Applied Control Systems

Part 03

Linear Control – Practical aspects

Y. SELLAMI 2019

1

Summary

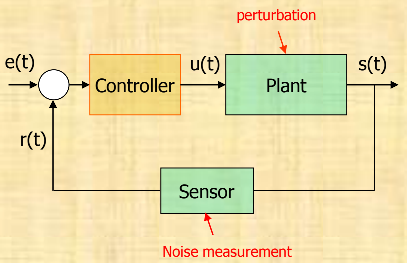
part 1 ▪ Introduction, Internal/external model, Discrete/continuous transfer function, Discrete/continuous state space representation, SISO/MISO
Conversion from continuous to discrete, Stability analysis…

part 2 ▪ Sensors / Actuators :
Modeling temperature, position, speed, acceleration, IMU sensors
Hydraulic, pneumatic, electrical actuators
PWM drivers

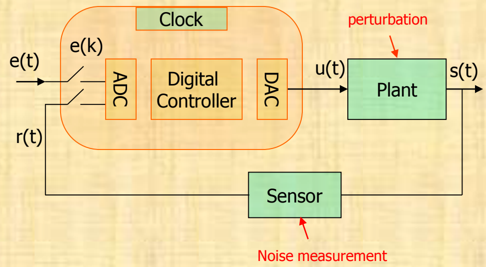
part 3 ▪ Practical aspects of control systems (PID, lead/lag; filtering, windup, …),
Converting continuous controller into discrete controller,
Control with state space representation …

Introduction

- Continuous control vs. Discrete control



Continuous time control system
(Analog control)



Discrete time control system
(Digital control)

Review: Representation of LTI systems

- LTI models

Continuous linear system: external model

$$\ddot{y}_1 + 3y_1 + \dot{y}_2 + y_2 = u_1 + u_2$$
$$\dot{y}_2 - y_1 + 3y_2 = 3u_1 + u_3$$

Discrete linear system: external model

$$y_1(k) + 0.1y_1(k-1) + y_2(k-2) = u_1(k) + u_2(k)$$
$$y_2(k) + 0.2y_1(k-1) - y_2(k-2) = 2u_2(k)$$

Continuous transfer function

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11(s)} & G_{12(s)} & G_{13(s)} \\ G_{21(s)} & G_{22(s)} & G_{23(s)} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \end{bmatrix}$$

Discrete transfer function

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} G_{11(z)} & G_{12(z)} \\ G_{21(z)} & G_{22(z)} \end{bmatrix} \begin{bmatrix} U_1(z) \\ U_2(z) \end{bmatrix}$$

G_{ij} : transfer function that links the output i to the input j (assuming the other inputs null)

Continuous state space

$$\dot{X} = AX + Bu$$
$$y = CX + Du$$

p entrées et q sorties

$X: 1 \times n$, $A: n \times n$, $B: n \times p$, $u: p \times 1$
$y: q \times 1$, $C: q \times n$, $D: q \times p$

Discrete state space

$$X_{k+1} = FX_k + Gu_k$$
$$y_k = CX_k + Du_k$$

p entrées et q sorties

$X_k: 1 \times n$, $F: n \times n$, $G: n \times p$, $u_k: p \times 1$
$y_k: q \times 1$, $C: q \times n$, $D: q \times p$

Review: Continuous → Discrete (transfer function)

- Continuous → Discrete

Differential Equations --> Recurrence Equations. Substitute the derivatives (in the diff. equ.) with the approximations:

$$\dot{x}(t) \approx \frac{x_k - x_{k-1}}{T} \quad , \quad \ddot{x}(t) \approx \frac{\dot{x}_k - \dot{x}_{k-1}}{T} = \frac{x_k - 2x_{k-1} + x_{k-2}}{T^2}, \ldots$$

First order method. Substitute the s operator (in the TF) with the approximation:

$$s \approx \frac{1}{T}(1 - z^{-1}) = \frac{1}{T}\frac{z-1}{z}$$

Bilinear (Tustin) method. Substitute the s operator (in the TF) with the approximation:

$$s \approx \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$$

Zero order hold method. Take into account the DAC model and use the Z transform table of usual signals:

$$G(z) = \mathcal{Z}\{B_0(s)G(s)\} = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\}$$

Review: Continuous → Discrete (state space)

- State space representation

Linear case: $\dot{X} = AX + Bu + H_c v$ $y = CX + Du + L\,w$	$X_{k+1} = F\,X_k + G\,u_k + H_d v_k$ $y_k = CX_k + Du_k + L\,w_k$ **Exact form exacte of F and G :** $\boldsymbol{F} = \boldsymbol{e}^{\boldsymbol{A}\,\boldsymbol{T}} \cong I + A\,T + A^2\frac{T^2}{2!} + \cdots$ $\boldsymbol{G} = \int_0^T e^{A\,t}\,B\,dt \quad ; \quad \boldsymbol{H_d} = \int_0^T e^{A\,t}\,H_c\,dt$ F = expm(A*Ts) : matrix exponential of A.Ts
	1st order approximation $\dot{\boldsymbol{X}} \cong (\boldsymbol{X}_{k+1} - \boldsymbol{X}_k)/\boldsymbol{T}$ $X_{k+1} = (\boldsymbol{I} + \boldsymbol{A}\,\boldsymbol{T})\,X_k + \boldsymbol{B}\,\boldsymbol{T}\,u_k$ $y_k = C\,X_k + D\,u_k$
Nonlinear case: $\dot{x}(t) = f\big(x(t), u(t), v(t)\big)$ $y(t) = h\big(x(t), u(t), v(t)\big)$	**1st order approximation** $x_{k+1} = x_k + T\,f(x_k, u_k, v_k) = f_d(x_k, u_k, v_k)$ $y_k = h(x_k, u_k, v_k)$

Stability/performance analysis

of a controlled system

7

Stability : Definitions

- Standard definition

  - A LTI system is stable if its impulse response $y_\delta(t)$ is bounded
    It is asymptotically stable if $y_\delta(t) \rightarrow 0$ when $t \rightarrow \infty$
    It is marginally stable if it is not asymptotically stable but $y_\delta(t)$ is bounded

- Relative stability
  - Relative stability defines the stability degree of the system.
    How stable is the system (does it become more or less stable than before)?

    Numerically measure of the stability is needed.
    Two known measures of relative stability: phase margin and gain margin.

Stability of LTI systems

- By studying the poles (or eigen values)

A linear system : is stable if its impulse response is bounded.

Continuous Case:	Discrete case:
• In time domain: $g(t) = \mathcal{L}^{-1}\{FTBF(s)\}$: is bounded For MIMO systems : The impulse responses of all pairs (I/O) are bounded • In complex domain (S plane): $Real(poles) \leq 0$ • In continuous state space: $\dot{x} = Ax + Bu$ $Real(\lambda_i) \leq 0$ $eigen\ values\ of\ A: \lambda_i \equiv poles\ of\ CLTF(s)^*$ CE**: $det(sI - A) \equiv denominator of\ CLTF(s)$	• In time domaine: $g(k) = \mathcal{Z}^{-1}\{FTBF(z)\}$: is bounded For MIMO systems : The impulse responses of all pairs (I/O) are bounded • In complex domain (Z plane): $Module(poles) \leq 1$ • In discrete state space: $x_k = Fx_{k-1} + G\,u_{k-1}$ $Module(\lambda_i) \leq 1$ $eigen\ values\ of\ F: \lambda_i \equiv poles\ of\ CLTF(z)$ CE: $det(zI - F) \equiv denominator\ of\ CLTF(z)$

*CLTF : closed loop transfer function
**CE : caracteristic equation

9

Performance analysis

- Performance analysis

Temporal analysis (peak response, settling time, rise time, overshoot, steady state error, minimal phase, …)	Easy to study linear systems, but it is not generally the case with nonlinear systems.
Frequency analysis (Bode/Nyquist diagram …) If the input of a linear system is sinusoidal, then the output is also **sinusoidal**, with the **same frequency**. It is possible to plot the Bode diagram.	For nonlinear systems, the output is not necessary sinusoidal. - No Bode diagram - No Laplace/Z transforms - No transfer function
	Analysis by phase plane (if n = 2)
Stability : If a linear system is stable, then the impulse response is bounded **for all initial conditions**.	For nonlinear systems, there is a region of stability. If the system is initially outside this region, the stability is not guaranteed.

10

Controller design

- Criterion for designing a regulator

- Compromise between stability / precision
- Good temporal behavior (acceptable errors, …)
- Stable for all initial conditions in the operation region
- Robust control according to parameter uncertainties, modeling uncertainties and external disturbances

Control design

from transfer function representation

Discrete/Continuous conversion, Filtered PID, Anti-windup

Some known controllers

- Some known classical control:
  - PID control, lead / lag control
  - Discrete control $C(z)$ obtained from continuous control $C(s)$ by discretization (see part 1)
  - Pole placement control
  - R-S-T control,
  - …

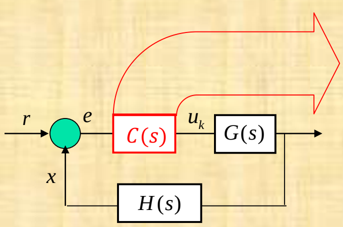
Implementation of a continuous controller $C(s)$

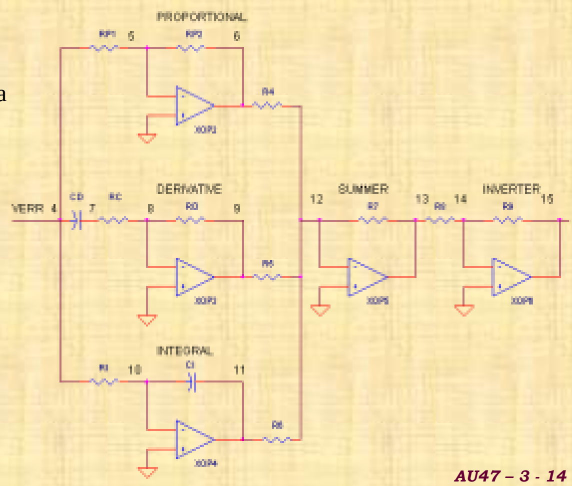
- Implementing a digital controller from a discrete transfer function

Continuous transfer function $C(s)$ can be implemented by using operational amplifiers.

- What about nonlinear controllers?

- What should you do if you design a controller with op-amp and you realize that you made a mistake?

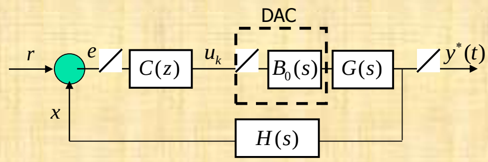




Implementation of a discrete controller $C(z)$

- Implementing a digital controller from a discrete transfer function

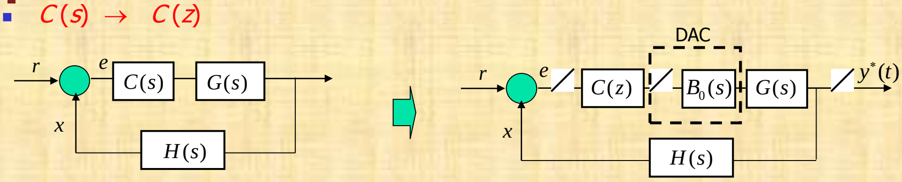
Once the discrete transfer function $C(z)$ designed, write the corresponding recurrence equation expressing the new control output $u(k)$ as a function of the previous control outputs $u(k\text{-}1)$, $u(k\text{-}2)$, … and the inputs $e(k)$, $e(k\text{-}1)$, …



For real-time purposes, the control output $u(k)$ should be updated at specific sample times. This could be done by configuring a *timer* and associated *interrupts*.

From continuous to discrete controller

- $C(s) \rightarrow C(z)$



If a continuous controller $C(s)$ is previously designed, it is possible to design a discrete controller $C(z)$ by **discretization** (see part 1 of this course).

The most commonly used discretization method for controllers is **Tustin** (Bilinear approximation). The phase is better approximated.

In Matlab:

```
>> Sc = tf(Num,Den); or Sc = zpk(zeros,poles,k); or s=tf('s'), G=...
>> Sd = c2d(Sc , T, 'tustin');
```

Classical PID control

- Proprortional-Integral-Derivative (PID) : Continuous / Discrete

Continuous PID

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \dot{e}(t)$$

$$C(s) = \frac{U(s)}{E(s)} = K_P + K_I \frac{1}{s} + K_D s$$

A classical discrete PID

$$u(k) = K_P e(k) + K_I \sum_{i=0\ldots k} e(i) + K_D \frac{e(k) - e(k-1)}{T}$$

$$C(z) = K_P + K_I T \frac{z}{z-1} + K_D \frac{1}{T}\frac{z-1}{z}$$

Objectives : is to find appropriate values of the parameters K_P, K_I, K_D in order to have:
♦ Stable system, even if the OL is unstable
♦ Improved performances (stability / precision / …)

Proportional action	*K*	*K*	Rise time, precision and stability
Integral action	1/*s*	*z*/(*z*–1)	Rise time, precision and stability
Derivative action	*s*	(*z*–1)/*z*	Overshoot and stability

Implementation of classical PID

- Parallel form $C(z) = K_p + K_I T \frac{1}{1 - z^{-1}} + \frac{K_D}{T}(1 - z^{-1})$
- Other form $C(z) = K_p \left(1 + \frac{T}{T_i}\frac{1}{1 - z^{-1}} + \frac{T_D}{T}(1 - z^{-1})\right)$

T: sample time
K_p: proportional gain
T_i: integral time constant
T_D: derivative time const

- PID = 2 zeros + 1 pole $\boxed{C(z) = \frac{K_1 + K_2 z^{-1} + K_3 z^{-2}}{1 - z^{-1}} = \frac{U(z)}{E(z)}}$

with $K_1 = K_p\left(1 + \frac{T}{T_i} + \frac{T_D}{T}\right)$; $K_2 = K_p\left(1 + \frac{2T_D}{T}\right)$; $K_3 = K_p \frac{T_D}{T}$

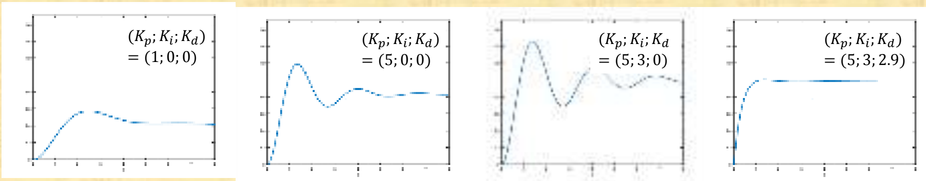
- Implementation

$$u(k) = u(k-1) + K_1 e(k) + K_2 e(k-1) + K_3 e(k-2)$$

Tuning PID

- Manual tuning (see this wikipedia link)
  - Set K_i and K_d to zero, increase K_p until the output oscillates, then K_p should be set to approximately half of that value.
  - Increase K_i until the offset output is corrected in sufficient time
  - Increase K_d until the output is acceptably quick to reach its reference after a load disturbance

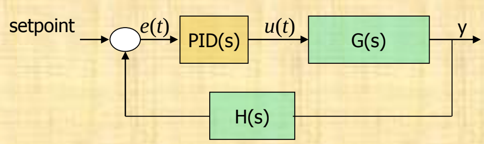
  Be careful: Too much K_i will cause instability, and too much K_d will cause excessive response and overshoot.



- Automatic tuning
  - Some PID tuning software use available I/O data and loop optimization methods to ensure consistent results
  - Example: Control Design Toolbox (with Simulink)

PID Filter Control (continuous case)

- Problems with derivative action

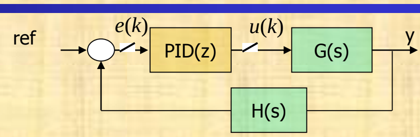


Plot the Bode gain diagram (spectrum) of the derivative action s.
In classical PID controllers, the derivative action accentuates high frequency noises.
⇒ Amplification of the measurement noise ⇒ can damage the actuator
In practice, PID filter control is preferred to classical PID.

$$PID(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d\, s \quad \Rightarrow \quad \boxed{PID(s) = K_p + \frac{K_i}{s} + K_d \frac{s}{1 + s/N}}$$

♦ Compare the Bode gain plot (spectrum) of the derivative action s to the filtered action.
Give a physical signification of the parameter N ?

PID Filter Control (discrete case)



- Classical PID:

$$C(z)=\frac{U(z)}{E(z)}=K_P+K_I T\frac{z}{z-1}+K_D\frac{1}{T}\frac{z-1}{z}$$

- In order ro reduce high frequency noises PID filter control is used.

$$K_D\frac{1}{T}\frac{z-1}{z} \quad \Rightarrow \quad K_D\frac{1}{T}\frac{z-1}{z-\alpha} \quad or \quad K_D\frac{1}{T}\frac{1-z^{-1}}{1-\alpha z^{-1}} \qquad \alpha<1$$

which leads to:

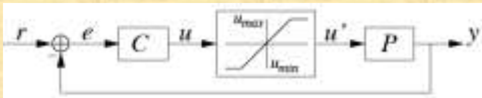
$$C(z)=\frac{U(z)}{E(z)}=K_P+K_I T\frac{z}{z-1}+K_D\frac{1}{T}\frac{z-1}{z-\alpha}$$

Windup (problems with integration)

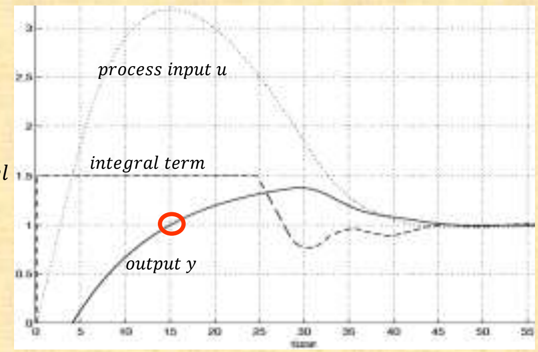
- Integral term
  - Integral term $u(k)=K\sum^k e(i)=u(k-1)+Ke(k)$ provides the "memory" to allow a controller to continue to generate output u when the error is not null.
  - Integral term accumulates the signed error remaining after each control cycle to use in the next control cycle.
  - Integral term stop varying when the output reaches the setpoint (reference)
- Integrator windup phenomenon
  - Integrator windup occurs when a large change in setpoint occurs (example: step response) and the integral term accumulates a significant error that leads to actuator saturation
  - The control error increases more slowly than in ideal case (where there is no saturation limits), therefore the integral term becomes large (it winds up).
  - Even if the output reaches the reference, the controller still saturates due to the integral term, that leads to large overshoots and setting times.

Windup phenomenon

- Integrator windup phenomenon



Control sheme with saturation
$(u_{min} = 0\,; u_{max} = 1.5)$

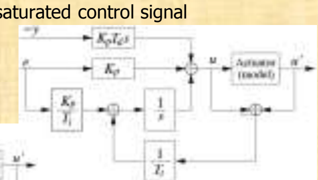


Step response:
At $t = 15$, $y = setpoint\ value\ (e = 0)$
but integral term is high
$\Rightarrow$ *the output still remains at the* max *level*

The error should not be integrated when saturation

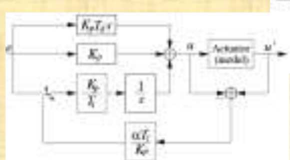
Anti-windup (solutions)

- Some solutions to counter with integrator windup phenomenon
  - Avoiding saturation of the control input u, by limiting or smoothing the setpoint changes
    - $\Rightarrow$ difficult in practice
  - Conditional integration: switching off the integration within a certain condition
    - Integral term is limited to a certain value
    - Integration is stopped when the error is greater than a certain threshold
    - Integration is stopped when the control variable saturates
  - Back-calculation: re-computing the integral term when the controller saturates
    - By feeding back the difference of the saturated and unsaturated control signal
    - $u_i = \int \frac{K_p}{T_i} e + \frac{1}{T_t}(u' - u) \quad dt$ when saturation occurs;
    - in general : $T_t = T_i \quad or \quad K_p \quad \ldots$



  - Combined approaches
    - Variable structure PID

  …



Source: Practical PID Control, Antonio Visioli, Ed. Springer, 2006

Control design

from state space representation

Controlability, Pole placement control

26

Controllability

- Controllability of LTI systems :

A linear system is controllable if it possible, by means of an external input u(t), to move the internal states from any initial state to any other final state in a finite time interval [t0,tf].

LTI system is controllable if and only if the **Controllability matrix M_c** has full row rank.

For continuous case: $\boldsymbol{M_c = [B \;\; AB \;\; A^2B \;\; \dots \;\; A^{n-1}B]} \quad ; \quad \boldsymbol{rank(M_c) = n}$

For discrete case: $\boldsymbol{M_c = [G \;\; FG \;\; F^2G \;\; \dots \;\; F^{n-1}G]} \quad ; \quad \boldsymbol{rank(M_c) = n}$

Invariance with state transformation:

	State transformation	
$\dot{x} = Ax + Bu$	$z = Px$	$\dot{z} = A_2 z + B_2 u$ $\dot{z} = PAP^{-1}z + PBu$
$M_c(A_2, B_2) = M_c(PAP^{-1}, PB) = P \cdot M_c(A, B)$		
(A, B) controllable	⇔	(A_2, B_2) controllable

Observability

- Observability

A system is observable if it is possible to determine any (arbitrary initial) state x(0) by using only finite records of the output y(t), t<T.

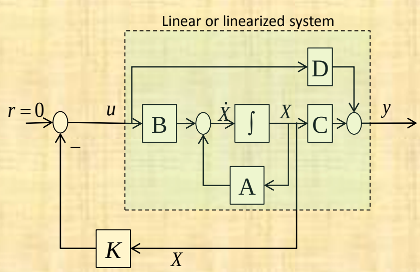
LTI system is observable if the observability matrix $M_o(A, C)$ (for continuous case) or $M_o(F, C)$ (for discrete case) has full colomn rank.

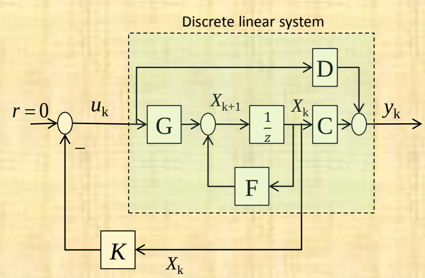
Continuous case: $$\boldsymbol{M_o(A,C)} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA^{n-1}} \end{bmatrix} \quad ; \quad \boldsymbol{rank(M_o) = n}$$

Discrete case: $$\boldsymbol{M_o(F,C)} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CF} \\ \vdots \\ \boldsymbol{CF^{n-1}} \end{bmatrix} \quad ; \quad \boldsymbol{rank(M_o) = n}$$

Pole placement

- Pole placement control for linear systems





$$\boldsymbol{u = -KX + r}$$

Verify that for linear SISO systems, **K** matrix is a row matrix of size $\mathbf{1} \times \boldsymbol{n}$

Pole placement: Principe

- Pole placement control

Plant:	Control law:
$x_{k+1} = Fx_k + Gu_k$ $y_k = Cx_k$	$\boldsymbol{u_k = -Kx_k + r}$
Closed loop system: $x_{k+1} = Fx_k + G(-Kx_k + r)$ $y_k = Cx_k$	$x_{k+1} = (\boldsymbol{F - GK})x_k + Gr$ $y = Cx$

$(\boldsymbol{F - GK})$ is the new state matrix (of the closed loop system).

In open loop, the characteristic equation of the state matrix F :

$$CE_F = det(\lambda I - F) \quad = (\lambda - \lambda_1) \dots (\lambda - \lambda_n) = \lambda^n + a_1\lambda^{n-1} + \dots + a_n$$

In closed loop, the characteristic equation of the new state matrix $F - GK$:

$$CE_{CL} = det(\lambda I - F + GK) = (\lambda - \mu_1) \dots (\lambda - \mu_n) = \lambda^n + \alpha_1\lambda^{n-1} + \dots + \alpha_n$$

The aim is to find a matrix **K** so that the closed loop system achieves desired poles : $\mu_1 \dots \mu_n$, (or equivalently CE_{CL} achieves a desired characteristic equation $\lambda^n + \alpha_1\lambda^{n-1} + \dots + \alpha_n$).

Pole placement for SISO systems

- By identification
  - Compute the $\mathrm{CE_{CL}} = \det(sI - F + GK)$ of the closed loop system
  - Choose desired poles and write the desired CE
  - Compute the parameters of K by identifying the two above polynomials
  - It is easier to compute K by writing the system in a modal form before identification
  - There are other methods for computing the gain matrix K (like Ackermann method).
    Details on the theoretical aspects of Pole placement are not considered in this lecture.

- In Matlab

with Control System Toolbox

```
Mc = cntl(F,G);     % controllability matrix, equivalent to  Mc = [G , F*G , F^2*G, ...]
rank(Mc)==n         % verify if all the dynamics of the system are controllable
      K = place(F,G,[-10,-2+4i,-2-4i]); % 3rd parameter = desired poles
or : K = aker(A,B,[-10,-2+4i,-2-4i]);
```

`place` and `aker` give the same result for linear **SISO systems**.

Pole placement for MIMO systems

- MIMO case

Unlike SISO case, there is no unique solution for the control matrix K.

The size of K matrix is $p \times n$. There are **infinite number of solutions for K**.

place and aker do not give the same result.

Plant: $x_{k+1} = Fx_k + Gu_k$ $y_k = Cx_k$	**Control law:** $u_k = -Kx_k + r$
Closed loop system: $\boxed{\boldsymbol{x_{k+1} = (F - GK)x_k}} + Gr$ $y_k = Cx_k$	$u_k : p \times 1$; $G : n \times p$; $K : n \times p$

$$\boxed{\begin{array}{l} \boldsymbol{K = place(F, G, [\mu_1, \dots, \mu_n])} \\ \boldsymbol{K = aker(F, G, [\mu_1, \dots, \mu_n])} \end{array}}$$

$\mu_1, \dots, \mu_n$ are the desired poles of the closed loop system (eigen values of $F - GK$).

- Any questions?

Do not hesitate to let me know
if you find any errors!!!