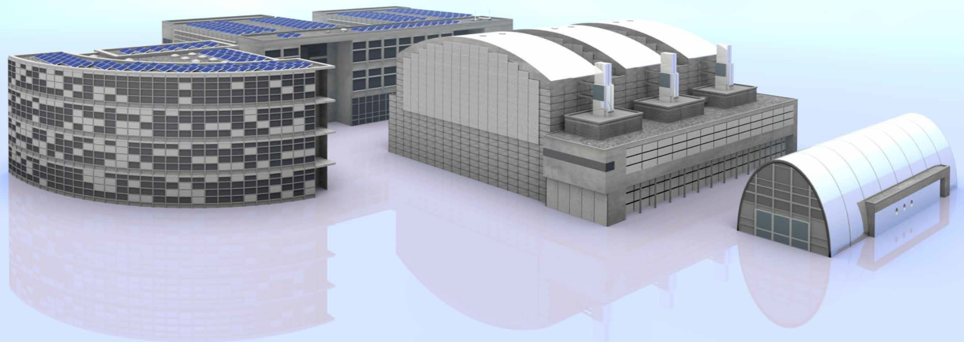
Control and operation of tokamaks PHYS-734
Session 1 Fundamentals in control theory

Federico Felici
(Based on slides by Jean-Marc Moret)

Swiss Plasma Center
École Polytechnique Fédérale de Lausanne
6-17th Feb. 2023



Content

Dynamical systems
- Linear time invariant (LTI) systems
- Plants

Integral transforms
- Laplace
- z-transform (sampled signals)

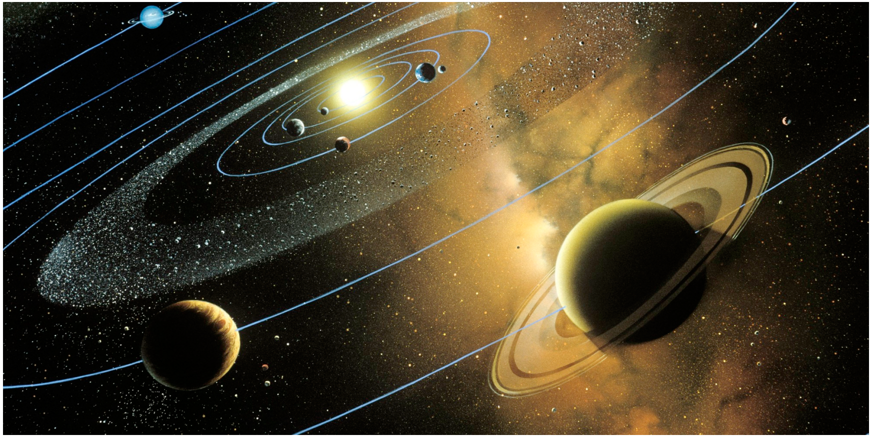
Transfer functions
- continuous time
- discrete time
- continuous / discrete time mappings

Control systems
- controller architectures
- relevant closed loop performances
- input / output noise rejection
- feedforward

Controller design
- closed loop stability
- gain and phase margin
- Nyquist stability criterion
- root locus method
- PID controllers

Dynamic(al) systems

Mathematical concept consisting in



- S : a state space whose variables describe the state of the system at any point of its evolution
- φ : an evolution rule, a function that specifies the future of all state variables given only the present state
- T : a group to label the state evolution (time)

$$\varphi : T \times S \to S$$

Classification of dynamical systems

The state space S can be

- continuous
- discrete (on/off, lattices, finite state machines, automata)

- finite dimensional
- infinite dimensional (temperature distribution in a solid, PDEs)

The evolution rule φ can be

- deterministic (each state has a unique consequent)
- stochastic (random future state)

The *time* T can be

- discrete (population generations, sampled systems)
- continuous

Discrete time

Deterministic evolution rule for a discrete time dynamical system, $\varphi : \mathbb{Z} \times S \to S$, such that

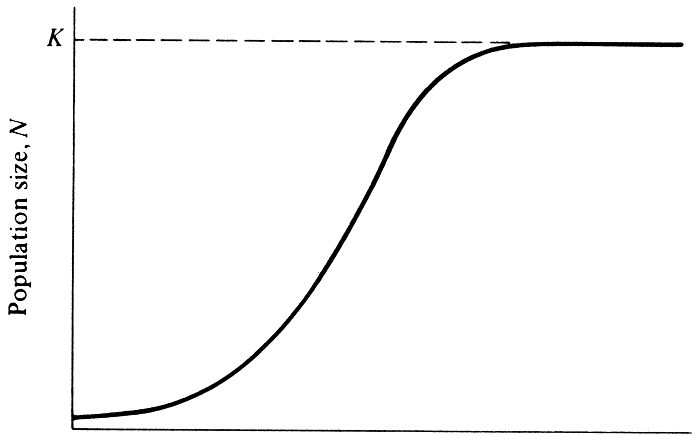
$$x(k+1) = \varphi\left([k,]\, x(k)\right)$$

Example: population equation

$$N(k+1) = rN(k)\left(\frac{K - N(k)}{K}\right)$$

r : growth rate

K : carrying capacity



Continuous time

Deterministic evolution rule for a continuous time dynamical system with continuous state space

- $\varphi : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}^n ; (t,x) \mapsto \varphi(t,x)$
- Orbit : $x(t) = \varphi(t, x(0))$ with $x(0)$ the initial conditions
- Identity : $\varphi(0,x) = x$
- Associativity : $\varphi(t+s,x) = \varphi(t, \varphi(s,x))$
  Dynamics can be restarted at any time s to get the same resulting orbit (future depends only on present)
- Differentiability : $\dfrac{d\varphi}{dt}$ exists, so the orbit is the solution of an ODE, i.e. most of the physical time evolution laws : $\dfrac{dx}{dt} = \dfrac{d\varphi(t,x(0))}{dt} \equiv \Phi(t,x)$

Linear time independent dynamical systems

Continuous time: $\Phi(t,x)$ is a linear time independent function of x

$$\frac{dx}{dt} = \Phi(\cancel{t},x) = A \cdot x$$

With A a $n \times n$ matrix, this is a 1st order homogeneous linear ODE

Discrete time: the map $\varphi(t,x)$ is time independent and linear in x

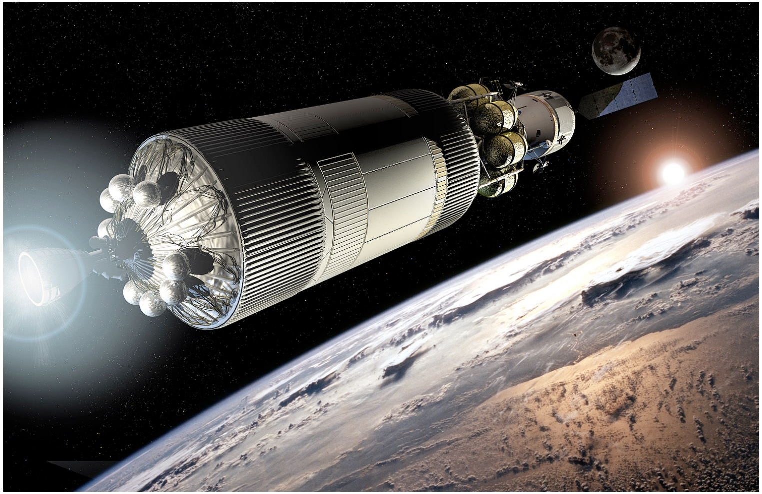
$$x(k+1) = \varphi\left(\cancel{k},x(k)\right) = A \cdot x(k)$$

This is a 1st order difference linear equation

Both are **state space model** of linear dynamical systems

Plants

A plant is a dynamical system driven by externally imposed actions u (inputs). Formally



$$\dot{x} = \Phi(t, x, u)$$

$$x(k+1) = \varphi\big(k, x(k), u(k)\big)$$

The **measured quantities** or the relevant parameters of the plant y (outputs) may not directly be the state variables, but are given by some function $y = h(x, u)$

Linear plants - state space model

Linear functions for $\Phi(x,u)$ and $h(x,u)$

$$\begin{cases} \dot{x}(t) = \underbrace{A}_{n_x \times n_x} \cdot x(t) + \underbrace{B}_{n_x \times n_u} \cdot u(t) \\ y(t) = \underbrace{C}_{n_y \times n_x} \cdot x(t) + \underbrace{D}_{n_y \times n_u} \cdot u(t) \end{cases}$$

For discrete time plants

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

A defines the dynamic (evolution rule) of the dynamical system
B defines how each input acts on the evolution of each state
C relates the outputs and the states (measurement matrix)
D direct feedthrough (sensitivity of the measurements to the inputs)

```
Matlab>> sys = ss(A,B,C,D);
```

State space transformation

Apply the similarity transformation $x' \equiv T \cdot x$ to the state space model (T is a square invertible matrix)

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = C \cdot x + D \cdot u \end{cases} \xrightarrow[x = T^{-1} \cdot x']{} \begin{cases} T^{-1} \cdot \dot{x}' = A \cdot T^{-1} \cdot x' + B \cdot u \\ y = C \cdot T^{-1} \cdot x' + D \cdot u \end{cases} \xrightarrow[T \cdot]{} \begin{cases} \dot{x}' = T \cdot A \cdot T^{-1} \cdot x' + T \cdot B \cdot u \\ y = C \cdot T^{-1} \cdot x' + D \cdot u \end{cases}$$

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = C \cdot x + D \cdot u \end{cases} \xrightarrow{\substack{A' \equiv T \cdot A \cdot T^{-1} \\ B' \equiv T \cdot B \\ C' \equiv C \cdot T^{-1} \\ D' \equiv D}} \begin{cases} \dot{x}' = A' \cdot x' + B' \cdot u \\ y = C' \cdot x' + D' \cdot u \end{cases}$$

The choice of T has no influence on the input to output relation of the plant (same output for same input)

System stability

Use the eigenvalue decomposition of A, $A = V \cdot P \cdot V^{-1}$ (P is diagonal), to define the state space transformation $x' = V^{-1} \cdot x, A' \equiv V^{-1} \cdot A \cdot V = P$

The homogenous ($u = 0$) evolution equation is then

$$\dot{x}' = P \cdot x' \rightarrow \dot{x}_n' = p_n x_n', n = 1 \ldots n_x \rightarrow x_n'(t) = x_n'(0) e^{p_n t}$$

The case of complex eigenvalues (they come in complex conj. pairs)

$$\begin{cases} p_n = \gamma_n + i\omega_n \\ p_{n+1} = \gamma_n - i\omega_n \end{cases} \Rightarrow x_n'(t) + x_{n+1}'(t) = e^{\gamma_n t}\left(x_n'(0) e^{i\omega_n t} + x_{n+1}'(0) e^{-i\omega_n t} \right)$$

The system is stable (in the sense that its solution is finite), if for all eigenvalues $\boxed{\Re\left(p_n\right) \leq 0}$

For discrete time systems

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

with $u = 0$ and $A' = P$

$$x'(k+1) = P \cdot x'(k) \Rightarrow x'_n(k+1) = p_n x'_n(k), n = 1 \ldots n_x$$

The system is stable if for all z-plane eigenvalues $\boxed{|p_n| \leq 1}$

Integral transforms

You all know about the Fourier transform of a time signal $f(t)$:

$$\mathcal{F}(f(t)) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t)\,dt \equiv F(\omega) \text{ with } \omega \in \mathbb{R}$$

Integral transform with a more general kernel $g(\gamma, t)$

$$\mathcal{G}(f(t)) = \int g(\gamma, t) f(t)\,dt \equiv F(\gamma)$$

Laplace transform: $g = e^{-st}$

$$\mathcal{L}(f(t)) = \int_{0}^{\infty} e^{-st} f(t)\,dt \equiv F(s) \text{ with } s \in \mathbb{C}$$

Laplace transform user guide

Useful properties / applications

Linearity	$\mathcal{L}\left(af(t)\right)=a\mathcal{L}\left(f(t)\right)$ with $a\in\mathbb{R}$	LTI
Superposition	$\mathcal{L}\left(f(t)+g(t)\right)=\mathcal{L}\left(f(t)\right)+\mathcal{L}\left(g(t)\right)$	LTI
Time derivation	$\boxed{\mathcal{L}\left(\dot{f}(t)\right)=s\mathcal{L}\left(f(t)\right)}$	ODE
Time delay	$\mathcal{L}\left(f(t-\tau)\right)=e^{-s\tau}\mathcal{L}\left(f(t)\right)$	Discrete time
Final value	$\lim\limits_{s\to 0} sF(s)=\lim\limits_{t\to\infty} f(t)$	DC gain: just remember $s=i\omega=0$
Fourier	$F(\omega)=F(s=i\omega)$	Frequency response

Transfer functions - continuous time

Laplace transform of the state space equations

$$\begin{cases} \dot{x}(t) = A \cdot x(t) + B \cdot u(t) \\ y(t) = C \cdot x(t) + D \cdot u(t) \end{cases} \rightarrow \begin{cases} sx(s) = A \cdot x(s) + B \cdot u(s) \\ y(s) = C \cdot x(s) + D \cdot u(s) \end{cases}$$

Using $(s\mathrm{I} - A) \cdot x(s) = B \cdot u(s)$, define transfer functions in the Laplace domain as ratios of the transform of the input and output signals

$$\frac{x(s)}{u(s)} = (s\mathrm{I} - A)^{-1} \cdot B$$

$$\underbrace{H}_{n_y \times n_u}(s) \equiv \frac{y(s)}{u(s)} = C \cdot (s\mathrm{I} - A)^{-1} \cdot B + D$$

We have a MIMO system (**M**ultiple n_u **I**nput **M**ultiple n_y **O**utput)

Each transfer function is a rational function of s: take the state space transformation where $A' = P$ (diagonal) (this does not change the input output relationship)

$$H(s) = C' \cdot (s\mathrm{I} - P)^{-1} \cdot B' + D'$$

$$(s\mathrm{I} - P)^{-1} = \begin{pmatrix} s - p_1 & & 0 \\ & \ddots & \\ 0 & & s - p_{n_x} \end{pmatrix}^{-1} = \begin{pmatrix} \dfrac{1}{s - p_1} & & 0 \\ & \ddots & \\ 0 & & \dfrac{1}{s - p_{n_x}} \end{pmatrix}$$

$$H(s) = C' \cdot \begin{pmatrix} \frac{1}{s-p_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{s-p_{n_x}} \end{pmatrix} \cdot B' + D'$$

$$H_{ij}(s) = \sum_{k=1}^{n_x} \frac{C'_{ik} B'_{kj}}{s-p_k} + D'_{ij} = \frac{\beta_0 + \beta_1 s + \ldots + \beta_M s^M}{1 + \alpha_1 s + \ldots + \alpha_N s^N}$$

with $N = n_x$, $M = \begin{cases} N-1, D'_{ij} = 0 \\ N, D'_{ij} \neq 0 \end{cases}$

Note that all the transfer functions share the same denominator (poles)

Zeros and poles

Let $H(s)=\dfrac{N(s)}{D(s)}$ be a continuous-time (SISO) transfer function

- Poles of $H(s)$ are the roots of $D(s)$
- Zeros of $H(s)$ are the roots of $N(s)$

Let (A,B,C,D) be a state-space representation of $H(s)$ such that

$$H(s)\equiv\frac{y(s)}{u(s)}=C\cdot\left(s\mathrm{I}-A\right)^{-1}\cdot B+D$$

- Poles of $H(s)$ are the eigenvalues of A
- Zeros of $H(s)$ are finite values of s for which $\begin{pmatrix} sI-A & -B \\ C & D \end{pmatrix}$ loses rank

Exercise: Manipulating LTI systems in MATLAB

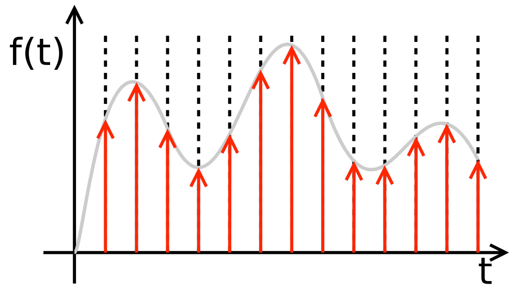
Explore the properties of the transfer function $G = \dfrac{(s+0.5)}{(s+1)(s+2)}$

Verify the system's stability, check that the step response settles to the DC gain.

```
>> s=tf('s') % define s operator
>> Htf = (s+1)/((s+3)*(s-2); % transfer function representation
>> pole(H) % poles of H - is it stable?
>> zero(H) % zeros of H
>> Hss = ss(H) % state-space representation
>> Hzpk = zpk(H) % zero-pole-gain representation
>> step(Htf,3) % plot 'step response' until t=3s
>> dcgain(Htf); % 'DC gain': value at s=0
```

Sampled signals

A sampled signal (with constant sampling time T) can be represented as



$$f(t)\cdot\sum_{k=-\infty}^{\infty}\delta(t-kT)$$

Its Laplace transform is

$$\int_0^{\infty}\sum_{k=-\infty}^{\infty}f(t)\delta(t-kT)e^{-st}dt=\sum_{k=0}^{\infty}f(kT)e^{-skT}\underbrace{\int_0^{\infty}\delta(t-kT)dt}_{1}$$

So it is natural to define the

z-transform

Let $z \equiv e^{sT}$; the Laplace transform of our sampled signal, $f(k) \equiv f(kT)$, becomes the z-transform (just a new kernel for the general integral transform)

$$\mathcal{Z}\big(f(k)\big) = \sum_{k=0}^{\infty} f(k) z^{-k} \equiv F(z)$$

z-transform user guide

1 sample delay	$\mathcal{Z}\big(f(k-1)\big)=z^{-1}\mathcal{Z}\big(f(k)\big)$	abusively written $\boxed{f(k-1)=z^{-1}f(k)}$
Final value	$\lim\limits_{z\to 1}(z-1)F(z)=\lim\limits_{k\to\infty} f(k)$	DC gain: just remember $f(k-1) \underset{\infty\leftarrow k}{\longleftarrow} f(k)$ $f(k-1)=f(k)$ $f(k-1)=z^{-1}f(k) \Leftrightarrow z=1$
Fourier	$F(\omega)=F(z=e^{i\omega T})$	Frequency response
$F(z)$ is a rational function		

Transfer functions - discrete time

z-transform of the state space equations

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

Using $\left(z\mathrm{I} - A\right)\cdot x(z) = B\cdot u(z)$, define transfer functions in the z domain as ratios of the transform of the input and output signals

$$\frac{x(z)}{u(z)} = \left(z\mathrm{I} - A\right)^{-1}\cdot B$$

$$H(z) \equiv \frac{y(z)}{u(z)} = C\cdot\left(z\mathrm{I} - A\right)^{-1}\cdot B + D$$

$H(z)$ is a **rational function**; similarly to the continuous time case

$$H_{ij}(z) = \sum_{k=1}^{n_x} \frac{C'_{ik} B'_{kj}}{z - q_k} + D'_{ij} = \sum_{k=1}^{n_x} z^{-1} \frac{C'_{ik} B'_{kj}}{1 - q_k z^{-1}} + D'_{ij} = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}}$$

with $N = n_x$

Recursive estimation of the output (IIR digital filter)

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}} = \frac{y(z)}{u(z)}$$

$$\left(1 + a_1 z^{-1} + \ldots + a_N z^{-N}\right) y(z) = \left(b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}\right) u(z)$$

$$y(k) + a_1 y(k-1) + \ldots + a_N y(k-N) = b_0 u(k) + b_1 u(k-1) + \ldots + b_N u(k-N)$$

$$\boxed{y(k) = b_0 u(k) + b_1 u(k-1) + \ldots + b_N u(k-N) - a_1 y(k-1) - \ldots - a_N y(k-N)}$$

Continuous to discrete time

You want to simulate / predict / control a

- physical system (continuous time) with
  - digitally acquired signals (ADCs)
  - a computer (clocked)

User guide:

- Take the continuous time Laplace domain transfer function

$$H(s)=\frac{\beta_0+\beta_1 s+\ldots+\beta_M s^M}{1+\alpha_1 s+\ldots+\alpha_N s^N}$$

- Remember $z\equiv e^{sT}$, so replace $s=\frac{1}{T}\ln z=-\frac{1}{T}\ln z^{-1}$. Oops, $H(z)$ is not a rational function of z^{-1}, so cannot apply $f(k-1)=z^{-1}f(k)$

- Find an approximation for s as a rational function of z^{-1}, eg. trapeze approximation of the integration (Tustin transform)

$$y(t)=\int_0^t u(t')dt' \leftrightarrow y(s)=\frac{1}{s}u(s)$$

$$y(k)=y(k-1)+T\,\frac{u(k)+u(k-1)}{2}$$

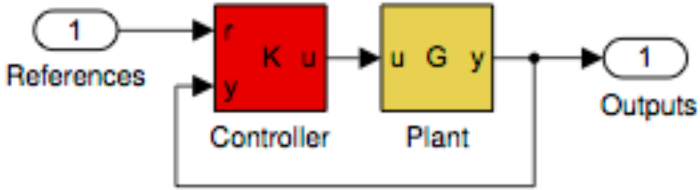
$$y(z)=z^{-1}y(z)+T\,\frac{u(z)+z^{-1}u(z)}{2}$$

$$y(z)=\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}u(z)$$

$$s\leftarrow\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$$

```
Matlab>> T=1e-3; Cz = c2d(Cs,T,'tustin');
```

Control systems

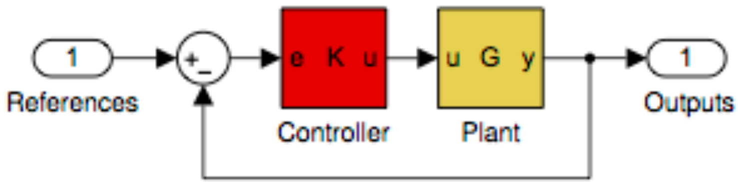


A controller (control system) is a dynamical system

- that takes as inputs
  - reference signals r and
  - the outputs y of the plant G
- and produces as outputs
  - the inputs u to be applied to the plant to satisfy
- control objectives:
  - stabilise an unstable plant G
  - the plant outputs y follow the references r
  - the control inputs u remain within given bounds
  - …
- $u = K(r, y)$ is often called the **control law**

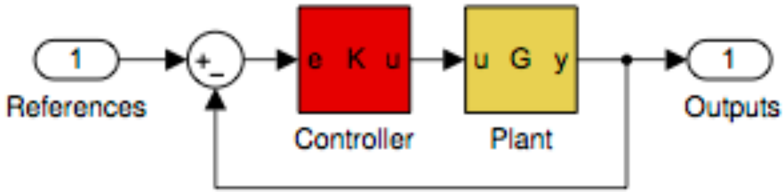
Exercise: Closed-Loop transfer function

- Let the control law be: $u = K(r - y)$ with linear transfer function *K*



- Write the transfer function, in terms of G(s) and K(s) describing the relation between:
  - *Y(s)* and *R(s)*
  - *E(s)* and *R(s)*
  - *U(s)* and *R(s)*

A simple controller

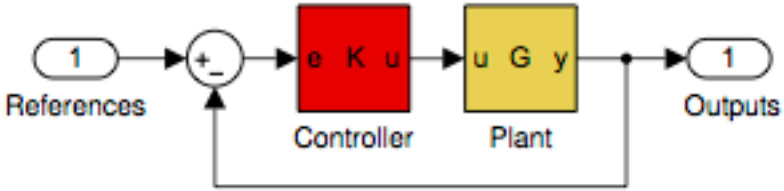


Reference tracking performance (closed loop, r to y)

$$y = Gu = GKe = GK(r-y) \to \boxed{y = \frac{GK}{1+GK}r} \to r$$

Tracking error performance (closed loop, r to e)

$$e = r - y = \left(1 - \frac{GK}{1+GK}\right)r \to \boxed{e = \frac{1}{1+GK}r} \to 0$$



Controller effort (closed loop, r to u)

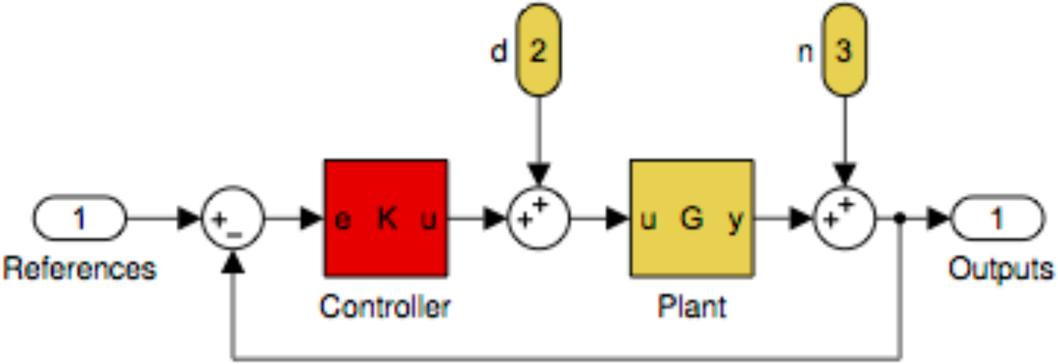
$$u = K(r-y) = K(r-Gu) \rightarrow \boxed{u = \frac{K}{1+KG} r} \rightarrow G^{-1} r$$

Open loop (open loop, r to y)

$$y = GKr$$

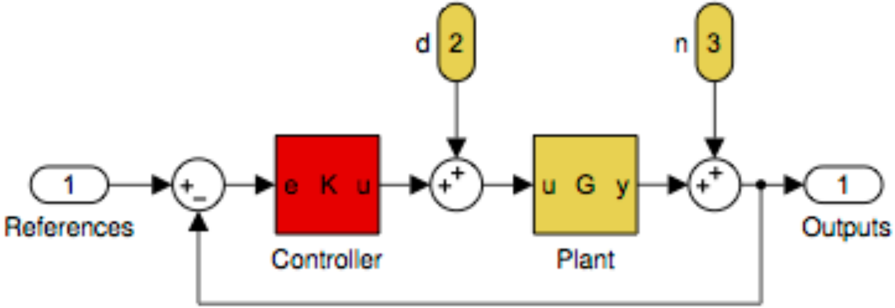
```
Matlab>>
s=tf('s'); G = 1/(s+1); K = 1;
sys_yr = feedback(G*K,1); sys_er = feedback(1,K*G);
sys_ur = feedback(K,G)
```

In a more realistic situation



we may have to deal with

- input noise (load disturbance) d
- output noise n
- model uncertainties (not treated here)



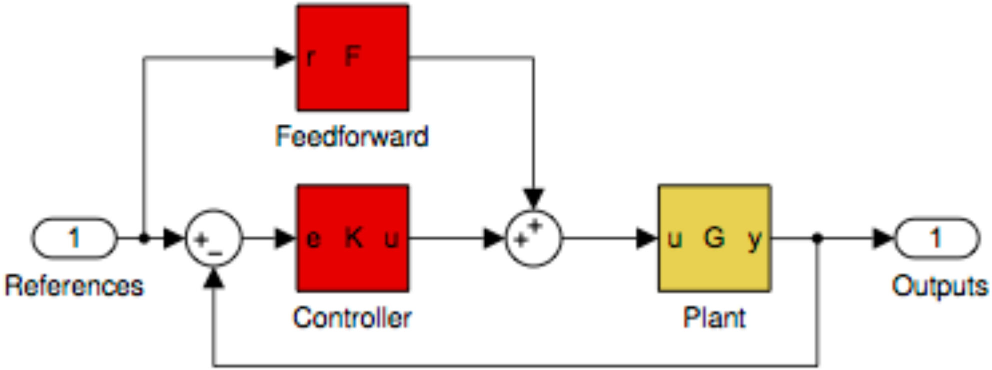
$$y = G(u+d)+n = G(K(r-y-n)+d)+n \Rightarrow$$

$$y = \frac{GK}{1+GK}r + \frac{G}{1+GK}d + \frac{1}{1+GK}n$$

Input noise rejection (d to y): $\dfrac{G}{1+GK} \to 0$

Output noise rejection (sensitivity function) (n to y): $\dfrac{1}{1+GK}$

Feedforward



Tracking error (r to e): $\dfrac{1-GF}{1+GK}$

Note that choosing $F \cong G^{-1}$ allows for similar tracking performance at smaller control gain K

Controller design - closed loop stability

How to chose K ? First priority: closed-loop stability!

Poles of closed-loop are roots of $1+G(s)K(s)$, i.e. values $p\in\mathbb{C}$ for which $G(s)K(s)=-1$. Ensure these are stable, i.e.. $\Re(p)<0$

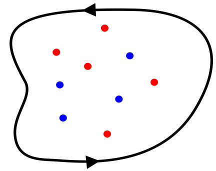
- *Routh-Hurwitz theorem (test in terms of polynomial coefficients)*
- OR *Matlab (21st century):*

```
>> s=tf('s'); G = (s-3)/((s+1)*(s+2)); % plant definition
>> K=1; sys_cl = 1/(1+G*K); all(real(pole(sys_cl))<0)
      false
>> K=0.5; sys_cl = 1/(1+G*K); all(real(pole(sys_cl))<0)
      true
```

- *Nyquist stability criterium*

Nyquist plot

We want the transfer function $\frac{1}{1+GK}$ to be stable, no poles with $\Re > 0$



Cauchy's argument principle :

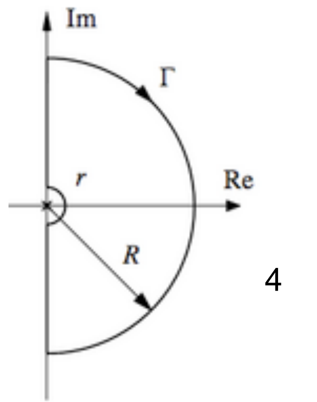
Let F(s) be a function $F : \mathbb{C} \rightarrow \mathbb{C}$

Let P / Z be number of poles/zeroes of $F(s)$ in the contour Γ

$Z - P = N$ where *N* is *winding number:*

(number of clockwise encirclements - number of counterclockwise encirclements) by $F(\Gamma)$ of 0

Nyquist contour: a contour Γ that captures RHP ($\Re > 0$)



Nyquist plot: locus $\left(\Re(F(i\omega)), \Im(F(i\omega))\right)$ $\omega \in \left]-\infty, \infty\right[$

Nyquist stability criterion

P : the number of RHP poles of $1+GK$, (= # RHP poles of GK)

Z : the number of RHP zeros of $1+GK$

We require $Z=0$ to guarantee stability of $\dfrac{1}{1+GK}$.

Winding number of the Nyquist plot of $1+GK$ around 0
(or that of GK around -1) : $N = Z - P = -P$

The closed loop transfer function $\dfrac{1}{1+GK}$ is stable (Z=0)
if the number of unstable poles of GK (P)
is equal to the negative of the winding number (N) of the Nyquist plot of GK around s=-1

Exercise: Nyquist plot

NB: The Nyquist criterium tells us about the stability of the closed-loop $\frac{1}{1+GK}$ by testing only a property of GK!

Exercise: Given $G = \frac{-(s-0.5)}{(s+1)(s+2)}$ how many windings should the Nyquist contour of GK have in order for the closed loop to be stable?

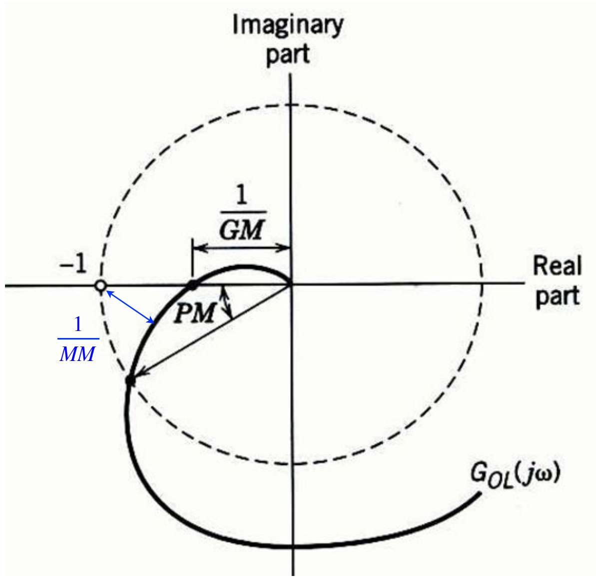
For which values of $K \in \Re$ is the closed-loop $\frac{1}{1+GK}$ stable?

Matlab tips:

- `s=tf('s'); G = -(s-0.5)/((s+2)*(s+1));`
- `K1=1; K2=-1; nyquist(G*K1,G*K2);`

Stability margin on the Nyquist plot

Assuming the closed loop is stable, we still want to *stay away* from encircling the -1 point!



Phase margin : $\angle GK + \pi$ at ω where $|GK| = 1$
i.e. by what amount can I change $\angle GK$ before instability?

Gain margin : $|GK|^{-1}$ at ω where $\angle GK = -\pi$
i.e. by what amount can I change $|GK|$ before instability?

Modulus margin : $\dfrac{1}{\min|1+GK|}$

Gain and phase margins

Bode plot $GK(i\omega)$:

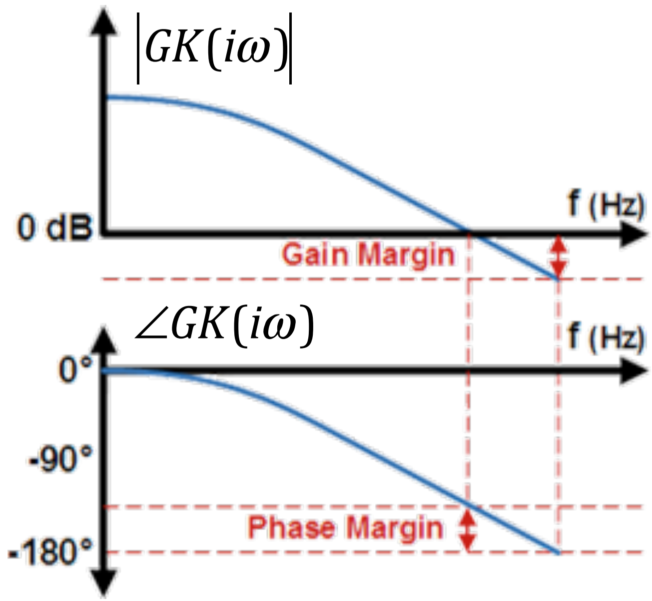
Gain margin: by how much can

- K the controller gain be increased
- G the plant model be wrong

before reaching $\angle GK = -\pi$

- Phase margin: by how much can
  - K the controller phase be increased (faster closed loop response)
  - G the plant model be wrong

before reaching $|GK| = 1$



Exercise - bode plot

Exercise: Given $G = \dfrac{-(s-0.5)}{(s+1)(s+2)}$ (same as previous exercises).

- What is the gain and phase margin for K=1?
- Relate this to the answer you found in the previous exercise.

Matlab tips:

- `s=tf('s'); G = -(s-0.5)/((s+2)*(s+1));`
- `K1=1; K2=-1; bode(G*K1,G*K2);`

Controller design - tradeoffs

- **Performance:** make transfer functions e.g. $d \rightarrow e$, $r \rightarrow e$ ‘small’ at up to some bandwidth frequency $\omega < \omega_b$.
- **Noise rejection:** make transfer functions $n \rightarrow e$, $n \rightarrow u$ ‘small’ at high frequency $\omega > \omega_b$
- **Robustness:** Maintain stability/performance even if the true plant is not exactly equal to G. e.g. via stability margins.

There is usually a tradeoff between performance and robustness. A ‘tame’ controller (low performance, low bandwidth ω_b) will work even if the plant model is very wrong. High performance control (high ω_b) requires good knowledge of model and uncertainties.

Tradeoff between these conflicting requirements: the ‘art’ of control engineering

PID controller

The PID (Proportional Integral Derivative) controller is a simple and old controller with

$$K = K_P + \frac{1}{s} K_I + s K_D = K_P \left(1 + \frac{1}{s T_I} + s T_D \right)$$

P term

Example: $G = \dfrac{1}{s-p} \Rightarrow 1 + GK = \dfrac{s - p + K_P}{s-p}, \; 1 + GK = 0 \Rightarrow p_0 = p - K_P$

- Stabilise an unstable plant ($p > 0$) if $p_0 < 0 \Rightarrow K_P > p$
- Make the closed loop response faster: $p_0 = p - K_P < p$

D term

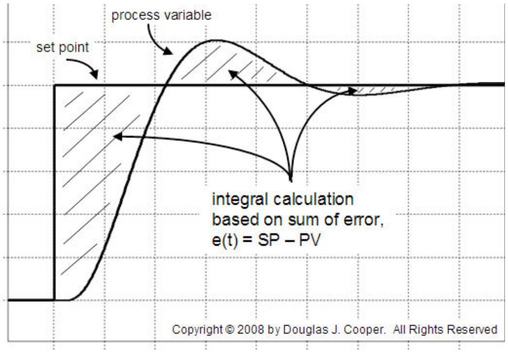
- Add a factor $(1+sT_D)$ to the open loop response GK_P, so reduce the phase lag by $\angle(1+i\omega T_D)\in[0,\pi]$ (if allowed by gain margin)

- Destabilises plants with pure delay

Example: $G = e^{-sL}\dfrac{1}{1+sT} \Rightarrow GK(i\omega) = K_P \dfrac{1+i\omega T_D}{1+i\omega T} e^{-i\omega L}$

D gain	$\lvert GK\rvert \le 1$	$\lvert GK(\omega_0)\rvert = 1$	$\angle GK(\omega_0)$	Stability
$T_D \le T$	$K_P \le 1$	$\omega_0 = 0$	0	stable
$T_D > T$	$T_D \le \dfrac{T}{K_P}$	$\omega_0 = \infty$	$-\infty$	no phase margin

I term



- Eliminate DC tracking error

For a P or PD controllers, the DC tracking error of the closed loop

$$\left.\frac{1}{1+GK}\right)_{s=0} = \frac{1}{1+G(s=0)K_P} > 0$$

With an I term $\frac{1}{sT_I}$: $K(s=0)=\infty \Rightarrow \left.\frac{1}{1+GK}\right)_{s=0} = 0$