sampling

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

Continuous-Time System Model

Linear time-invariant system model in continuous time:

$$\begin{cases} \dfrac{dx}{dt} = Ax + Bu \\[2mm] y = Cx + Du \end{cases}$$

Solution (see basic course in control):

$$x(t) = e^{At} x(t_0) + \int_{t_0}^t e^{A(t-\tau)} Bu(\tau) ds$$

$$y(t) = Ce^{At} x(t_0) + C \int_{t_0}^t e^{A(t-\tau)} Bu(\tau) ds + Du(t)$$

Use this to derive a discrete-time model

Sampling a Continuous-Time System

Solve the system equation

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

from time t_k to time t under the assumption that u is piecewise constant (ZOH sampling)

$$\begin{aligned} x(t) &= e^{A(t-t_k)} x(t_k) + \int_{t_k}^t e^{A(t-s')} Bu(s')\,ds' \\ &= e^{A(t-t_k)} x(t_k) + \int_{t_k}^t e^{A(t-s')}\,ds'\,Bu(t_k) \quad (Bu(t_k) \text{ const.}) \\ &= e^{A(t-t_k)} x(t_k) + \int_{t-t_k}^0 -e^{As}\,ds\,Bu(t_k) \quad (\text{var. change } s = t - s') \\ &= e^{A(t-t_k)} x(t_k) + \int_0^{t-t_k} e^{As}\,ds\,Bu(t_k) \quad (\text{change int. limits}) \\ &= \Phi(t, t_k) x(t_k) + \Gamma(t, t_k) u(t_k) \end{aligned}$$

The General Case

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

where

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

No assumption about periodic sampling

Periodic Sampling

Assume periodic sampling, i.e. $t_k = kh$. Then

$$
\begin{aligned}
x(kh + h) &= \Phi x(kh) + \Gamma u(kh) \\
y(kh) &= Cx(kh) + Du(kh)
\end{aligned}
$$

where

$$
\begin{aligned}
\Phi &= e^{Ah} \\
\Gamma &= \int_0^h e^{As} ds \, B
\end{aligned}
$$

NOTE: Time-invariant linear system!

No approximations

Example: Sampling of Double Integrator

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

Periodic sampling with interval h:

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

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

Calculating the Matrix Exponential

Pen and paper for small systems

$$
\Phi = \mathcal{L}^{-1} \left(sI - A \right)^{-1}
$$

Matlab for large systems (numeric or symbolic calculations)

```
>> syms h

>> A = [0 1; 0 0];

>> expm(A*h)

ans =

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

Calculating the Matrix Exponential
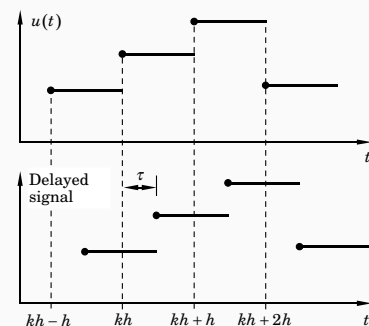
One can show that

$$
\begin{pmatrix} \Phi & \Gamma \\ 0 & I \end{pmatrix} = \exp\left(\begin{pmatrix} A & B \\ 0 & 0 \end{pmatrix} h \right)
$$

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 $\tau \leq h$ (assumed to be constant)

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

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

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

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

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

Sampling of System with Time Delay

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

Sampled system in state-space form

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

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

- $h < \tau \leq 2h \Rightarrow$ two extra state variables, etc.

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 $\tau \leq h$

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

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

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

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

Solution of the Discrete System Equation

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

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 in the form

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

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

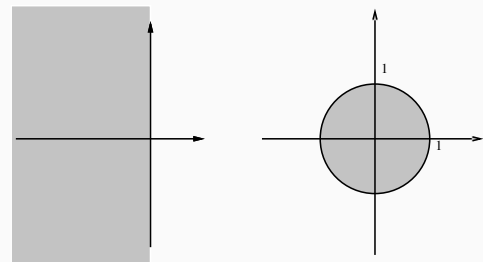Eigenvalues obtained from the characteristic equation

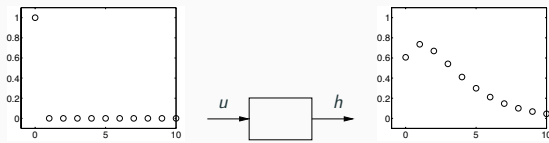$$\det(\lambda I - \Phi) = 0$$

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

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

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

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

Convolution

Swedish: Faltning

Continuous time:

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

Discrete time:

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

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 Equations

Difference equation of order n:

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

Differential equation of order n:

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

From Difference Equation to State Space Form

Start with $b_1 = 1$ and $b_2 = \cdots = b_n = 0$ in difference equation above

Put $k \to k + 1$, and $y(k) = z(k)$:

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

gives

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

$$z(k) = [1 \quad 0 \quad \ldots \quad 0] x(k)$$

Controllable Canonical Form

Let

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

Then (think superposition!)

$$x(k+1) = \begin{bmatrix} -a_1 & -a_2 & \ldots & -a_n \\ 1 & 0 & \ldots & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k)$$

$$y(k) = [b_1 \quad b_2 \quad \ldots \quad b_n] x(k)$$

which corresponds to

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

State-Space Realizations

By choosing different state variables, different state-space models can be derived which all describe the same input–output relation

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.

Some realizations have better numerical properties than others, see Lecture 11.

43

In Matlab

```
>> A = [0 1;0 0]
>> B = [0;1]
>> C = [1 0]
>> D = 0
>> contsys = ss(A,B,C,D)
>> h = 0.1
>> discsys = c2d(contsys,h)  % ZOH sampling
>> pole(discsys)
>> impulse(discsys)
>> step(discsys)
```

44