Modelling, analysis and control of linear systems using state space representations

O. Sename

1Gipsalab, CNRS-INPG, FRANCE
Olivier.Sename@inpg.fr
www.lag.ensieg.inpg.fr/sename

Approche Etat pour la commande / IEG- SEM
Outline

Introduction

Modelling of dynamical systems

Properties

Discrete-time systems

State feedback control

Observer

Integral Control

A polynomial approach

Further in discrete-time control

Conclusion
References

Some interesting books:

Objective of any control system:

**Nominal stability (NS):** The system is stable with the nominal model (no model uncertainty)

**Nominal Performance (NP):** The system satisfies the performance specifications with the nominal model (no model uncertainty)

**Robust stability (RS):** The system is stable for all perturbed plants about the nominal model, up to the worst-case model uncertainty (including the real plant)

**Robust performance (RP):** The system satisfies the performance specifications for all perturbed plants about the nominal model, up to the worst-case model uncertainty (including the real plant).
Recall of the "control design" process:

- Plant study and modelling
- Determination of sensors and actuators (measured and controlled outputs, control inputs)
- Performance specifications
- Control design (many methods)
- Simulation tests
- Implementation, tests and validation
State space representations (SEM)

O. Sename

Introduction

Modelling of dynamical systems

Properties

Discrete-time systems

State feedback control

Observer

Integral Control

A polynomial approach

Further in discrete-time control

Conclusion

Different issues for modelling:

Identification based method:

▸ System excitations using PRBS (Pseudo Random Binary Signal) or sinusoïdal signals
▸ Determination of a transfer function reproducing the input/output system behavior

Knowledge-based method:

▸ Represent the system behavior using differential and/or algebraic equations, based on physical knowledge.
▸ Formulate a nonlinear state-space model, i.e. a matrix differential equation of order 1.
▸ Determine the steady-state operating point about which to linearize.
▸ Introduce deviation variables and linearize the model.
Why state space equations?

- Dynamical systems where physical equations can be derived: electrical engineering, mechanical engineering, aerospace engineering, microsystems, process plants, etc.
- Include physical parameters: easy to use when parameters are changed for design.
- State variables have physical meaning.
- Easy to extend to Multi-Input Multi-Output (MIMO) systems.
- Advanced control design methods are based on state space equations (reliable numerical optimisation tools).
Some physical examples
Many dynamical systems can be represented by Ordinary Differential Equations (ODE) as

\[
\begin{align*}
\dot{x}(t) &= f((x(t), u(t), t), \quad x(0) = x_0 \\
y(t) &= g((x(t), u(t), t) \quad (1)
\end{align*}
\]

where \( f \) and \( g \) are non linear functions.
Example: Inverted pendulum

It is described by:

Parameters:

- $y(t)$ - distance from some reference point
- $\theta(t)$ - angle of pendulum
- $M$ - mass of cart
- $m$ - mass of pendulum (assumed concentrated at tip)
- $l$ - length of pendulum
- $f(t)$ - forces applied to pendulum
Example: Inverted pendulum

The dynamical equations are as follows:

\[
\ddot{y} = \frac{1}{\lambda_m + \sin^2 \theta(t)} \left[ \frac{f(t)}{m} + \dot{\theta}^2(t) \ell \sin \theta(t) - g \cos \theta(t) \sin \theta(t) \right]
\]

\[
\ddot{\theta} = \frac{1}{\ell \lambda_m + \sin^2 \theta(t)} \left[ -\frac{f(t)}{m} \cos \theta(t) + \dot{\theta}^2(t) \ell \sin \theta(t) \cos \theta(t) + (1 - \lambda_m)g \sin \theta(t) \right]
\]

where \( \lambda_m = (M/m) \)
Example: Lateral vehicle model

The dynamical equations are as follows:

**NL Horizontal chassis dynamics**

\[
\begin{align*}
\dot{a}_x &= \dot{v}_x - \psi v_y \\
\dot{a}_y &= \dot{v}_y + \psi \dot{v}_x \\
ma_y &= 2F_{tyf} \cos(\delta_f) + 2F_{tyr} + F_{yd} \\
I_z \ddot{\psi} &= 2F_{tyf} \cos(\delta_f) l_f = 2F_{tyr} l_r + M_{yd}
\end{align*}
\]

**Assumption**

\[F_{tyf} \cos(\delta_f) >> F_{txf} \sin(\delta_f)\]

**NL Lateral Tire Forces**

\[
\begin{align*}
\alpha_f &= \arctan \left( \frac{v_y + l_f \dot{\psi}}{v_x} \right) - \delta_f \\
\alpha_r &= \arctan \left( \frac{v_y - l_r \dot{\psi}}{v_x} \right)
\end{align*}
\]
A **continuous-time** LINEAR state space system is given as :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \\ y(t) = Cx(t) + Du(t) \end{cases} \tag{2}$$

where $x(t) \in \mathbb{R}^n$ is the system state (vector of state variables), $u(t) \in \mathbb{R}^m$ the control input and $y(t) \in \mathbb{R}^p$ the measured output. $A$, $B$, $C$ and $D$ are real matrices of appropriate dimensions. $x_0$ is the initial condition. $n$ is the order of the state space representation.

**Matlab :** `ss(A,B,C,D)` creates a SS object `SYS` representing a continuous-time state-space model.
A first example: DC Motor

The dynamical equations are:

\[ Ri + L \frac{di}{dt} + e = u \quad e = K_e \omega \]
\[ J \frac{d\omega}{dt} = -f \omega + \Gamma_m \quad \Gamma_m = K_c i \]

System of 2 equations of order 1 \( \Rightarrow \) 2 state variables. A possible choice \( x = \begin{pmatrix} \omega \\ i \end{pmatrix} \)

It gives:

\[
A = \begin{pmatrix}
-\frac{f}{J} & \frac{K_c}{J} \\
-\frac{K_e}{L} & -\frac{R}{L}
\end{pmatrix}
\]
\[
B = \begin{pmatrix}
0 \\
\frac{1}{L}
\end{pmatrix}
\]
\[
C = \begin{pmatrix}
0 \\ 1
\end{pmatrix}
\]

Extension: measurement= motor angular position
Example: Wind turbine

State space representations (SEM)

O. Senaye

Introduction

Modelling of dynamical systems

Properties

Discrete-time systems

State feedback control

Observer

Integral Control

A polynomial approach

Further in discrete-time control

Conclusion
A linearisation within 2 regions gives

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) + Ed(t) \\
y(t) &= Cx(t)
\end{align*}
\]

with

\[
A = \begin{pmatrix}
\frac{\gamma - C_d}{I_{rot}} & -1 & \frac{C_d}{I_{rot}} \\
K_d & 0 & -K_d \\
\frac{C_d}{I_{gen}} & \frac{1}{I_{gen}} & \frac{C_d}{I_{gen}}
\end{pmatrix}, 
B = \begin{pmatrix}
0 \\
0 \\
-\frac{1}{I_{gen}}
\end{pmatrix}, 
E = \begin{pmatrix}
\frac{\alpha}{I_{rot}} \\
0 \\
0
\end{pmatrix},
\]

and 
\[
C = \begin{pmatrix}
0 & 0 & 1
\end{pmatrix}
\]

\(x_1 = \text{rotor-speed} \quad x_2 = \text{drive-train torsion spring force} \quad x_3 = \text{rotational generator speed}\)

\(u = \text{generator torque} \quad d : \text{wind speed}\)

\(I_{rot} : \text{rotor rotational inertia} \quad I_{gen} : \text{generator rotational inertia} \quad K_d : \text{spring constant} \quad C_d : \text{torsional damping constant} \quad \alpha : \text{partial derivative of rotor aerodynamic torque with respect blade pitch angle} \quad \gamma : \text{partial derivative of rotor aerodynamic torque with respect to rotor speed}\)
Examples: Suspension

Let the following mass-spring-damper system.

where $x_1$ is the relative position, $M_1$ the system mass, $k_1$ the spring coefficient, $u$ the force generated by the active damper, and $F_1$ is an external disturbance. Applying the mechanical equations it leads:

$$M_1 \ddot{x}_1 = -k_1 x_1 + u + F_1$$  \hspace{1cm} (3)
Examples: Suspension cont.

The choice \( x = \begin{pmatrix} x_1 \\ \dot{x}_1 \end{pmatrix} \) gives

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) + Ed(t) \\
y(t) &= Cx(t)
\end{align*}
\]

where \( d = F_1 \), \( y = x_1 \) with

\[
A = \begin{pmatrix} 0 & 1 \\ -k_1/M_1 & 0 \end{pmatrix}, \quad B = E = \begin{pmatrix} 0 \\ 1/M_1 \end{pmatrix}, \quad \text{and} \quad C = \begin{pmatrix} 0 & 1 \end{pmatrix}
\]
Exercice

Let the following quarter car model with active suspension. \( Z_{caisse} \) and \( Z_{roue} \) are the relative position of the chassis and of the tire, \( m_c \) (resp. \( m_r \)) the mass of the chassis (resp. of the tire), \( k \) (resp. \( k_p \)) the spring coefficient of the suspension (of the tire), \( u \) the active damper force, \( Z_{sol} \) is the road profile.

Choose some state variables and give a state space representation of this system.
Linearisation
Equilibrium point

An equilibrium point satisfies:

\[ 0 = f(x_{eq}(t), u_{eq}(t), t) \]  \hspace{1cm} (4)

For the pendulum, we can choose \( y = \theta = f = 0 \).
Linearisation Method (1)

The linearisation can be done around an equilibrium point or around a particular point.

Thus consider

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \\
y(t) &= g(x(t), u(t))
\end{align*}
\]

Say that \(\{x_Q(t), u_Q(t), y_Q(t); t \in \mathbb{R}\}\) is a given set of trajectories that satisfy the above equations, i.e.

\[
\begin{align*}
\dot{x}_Q(t) &= f(x_Q(t), u_Q(t)); \quad x_Q(t_0) \text{ given} \\
y_Q(t) &= g(x_Q(t), u_Q(t))
\end{align*}
\]

\[
\begin{align*}
\dot{x}(t) &\approx f(x_Q, u_Q) + \left. \frac{\partial f}{\partial x} \right|_{x=x_Q, u=u_Q} (x(t) - x_Q) + \left. \frac{\partial f}{\partial u} \right|_{x=x_Q, u=u_Q} (u(t) - u_Q) \\
y(t) &\approx g(x_Q, u_Q) + \left. \frac{\partial g}{\partial x} \right|_{x=x_Q, u=u_Q} (x(t) - x_Q) + \left. \frac{\partial g}{\partial u} \right|_{x=x_Q, u=u_Q} (u(t) - u_Q)
\end{align*}
\]
Linearisation Method (2)

This leads to a linear state space representation of the system, around the equilibrium point.
Defining $\tilde{x} = x - x_{eq}$, $\tilde{u} = u - u_{eq}$ and $\tilde{y} = y - y_{eq}$ we get

$$\begin{align*}
\dot{\tilde{x}}(t) &= A\tilde{x}(t) + B\tilde{u}(t), \\
\tilde{y}(t) &= C\tilde{x}(t) + D\tilde{u}(t)
\end{align*}$$

(5)

with $A = \frac{\partial f}{\partial x} \bigg|_{x=x_{eq},u=u_{eq}}$, $B = \frac{\partial f}{\partial u} \bigg|_{x=x_{eq},u=u_{eq}}$, $C = \frac{\partial g}{\partial x} \bigg|_{x=x_{eq},u=u_{eq}}$ and $D = \frac{\partial g}{\partial u} \bigg|_{x=x_{eq},u=u_{eq}}$
Example: Inverted pendulum (2)

Applying the linearisation method leads to:

This is a linear state space model in which A, B and C are:

\[
A = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & \frac{-mg}{M} & 0 \\
0 & 0 & \frac{(M+m)g}{M\ell} & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}; \quad B = \begin{bmatrix}
0 \\
\frac{1}{M} \\
0 \\
-\frac{1}{M\ell}
\end{bmatrix}; \quad C = [1 \ 0 \ 0 \ 0]
\]
Linear systems: transfer function
Equivalence transfer function - state space representation

Consider a linear system given by:

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t), \quad x(0) = x_0 \\
y(t) &= Cx(t) + Du(t)
\end{align*}
\]  

(6)

Using the Laplace transform (and assuming zero initial condition \(x_0 = 0\)), (6) becomes:

\[
s \cdot x(s) = Ax(s) + Bu(s) \Rightarrow (sI_n - A)x(s) = Bu(s)
\]

Then the transfer function matrix of system (6) is given by

\[
G(s) = C(sI_n - A)^{-1}B + D = \frac{N(s)}{D(s)}
\]  

(7)

**Matlab:** if SYS is an SS object, then \(\text{tf}(\text{SYS})\) gives the associated transfer matrix. Equivalent to \(\text{tf}(N, D)\)
Conversion TF to SS

There mainly three cases to be considered

Simple numerator

$$\frac{y}{u} = G(s) = \frac{1}{s^3 + a_1 s^2 + a_2 s + a_3}$$

Numerator order less than denominator order

$$\frac{y}{u} = G(s) = \frac{b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3} = \frac{N(s)}{D(s)}$$

Numerator equal to denominator order

$$\frac{y}{u} = G(s) = \frac{b_0 s^3 + b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3} = \frac{N(s)}{D(s)}$$
Canonical forms
Canonical forms

Some specific state space representations are well-known and often used as the so-called controllable canonical form. It corresponds to the transfer function:

\[ G(s) = \frac{c_0 + c_1 s + \ldots + c_{n-1} s^{n-1}}{a_0 + a_1 s + \ldots + a_{n-1} s^{n-1} + s^n} \]

In Matlab, use `canon`
Modal form

Let us consider a transfer function as:

\[ G(s) = \frac{b_1}{s-a_1} + \frac{b_2}{s-a_2} + \ldots + \frac{b_n}{s-a_n} \]

- Define a set of transfer functions:
  \[ \frac{X_i(s)}{U(s)} = \frac{b_i}{s-a_i} \Rightarrow \dot{x}_i = a_ix_ib_iu_i \]

- This gives

\[
\begin{cases}
\dot{x}(t) = Ax(t) + Bu(t), & x(0) = x_0 \\
y(t) = Cx(t) + Du(t)
\end{cases}
\] (8)

with \( A = \begin{bmatrix} a_1 & 0 & \ldots & 0 \\ 0 & a_2 & 0 & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & a_n \end{bmatrix} \), \( B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \) and

\[ C = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}. \]
Solution of state space linear systems
Solution of state space equations - continuous case

The state \( x(t) \), solution of \( \dot{x}(t) = Ax(t) \), with initial condition \( x(0) = x_0 \) is given by

\[
x(t) = e^{At} x(0)
\]  

(9)

This requires to compute \( e^{At} \). There exist 3 methods to compute \( e^{At} \):

1. Inverse Laplace transform of \((sI_n - A)^{-1}\):
2. Diagonalisation of \( A \)
3. Cayley-Hamilton method
Complete state solution

The state $x(t)$, solution of system (6), is given by

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

(10)

In **Matlab**: use `expm` and not `exp`.

**Simulation of state space systems**

Use `lsim`.

Example:

```matlab
  t = 0:0.01:5; u = sin(t); lsim(sys,u,t)
```
Non unicity
Non unicity

Given a transfer function, there exists an infinity of state space representations (equivalent in terms of input-output behavior). Let

\[
\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t), \\
y(t) &= Cx(t) + Du(t)
\end{aligned}
\]  

(11)

the transfer matrix being \( G(s) = C(sl_n - A)^{-1}B + D \), and consider the change of variables \( x = Tz \) (\( T \) being an invertible matrix). Replacing \( x = Tz \) in the previous system gives:

\[
\begin{aligned}
T\dot{z}(t) &= ATz(t) + Bu(t) \\
y(t) &= CTz(t) + Du(t)
\end{aligned}
\]  

(12, 13)

Hence

\[
\begin{aligned}
\dot{z}(t) &= T^{-1}ATz(t) + T^{-1}Bu(t) \\
y(t) &= CTz(t) + Du(t)
\end{aligned}
\]  

(14, 15)
Defining $\tilde{A} = T^{-1}AT$, $\tilde{B} = T^{-1}B$ and $\tilde{C} = CT$, the transfer function of the previous system is:

$$\tilde{G}(s) = \tilde{C}(sl_n - \tilde{A})^{-1}\tilde{B} + D$$ \hspace{1cm} (16)

$$= CT\left(sI_n - T^{-1}AT\right)^{-1}T^{-1}B + D$$ \hspace{1cm} (17)

$$= CT T^{-1} T^{-1} (sI_n - A)^{-1} T T^{-1} B + D = G(s)$$ \hspace{1cm} (19)

Using $I_n = T^{-1}T$, we get
Stability
Stability

Definition
An equilibrium point $x_{eq}$ is stable if, for all $\rho > 0$, there exists a $\eta > 0$ such that:

$$\|x(0) - x_{eq}\| < \eta \implies \|x(t) - x_{eq}\| < \rho, \forall t \geq 0$$

Definition
An equilibrium point $x_{eq}$ is asymptotically stable if it is stable and, there exists $\eta > 0$ such that:

$$\|x(0) - x_{eq}\| < \eta \implies x(t) \to x_{eq}, \text{ when } t \to \infty$$

These notions are equivalent for linear systems (not for non linear ones).
Stability Analysis

The stability of a linear state space system is analyzed through the characteristic equation \( \det(sl_n - A) = 0 \).
The system poles are then the eigenvalues of the matrix \( A \).
It then follows:

**Proposition**

A system \( \dot{x}(t) = Ax(t) \), with initial condition \( x(0) = x_0 \), is stable if \( \text{Re}(\lambda_i) < 0 \), \( \forall i \), where \( \lambda_i \), \( \forall i \), are the eigenvalues of \( A \).

Using **Matlab**, if SYS is an SS object then `pole(SYS)` computes the poles P of the LTI model SYS. It is equivalent to compute `eig(A)`.
Stability Analysis - Lyapunov

The stability of a linear state space system can be analysed through the Lyapunov theory. It is the basis of all extension of stability for non linear systems, time-delay systems, time-varying systems ...

Theorem

A system \( \dot{x}(t) = Ax(t) \), with initial condition \( x(0) = x_0 \), is asymptotically stable at \( x = 0 \) if and only if there exist some matrices \( P = P^T > 0 \) and \( Q > 0 \) such that:

\[
A^T P + PA = -Q
\]

(20)

see `lyap` in MATLAB.

Proof: The Lyapunov theory says that a linear system is stable if there exists a continuous function \( V(x) \) s.t.:

\[
V(x) > 0 \text{ with } V(0) = 0 \quad \text{and} \quad \dot{V}(x) = \frac{dV}{dx} \geq 0
\]

A possible Lyapunov function for the above system is:
About zeros

- Roots of the transfer function numerator are called the system zeros.
- Need to develop a similar way of defining/computing them using a state space model.
- Zero: is a generalized frequency $\alpha$ for which the system can have a non-zero input $u(t) = u_0 e^{\alpha t}$, but exactly zero output $y(t) = 0$.
- The zeros are found by solving:

$$\begin{bmatrix} A - \lambda I_n & B \\ C & D \end{bmatrix} = 0 \quad (21)$$

In Matlab use `zero`

Example: find the zero of: $\frac{s+3}{s^2+5s+2}$
Controllability
Controllability

Controllability refers to the ability of controlling a state-space model using state feedback.

Definition

*Given two states* \( x_0 \) *and* \( x_1 \), *the system (6) is controllable if there exist* \( t_1 > 0 \) *and a piecewise-continuous control input* \( u(t), \ t \in [0, t_1] \), *such that* \( x(t) \) *takes the values* \( x_0 \) *for* \( t = 0 \) *and* \( x_1 \) *for* \( t = t_1 \).
Proposition

The controllability matrix is defined by
\[ C = [B, A.B, \ldots, A^{n-1}.B]. \] Then system (6) is controllable if and only if \( \text{rank}(C) = n \).

If the system is single-input single output (SISO), it is equivalent to \( \det(C) \neq 0 \).

Using \texttt{Matlab}, if SYS is an SS object then \texttt{crtb(SYS)} returns the controllability matrix of the state-space model SYS with realization (A,B,C,D). This is equivalent to \texttt{ctrb(sys.a,sys.b)}.
Exercises

Test the controllability of the previous examples: DC motor, suspension, inverted pendulum.
Observability
Observability

Observability refers to the ability to estimate a state variable.

Definition

A linear system (2) is completely observable if, given the control and the output over the interval $t_0 \leq t \leq T$, one can determine any initial state $x(t_0)$.

It is equivalent to characterize the non-observability as:

A state $x(t)$ is not observable if the corresponding output vanishes, i.e. if the following holds:

$$y(t) = \dot{y}(t) = \ddot{y}(t) = \ldots = 0$$
Where does observability come from?

Compare the transfer function of the two different systems*

\[ \dot{x} = -x + u \]
\[ y = 2x \]

and

\[ \dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u \]
\[ y = \begin{bmatrix} 2 & 0 \end{bmatrix} x \]
Proposition

The observability matrix is defined by

\[
\mathcal{O} = \begin{bmatrix}
C \\
CA \\
\vdots \\
CA^{n-1}
\end{bmatrix}.
\]

Then system (6) is observable if and only if \( \text{rank}(\mathcal{O}) = n \).

If the system is single-input single output (SISO), it is equivalent to \( \det(\mathcal{O}) \neq 0 \).

Using \texttt{Matlab}, if SYS is an SS object then \texttt{obsv(SYS)} returns the observability matrix of the state-space model SYS with realization \((A,B,C,D)\). This is equivalent to \texttt{OBSV(sys.a,sys.c)}. 

Observability cont.
Exercices

Test the observability of the previous examples: DC motor, suspension, inverted pendulum. Analysis of different cases, according to the considered number of sensors.
Minimality
Minimality

Definition

A state space representation of a linear system (2) of order $n$ is said to be minimal if it is controllable and observable. In this case, the corresponding transfer function $G(s)$ is of minimal order $n$, i.e. is irreducible (no cancellation of poles and zeros). When the transfer function is not of minimal order, there exists non controllable or non observable modes.
Kalman decomposition
Kalman decomposition

When the linear system (2) is not completely controllable or observable, it can be decomposed as shown. Use `ctrbf` and `obsvf` in Matlab.
Toward digital control

Digital control

Usually controllers are implemented in a digital computer as:

This requires the use of the discrete theory.

\(\uparrow\) (Sampling theory + Z-Transform) \(\uparrow\)
Z-Transform
Definitions

Mathematical definition
Because the output of the ideal sampler, \( x^*(t) \), is a series of impulses with values \( x(kT_e) \), we have:

\[
x^*(t) = \sum_{k=0}^{\infty} x(kT_e) \delta(t - kT_e)
\]

by using the Laplace transform,

\[
\mathcal{L}[x^*(t)] = \sum_{k=0}^{\infty} x(kT_e) e^{-ksT_e}
\]

Noting \( z = e^{sT_e} \), we can derive the so called Z-Transform

\[
X(z) = Z[x(k)] = \sum_{k=0}^{\infty} x(k)z^{-k}
\]
Properties

Definition

\[ X(z) = Z[x(k)] = \sum_{k=0}^{\infty} x(k)z^{-k} \]

Properties

\[
\begin{align*}
Z[\alpha x(k) + \beta y(k)] &= \alpha X(z) + \beta Y(z) \\
Z[x(k - n)] &= z^{-n} Z[x(k)] \\
Z[kx(k)] &= -z \frac{d}{dz} Z[x(k)] \\
Z[x(k) \ast y(k)] &= X(z) \cdot Y(z) \\
\lim_{k \to \infty} x(k) &= \lim_{1 \to z^{-1}} (z - 1)X(z)
\end{align*}
\]

The \( z^{-1} \) can be interpreted as a pure delay operator.
Exercise

Determine the Z-Transform of the step function (1) and of the ramp function (2)

\[ x_{\text{step}}(k) = \begin{cases} 1 & \text{if } k \geq 0 \\ 0 & \text{if } k < 0 \end{cases} \]

\[ x_{\text{ramp}}(k) = \begin{cases} k & \text{if } k \geq 0 \\ 0 & \text{if } k < 0 \end{cases} \]
Exercise

Determine the Z-Transform of the step function (1) and of the ramp function (2)

\[ x_{\text{step}}(k) = \begin{cases} 1 & \text{if } k \geq 0 \\ 0 & \text{if } k < 0 \end{cases} \quad \quad x_{\text{ramp}}(k) = \begin{cases} k & \text{if } k \geq 0 \\ 0 & \text{if } k < 0 \end{cases} \]

Solution

1) Step

\[ X_{\text{step}}(z) = 1 + z^{-1} + z^{-2} + \cdots = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1} \]

2) Ramp (note that \( x_{\text{ramp}}(k) = k x_{\text{step}}(k) \))

\[ X_{\text{ramp}}(z) = -z \frac{d}{dz} \left( \frac{z}{z - 1} \right) = \frac{z}{(z - 1)^2} \]
Zero order holder

Sampler and Zero order holder

A sampler is a switch that close every $T_e$ seconds. A Zero order holder holds the signal $x$ for $T_e$ seconds to get $h$ as:

$$h(t + kT_e) = x(kT_e), \quad 0 \leq t < T_e$$
Zero order holder (cont’d)

Model of the Zero order holder
The transfer function of the zero-order holder is given by:

\[
G_{BOZ}(s) = \frac{1}{s} - \frac{e^{-sT_e}}{s}
\]

\[
= \frac{1 - e^{-sT_e}}{s}
\]

Influence of the D/A and A/D
Note that the precision is also limited by the available precision of the converters (either A/D or D/A). This error is also called the amplitude quantization error.
Representation of the discrete linear systems

The discrete output of a system can be expressed as:

\[ y(k) = \sum_{n=0}^{\infty} h(k - n)u(n) \]

hence, applying the Z-transform leads to

\[ Y(z) = Z[h(k)]U(z) = H(z)U(z) \]

\[ H(z) = \frac{b_0 + b_1 z + \cdots + b_m z^m}{a_0 + a_1 z + \cdots + a_n z^n} = \frac{Y}{U} \]

where \( n \geq m \) is the order of the system

Corresponding difference equation:

\[ y(k) = \frac{1}{a_n} \left[ b_0 u(k - n) + b_1 u(k - n + 1) + \cdots + b_m u(k - n + m) \right] - a_0 y(k - n) - a_2 y(k - n + 1) - \cdots - a_{n-1} y(k - 1) \]
Some useful transformations

<table>
<thead>
<tr>
<th>$x(t)$</th>
<th>$X(s)$</th>
<th>$X(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta(t)$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\delta(t - kT_e)$</td>
<td>$e^{-ksT_e}$</td>
<td>$z^{-k}$</td>
</tr>
<tr>
<td>$u(t)$</td>
<td>$\frac{1}{s}$</td>
<td>$\frac{z}{z-1}$</td>
</tr>
<tr>
<td>$t$</td>
<td>$\frac{1}{s^2}$</td>
<td>$\frac{z}{zT_e}$</td>
</tr>
<tr>
<td>$e^{-at}$</td>
<td>$\frac{1}{s+a}$</td>
<td>$\frac{z}{(z-1)^2}$</td>
</tr>
<tr>
<td>$1 - e^{-at}$</td>
<td>$\frac{1}{s(s+a)}$</td>
<td>$\frac{z-e^{-aT_e}}{z(1-e^{-aT_e})}$</td>
</tr>
<tr>
<td>$\sin(\omega t)$</td>
<td>$\frac{\omega}{s^2+\omega^2}$</td>
<td>$\frac{z\sin(\omega T_e)}{z(z-e^{-aT_e})}$</td>
</tr>
<tr>
<td>$\cos(\omega t)$</td>
<td>$\frac{s}{s^2+\omega^2}$</td>
<td>$\frac{z(z-\cos(\omega T_e))}{z^2-2z\cos(\omega T_e)+1}$</td>
</tr>
</tbody>
</table>

**Exercise**

Discretize (sampling time $T_e$) the system described by the Laplace function (using a Zero order holder):

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1}{s(s+1)}$$
Exercise

Discretize the system described by the Laplace function (using a Zero order holder):

\[ H(s) = \frac{Y(s)}{U(s)} = \frac{1}{s(s+1)} \]

Adding the Zero order holder leads to:

\[ G_{BOZ}(s)H(s) = \frac{1 - e^{-sT_e}}{s} \frac{1}{s(s+1)} \]
\[ = \frac{1 - e^{-sT_e}}{s^2(s+1)} \]
\[ = (1 - e^{-sT_e})\left(\frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1}\right) \]

hence

\[ Z[G_{BOZ}(s)H(s)] = (1 - z^{-1})Z\left[\frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1}\right] \]
Exercise (cont’d)

\[ Z[G_{BOZ}(s)H(s)] = (1 - z^{-1})Z\left[ \frac{1}{s^2} - \frac{1}{s} + \frac{1}{s+1} \right] \]

\[ = (1 - z^{-1})\left[ \frac{z T_e}{(z-1)^2} - \frac{z}{z-1} + \frac{z}{z - e^{-T_e}} \right] \]

\[ = \frac{(ze^{-T_e} - z + z T_e) + (1 - e^{-T_e} - T_e e^{-T_e})}{(z - 1)(z - e^{-T_e})} \]

if \( T_e = 1 \), we have

\[ Z[G_{BOZ}(s)H(s)] = \frac{ze^{-T_e} - z + z T_e + (1 - e^{-T_e} - T_e e^{-T_e})}{(z - 1)(z - e^{-T_e})} \]

\[ = \frac{ze^{-1} + 1 - 2e^{-1}}{(z - 1)(z - e^{-1})} \]

\[ = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0} \]
Exercise (cont’d)

Let us return back to sampled-time domain

\[
\frac{Y(z)}{U(z)} = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0}
\]

\[
Y(z) = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0} U(z)
\]

\[
Y(z)(z^2 + a_1 z + a_0) = (b_1 z + b_0) U(z)
\]

\[
y(n+2) + a_1 y(n+1) + a_0 y(n) = b_1 u(n+1) + b_0 u(n)
\]

With an unit feedback, the closed loop function is given by:

\[
F_{cl}(z) = \frac{G(z)}{1 + G(z)}
\]
Poles, Zeros and Stability
Equivalence \( \{s\} \leftrightarrow \{z\} \)

\( \{s\} \rightarrow \{z\} \)

The equivalence between the Laplace domain and the Z domain is obtained by the following transformation:

\[
z = e^{sT_e}
\]

Two poles with an imaginary part that differs by \(2\pi/T_e\) give the same pole in Z.

Stability domain
Approximations

Forward difference (Rectangle inferior)

\[ s = \frac{z - 1}{T_e} \]

Backward difference (Rectangle superior)

\[ s = \frac{z - 1}{z T_e} \]
Approximations (cont’d)

Trapezoidal difference (Tustin)

\[ S = \frac{2z - 1}{T_e z + 1} \]
Systems definition

A **discrete-time** state space system is as follows:

\[
\begin{align*}
    x((k+1)h) &= A_d x(kh) + B_d u(kh), \quad x(0) = x_0 \\
    y(kh) &= C_d x(kh) + D_d u(kh)
\end{align*}
\]  

(22)

where \( h \) is the sampling period.

**Matlab**: \texttt{ss}(\( A_d, B_d, C_d, D_d, h \)) creates a SS object SYS representing a discrete-time state-space model.
Relation with transfer function

For discrete-time systems,

\[
\begin{align*}
    x((k+1)h) &= A_d x(kh) + B_d u(kh), \quad x(0) = x_0 \\
    y(kh) &= C_d x(kh) + D_d u(kh)
\end{align*}
\]  

(23)

the discrete transfer function is given by

\[
G(z) = C_d (zI_n - A_d)^{-1} B_d + D_d
\]

(24)

where \( z \) is the shift operator, i.e. \( zx(kh) = x((k+1)h) \)
### Recall Laplace & Z-transform

### From Transfer Function to State Space

<table>
<thead>
<tr>
<th>$H(s)$ to state space</th>
<th>$H(z)$ to state space</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\dot{X}}{Y} = den(s)$</td>
<td>$\frac{\dot{X}}{Y} = den(z)$</td>
</tr>
<tr>
<td>$\frac{X}{Y} = num(s)$</td>
<td>$\frac{X}{Y} = num(z)$</td>
</tr>
<tr>
<td>$\dot{X} = AX + BU$</td>
<td>$X_{k+1} = FX_k + GU_k$</td>
</tr>
<tr>
<td>$Y = CX + DU$</td>
<td>$Y_k = CX_k + DU_k$</td>
</tr>
</tbody>
</table>

$$Y(s) = \left[C[sI - A]^{-1}B + D\right] U(s)$$

$$Y(z) = \left[C[zI - F]^{-1}G + D\right] U(z)$$
Solution of state space equations - discrete case

The state $x_k$, solution of system $x_{k+1} = A_d x_k$ with initial condition $x_0$, is given by

$$x_1 = A_d x_0 \quad (25)$$
$$x_2 = A_d^2 x_0 \quad (26)$$
$$x_n = A_d^n x_0 \quad (27)$$

The state $x_k$, solution of system (22), is given by

$$x_1 = A_d x_0 + B_d u_0 \quad (28)$$
$$x_2 = A_d^2 x_0 + A_d B_d u_0 + B_d u_1 \quad (29)$$
$$x_n = A_d^n x_0 + \sum_{i=0}^{n-1} A_d^{n-1-i} B_d u_i \quad (30)$$
State space analysis (discrete-time systems)

Stability

A system (state space representation) is stable iff all the eigenvalues of the matrix $F$ are inside the unit circle.

Controllability definition

Definition

Given two states $x_0$ and $x_1$, the system (22) is controllable if there exist $K_1 > 0$ and a sequence of control samples $u_0, u_1, \ldots, u_{K_1}$, such that $x_k$ takes the values $x_0$ for $k = 0$ and $x_1$ for $k = K_1$.

Observability definition

Definition

The system (22) is said to be completely observable if every initial state $x(0)$ can be determined from the observation of $y(k)$ over a finite number of sampling periods.
State space analysis (2)

Controllability
The system is controllable iff
\[ \mathcal{C}_{(A_d, B_d)} = \text{rg} \begin{bmatrix} B_d & A_d B_d & \ldots & A_d^{n-1} B_d \end{bmatrix} = n \]

Observability
The system is observable iff
\[ \mathcal{O}_{(A_d, C_d)} = \text{rg} \begin{bmatrix} C_d & C_d A_d & \ldots & C_d A_d^{n-1} \end{bmatrix}^T = n \]

Duality
Observability of \((C_d, A_d)\) ⇔ Controllability of \((A_d^T, C_d^T)\).
(proof...)
Controllability of \((A_d, B_d)\) ⇔ Observability of \((B_d^T, A_d^T)\).
(proof...)

State space representations (SEM)
O.Sename
Introduction
Modelling of dynamical systems
Properties
Discrete-time systems
State feedback control
Observer
Integral Control
A polynomial approach
Further in discrete-time control
Conclusion
A state feedback controller for a continuous-time system is:

\[ u(t) = -Fx(t) \]  \hspace{1cm} (31)

where \( F \) is a \( m \times n \) real matrix.

When the system is SISO, it corresponds to:

\[ u(t) = -f_1x_1 - f_2x_2 - \ldots - f_nx_n \]

with \( F = [f_1, f_2, \ldots, f_n] \).

When the system is MIMO we have:

\[
\begin{bmatrix}
  \vdots \\
  u_1 \\
  u_2 \\
  \vdots \\
  u_m
\end{bmatrix} =
\begin{bmatrix}
  f_{11} & \cdots & f_{1n} \\
  \vdots & \ddots & \vdots \\
  f_{m1} & \cdots & f_{mn}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{bmatrix}
\]
State feedback (2)

Using state feedback controllers (31), we get in closed-loop (for simplicity $D = 0$)

\[
\begin{align*}
\dot{x}(t) &= (A - BF)x(t), \\
y(t) &= Cx(t)
\end{align*}
\]

and the stability (and dynamics) of the closed-loop system is then given by the eigenvalues of $A - BF$. For discrete-time system we get:

\[
\begin{align*}
x(k + 1) &= (A - BF)x(k), \\
y(k) &= Cx(k)
\end{align*}
\]
State feedback (3)

When the objective is to track some reference signal $r$, the state feedback control is of the form:

$$ u(t) = -Fx(t) + Gr(t) \quad (34) $$

or

$$ u(k) = -Fx(k) + Gr(k) \quad (35) $$

$G$ is a $m \times p$ real matrix. Then the closed-loop transfer matrix is:

$$ G_{CL}(s) = C(sI_n - A + BF)^{-1}BG \quad (36) $$

$G$ is chosen to ensure a unitary steady-state gain as:

$$ G = [C(-A + BF)^{-1}B]^{-1} \quad (37) $$

★ For discrete-time system:

$$ G = [C(I_n - A + BF)^{-1}B]^{-1} \quad (38) $$
Pole placement control

Proposition

Let a linear system given by $A, B$, and let $\gamma_i$, $i = 1, ..., n$, a set of complex elements (i.e. the desired poles of the closed-loop system). There exists a state feedback control $u = -Fx$ such that the poles of the closed-loop system are $\gamma_i$, $i = 1, ..., n$ if and only if the pair $(A, B)$ is controllable.
Why state feedback and not output feedback?

Example: \( G(s) = \frac{1}{s^2 - s} \)
Consider the canonical form.

**Case of output feedback:** \( u = -Ly \)
Then \( \dot{x}(t) = (A - BL)C \dot{x}(t) \)
For the example, the characteristic polynomial is \( P_{BF}(s) = s^2 - s - L \). The closed-loop system cannot be stabilized.

**Case of state feedback:** \( u = -Fx \)
Let \( F = [f_1, f_2] \). Then \( P_{BF}(s) = s^2 + (-1 + f_2)s + f_1 \)
So we can choose any \( F \). For instance \( f_1 = 1, f_2 = 3 \) gives \( P_{BF}(s) = (s + 1)^2 \)
Pole placement control (1)

First case: controllable canonical form

\[ A = \begin{bmatrix}
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \vdots & 0 & 1 \\
-a_0 & -a_1 & \ldots & \ldots & -a_{n-1}
\end{bmatrix}, \quad B = \begin{bmatrix}
0 \\
\vdots \\
0
\end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix}
c_0 & c_1 & \ldots & c_{n-1}
\end{bmatrix}.

Let \( F = \begin{bmatrix} f_1 & f_2 & \ldots & f_n \end{bmatrix} \)

Then

\[ A - BF = \begin{bmatrix}
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \vdots & 0 & 1 \\
-a_0 - f_1 & -a_1 - f_2 & \ldots & \ldots & -a_{n-1} - f_n
\end{bmatrix} \quad \text{(39)} \]
Consider the desired closed-loop polynomial:
\[(s - \gamma_1)(s - \gamma_2)\ldots(s - \gamma_n) = s^n + \alpha_{n-1}s^{n-1} + \ldots + \alpha_1 s + \alpha_0\]
The solution:
\[f_i = -a_{i-1} + \alpha_{i-1}, \quad i = 1, \ldots, n\]
ensures that the poles of \(A - BF\) are \(\{\gamma_i\}, \quad i = 1, n\)
When we consider a general state space representation, it is first necessary to use a change of basis to make the system under canonical form.

Use \texttt{F=acker}(A, B, P) where \(P\) is the set of desired closed-loop poles.
Pole placement control (3)

Procedure for the general case:

1. Check controllability of \((A, B)\)
2. Calculate \(\mathcal{C} = \begin{bmatrix} B, AB, \ldots, A^{n-1}B \end{bmatrix}\).

\[
\mathcal{C}^{-1} = \begin{bmatrix} q_1 & \vdots & q_n \end{bmatrix}
\]

Define \(T = \begin{bmatrix} q_n & q_nA & \vdots & q_nA^{n-1} \end{bmatrix}^{-1}\).

3. Note \(\tilde{A} = T^{-1}AT\) and \(\tilde{B} = T^{-1}B\) (which are under the controllable canonical form)

4. Choose the desired closed-loop poles and define the desired closed-loop characteristic polynomial:

\[s^n + \alpha_{n-1}s^{n-1} + \ldots + \alpha_1 s + \alpha_0\]

5. Calculate the state feedback \(u = -\bar{F}x\) with:

\[\bar{f}_i = -a_{i-1} + \alpha_{i-1}, \quad i = 1, \ldots, n\]

6. Calculate (for the original system):

\[u = -Fx, \text{ with } F = \bar{F} T^{-1}\]
Problem: To implement a state feedback control, the measurement of all the state variables is necessary. If this is not available, we will use a state estimation through a so-called Observer.
Observer form:

\[
\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) - L(C\hat{x}(t) - y(t))
\]

\[\hat{x}_0\]  

where \(\hat{x}(t) \in \mathbb{R}^n\) is the estimated state of \(x(t)\) and \(L\) is the \(n \times p\) constant observer gain matrix to be designed.
Observer

The estimated error, \( e(t) := x(t) - \hat{x}(t) \), satisfies:

\[
\dot{e}(t) = (A - LC)e(t)
\]  \hspace{1cm} (41)

If \( L \) is designed such that \( A - LC \) is stable, then \( \hat{x}(t) \) converges asymptotically towards \( x(t) \).

**Proposition**

(40) is an observer for system (2) if and only if the pair \((C, A)\) is observable, i.e.

\[
\text{rank}(\mathcal{O}) = n
\]

where

\[
\mathcal{O} = \begin{bmatrix}
C \\
CA \\
\vdots \\
CA^{n-1}
\end{bmatrix}.
\]
Observer design

The observer design is restricted to find $L$ such that $A - LC$ is stable. This is still a pole placement problem. In order to use the `acker` Matlab function, we will use the duality property between observability and controllability, i.e.:

$$(C, A) \text{ observable} \Leftrightarrow (A^T, C^T) \text{ controllable}.$$  

Then there exists $L^T$ such that the eigenvalues of $A^T - C^T L^T$ can be randomly chosen. As $(A - LC)^T = A^T - C^T L^T$ then $L$ exists such that $A - LC$ is stable.

Matlab: use $L = \text{acker}(A', C', Po)'$ where $Po$ is the set of desired observer poles.

Remark: usually the observer poles are chosen around 5 to 10 times higher than the closed-loop system, so that the state estimation is good as early as possible. This is quite important to avoid that the observer makes the closed-loop system slower.
Observer-based control

When an observer is built, we will use as control law:

\[ u(t) = -F\hat{x}(t) + Gr(t) \]  \hspace{1cm} (42)

We then need to study the stability of the complete closed-loop system, using the extended state:

\[ x_e(t) = \begin{bmatrix} x(t) & e(t) \end{bmatrix}^T \]

The closed-loop system with observer (40) and control (42) is:

\[ \dot{x}_e(t) = \begin{bmatrix} A-BF & BF \\ 0 & A-LC \end{bmatrix} x_e(t) + \begin{bmatrix} BG \\ 0 \end{bmatrix} r(t) \]  \hspace{1cm} (43)
Separation principle

The characteristic polynomial of the extended system is:

\[ \text{det}(sl_n - A + BF) \times \text{det}(sl_n - A + LC) \]

If the observer and the control are designed separately then the closed-loop system with the dynamic measurement feedback is stable, given that the control and observer systems are stable and the eigenvalues of (43) can be obtained directly from them. This corresponds to the so-called separation principle.
When the linear system (2) is not completely controllable or observable, it is then important to study the stability of the non controllable and non observable modes. Use `ctrbf` and `obsvf` Matlab commands.
Integral Control

A state feedback controller may not allow to reject the effects of disturbances (particularly of input disturbances). A very useful method consists in adding an integral term to ensure a unitary static closed-loop gain.

Considered system:

\[
\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) + Ed(t), \quad x(0) = x_0 \\
y(t) &= Cx(t)
\end{aligned}
\]  

where \( d \) is the disturbance.

The objective is to keep \( y \) close to a reference signal \( r \), even in the presence of \( d \), i.e to keep \( r - y \) asymptotically stable.
Integral Control

The method consists in extending the system by adding a new state variable:

$$\dot{z}(t) = r(t) - y(t)$$

and to use a new state feedback:

$$u(t) = -Fx(t) - Hz(t)$$

We get

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A - BF & BH \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) + \begin{bmatrix} E \\ 0 \end{bmatrix} d(t)$$
Integral control scheme

The complete structure has the following form:

When an observer is to be used, the control action simply becomes:

$$u(t) = -F\hat{x}(t) - Hz(t)$$
Equivalence RST controller and observer-based

The use of an observer-based controller is equivalent to the following controller:

\[ u(s) = -F\left(sI_n - A + BF + LC\right)^{-1}Ly(s) \]
\[ + \left[I_n - F\left(sI_n - A + BF + LC\right)^{-1}B\right]Gr(s) \]

which corresponds to a two-degrees of freedom controller

\[ u(s) = -\frac{R(s)}{S(s)}y(s) + \frac{T(s)}{S(s)}r(s) \]

and this can be implemented in an RST form.
Impose a maximal time response to a discrete system is equivalent to place the poles inside a circle defined by the upper bound of the bound given by this time response. The more the poles are close to zero, the more the system is fast.
Frequency analysis

As in the continuous time, the Bode diagram can also be used.
Example with sampling Time
\[ T_e = 1 \text{s} \Leftrightarrow f_e = 1 \text{Hz} \Leftrightarrow w_e = 2\pi \]:

Note that, in our case, the Bode is cut at the pulse \( w = \pi \).
see \texttt{SYSD = \texttt{c2d}(	exttt{SYSC, Ts, METHOD})} in \texttt{MATLAB}. 
Frequency analysis

As in the continuous time, the Bode diagram can also be used.

Example with sampling Time

\( T_e = 1 \text{s} \iff f_e = 1 \text{Hz} \iff w_e = 2\pi \):

![Discrete/Continuous frequency response](image)

Note that, in our case, the Bode is cut at the pulse \( w = \pi \).

see \( \text{SYSD} = \text{c2d}(\text{SYSC}, \text{Ts}, \text{METHOD}) \) in MATLAB.

Sampling \( \leftrightarrow \) Limitations

Recall the Shannon theorem that impose the sampling frequency at least 2 times higher that the system.
About sampling period and robustness

Influence of the sampling period on the poles

In theory, smaller the sampling period $T_e$ is, closer the discrete system is from the continuous one.

But reducing the sampling time modify poles location... Poles and zeros become closer to the limit of the unit circle ⇒ can introduce instability (decrease robustness).

⇒ Sampling influences stability and robustness
⇒ Over sampling increase noise sensitivity
Zeros

Influence of the sampling period on the poles
A discrete system with one or few zeros at the origin is faster than one with no origin zeros. In the time domain a zero at the origin induces a sample advance.
Stability

Recall

A linear continuous feedback control system is stable if all poles of the closed-loop transfer function $T(s)$ lie in the left half s-plane.

The Z-plane is related to the S-plane by $z = e^{-sT_e} = e^{(\sigma + j\omega)T_e}$. Hence

$$|z| = e^{\sigma T_e} \text{ and } \angle z = \omega T_e$$
Stability (cont’d)

Jury criteria

The denominator polynomial 
\( \text{den}(z) = a_0 z^n + a_1 z^{n-1} + \cdots + a_n = 0 \) has all its roots inside the unit circle if all the first coefficients of the odd row are positive.

\[
\begin{array}{c|cccccccc}
1 & a_0 & a_1 & a_2 & \ldots & a_{n-k} & \ldots & a_n \\
2 & a_n & a_{n-1} & a_{n-2} & \ldots & a_k & \ldots & a_0 \\
3 & b_0 & b_1 & b_2 & \ldots & b_{n-1} \\
2 & b_{n-1} & b_{n-2} & b_{n-3} & \ldots & b_0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
2n+1 & s_0 & & & & & & \\
\end{array}
\]

\[
b_0 = a_0 - a_n \frac{a_n}{a_0}
\]

\[
b_1 = a_1 - a_{n-1} \frac{a_n}{a_0}
\]

\[
b_k = a_k - a_{n-k} \frac{a_n}{a_0}
\]

\[
c_k = b_k - b_{n-1-k} \frac{b_{n-1}}{b_0}
\]
Example

Stability

Find the stability region of $D(z) = z^2 + a_1 z + a_2$
### Example

**Stability**

Find the stability region of \( D(z) = z^2 + a_1 z + a_2 \)

**Solution**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1 &gt; 0</td>
<td>( a_1 )</td>
<td>( a_2 )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( a_2 )</td>
<td></td>
<td>( a_1 )</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( 1 - a_2^2 ) &gt; 0?</td>
<td>( a_1 - a_1 a_2 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( a_1 - a_1 a_2 )</td>
<td>1 - ( a_2^2 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( \frac{(1-a_2^2)^2-(a_1^2(1-a_2^2)^2)}{1-a_2^2} ) &gt; 0?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

hence,

\[
1 - a_2^2 > 0 \\
(1 + a_2)^2 > a_1^2
\]
How to get a discrete controller

First way

► Obtain a discrete-time plant model (by discretization)
► Design a discrete-time controller
► Derive the difference equation

Second way

► Design a continuous-time controller
► Converse the continuous-time controller to discrete time (c2d)
► Derive the difference equation

Now the question is how to implement the computed controller on a real-time (embedded) system, and what are the precautions to take before?
Implementation characteristics
Anti-aliasing & Sampling

Anti-aliasing
Practically it is smart to use a constant high sampling frequency with an analog filter matching this frequency. Then, after the A/D converter, the signal is down-sampled to the frequency used by the controller. Remember that the pre-filter introduce phase shift.

Sampling frequency choice
The sampling time for discrete-time control are based on the desired speed of the closed loop system. A rule of thumb is that one should sample 4 – 10 times per rise time $T_r$ of the closed loop system.

$$N_{\text{sample}} = \frac{T_r}{T_e} \approx 4 - 10$$

where $T_e$ is the sampling period, and $N_{\text{sample}}$ the number of samples.
Problematic
Sampled theory assume presence of clock that synchronizes all measurements and control signal. Hence in a computer based control there always is delays (control delay, computational delay, I/O latency).

Origins
There are several reasons for delay apparition
- Execution time (code)
- Preemption from higher order process
- Interrupt
- Communication delay
- Data dependencies

Hence the control delay is not constant. The delay introduce a phase shift ⇒ Instability!
Delay (cont’d)

Admissible delay (Bode)

- Measure the phase margin: \( PM = 180 + \varphi_{w_0} [\text{º}] \), where \( \varphi_{w_0} \) is the phase at the crossover frequency \( w_0 \), i.e. \( |G(jw_0)| = 1 \)
- Then the delay margin is

\[
DM = \frac{PM \pi}{180 w_0} [\text{s}]
\]

Exercise: compute delay margin for these 3 cases
Delay (cont’d)

Static scheduling vs Minimal control delay

[Graphs showing the comparison between static scheduling and minimal control delay]
Delay (cont’d)

How to compensate the delay?
There are several ways

- Minimize the delay (case B - Minimal control delay)
- Compensate it off-line
- Make the controller robust (case A - static scheduling)
- Compensate on-line

Exple: Code that minimize the delay

```plaintext
LOOP %%% At each clock interrupt
  ADin
  CalculateOutput
  DAout
  UpdateStates
  IncTime %%% Evaluate remaining time
  WaitUntilTe
END
```
Delay (cont’d)

Exercise
Consider the following controller,

\[ x(k + 1) = Fx(k) + Gy(k) \]
\[ u(k) = Cx(k) + Dy(k) \]
Delay (cont’d)

Exercise
Consider the following controller,

\[
\begin{align*}
    x(k+1) &= Fx(k) + Gy(k) \\
    u(k) &= Cx(k) + Dy(k)
\end{align*}
\]

LOOP
ADin(y);
%%% CalculateOutput
u := u1 + D*y;
DAout(u)
%%% UpdateStates
x := F*x + G*y;
u1 := C*x;
%%% Wait for the next IncTime

Note that such a structure is not the only one! Controller can work on interrupts (exple: Brushless motor)
Quantification

Effects

- Non linear phenomena
- Limit cycles

Example (stable for $K<2$)

\[
H(z) = \frac{0.25}{(z-1)(z-0.5)}
\]
Quantification (cont’d)

Results

\[ K = 0.8 \]

\[ K = 1.2 \]

\[ K = 1.6 \]
Quantification (cont’d)

Results
The idea behind discretisation of a controller is to translate it from continuous-time to discrete-time, i.e.

\[ \frac{A}{D} + \text{ algorithm } + \frac{D}{A} \approx G(s) \]

To obtain this, few methods exist that approach the Laplace operator (see lecture 1-2).

Recall

\[
\begin{align*}
    s &= \frac{z - 1}{T_e} \\
    s &= \frac{z - 1}{zT_e} \\
    s &= \frac{2(z - 1)}{T_e(z + 1)}
\end{align*}
\]
Conclusion

- A state space approach to pole placement control
- A similar design can be done using a polynomial approach
- Continuous but directly extended to discrete-time systems.