control

Robótica Móvel e Inteligente / Mobile and Intelligent Robotics

Academic year 2020-21

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

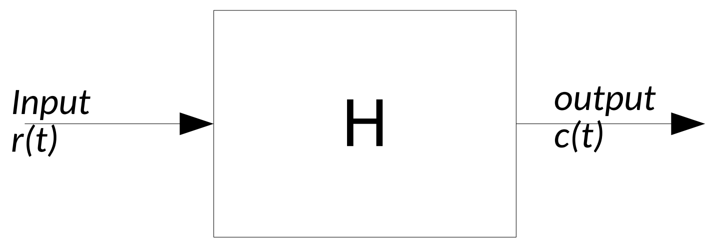
Control systems

- **Objective: to impose a given value of some physical quantity in a system by acting on some other physical quantity**

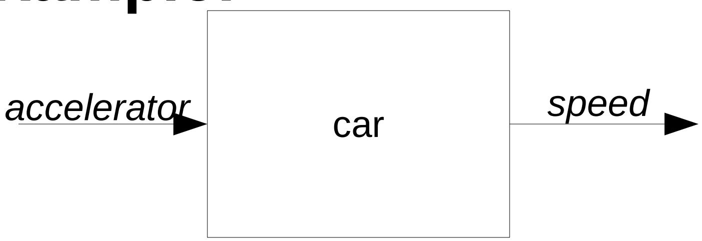
http://blog.caranddriver.com/nissan-develops-fully-electric-steer-by-wire-system-will-go-on-sale-next-year/

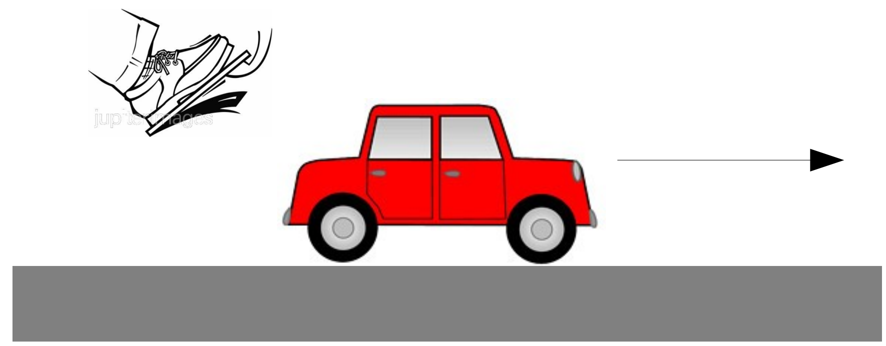
basic concepts

- **Systems approach:**
  - Input signal
  - Output signal
  - Process, transforming input into output
- **Objective: to impose a given value at a system's output, by acting in its input**



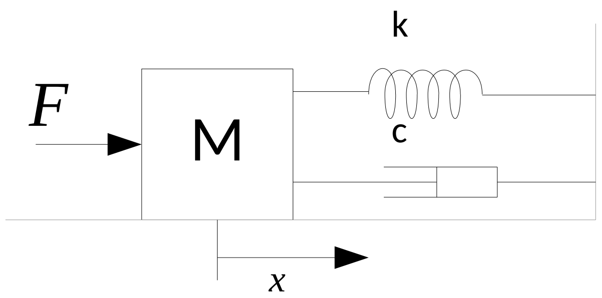
- **Example:**





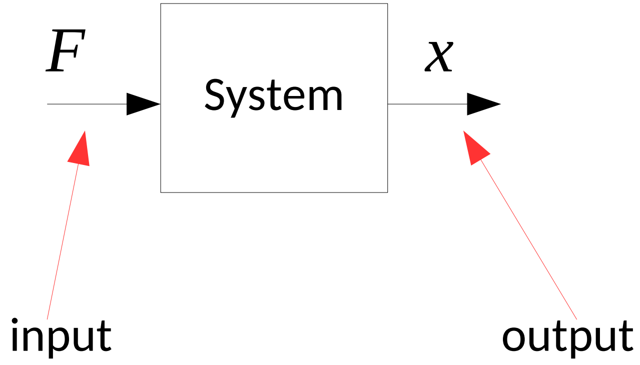
Input / output relationship

- **General case:**
  - input r*(t)* and output c*(t)* are related by differential equations
    - this is the “default” in physical systems...



Mathematical relation
between input and output

$$F(t) = M\frac{d^2x(t)}{dt^2} + c\frac{dx(t)}{dt} + k\,x(t)$$

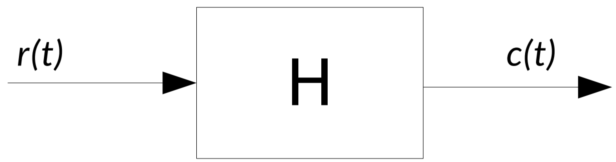


Input / output relationship

- **General case:**
  - input r(t) and output c(t) are related by differential equations

$$\underbrace{a_n \frac{d^n c(t)}{dt^n} + a_{n-1} \frac{d^{n-1} c(t)}{dt^{n-1}} + \ldots + a_0 c(t)}_{\text{Combination of } c(t) \text{ and its derivatives}} = \underbrace{b_m \frac{d^m r(t)}{dt^m} + b_{m-1} \frac{d^{m-1} r(t)}{dt^{m-1}} + \ldots + b_0 r(t)}_{\text{Combination of } r(t) \text{ and its derivatives}}$$

Difficult to solve and convert to a systems perspective

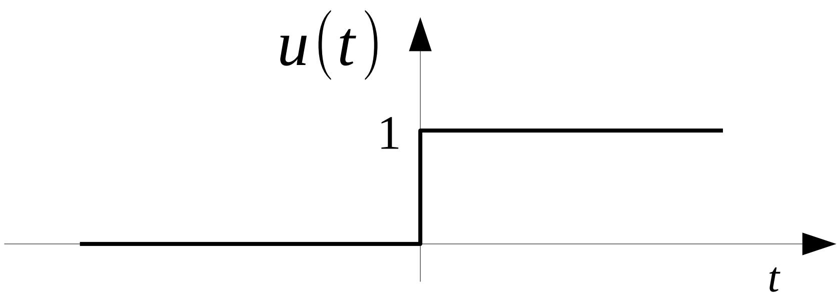


Laplace transform

- **Differential equations are simplified by the use of Laplace transforms.**

$$L\{f(t)\}=F(s)=\int_{0}^{+\infty} e^{-st} f(t)\,dt$$

$$L^{-1}\{F(s)\}=\frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} e^{st} F(s)\,dt=f(t)\cdot u(t)$$



u(t) is the unit step function. We only consider the function *f(t)* to have non-null values for $t>0$.

Computing the Laplace transform

- **Example for u(t)**

$$
\begin{aligned}
L\{u(t)\} &= \int_{0}^{+\infty} e^{-st} u(t)\, dt \\
&= \int_{0}^{+\infty} e^{-st}\, dt \\
&= \left[-\frac{1}{s} e^{-st} \right]_{0}^{+\infty} \\
&= 0 - \left(-\frac{1}{s} \right) \\
&= \frac{1}{s}
\end{aligned}
$$

Laplace transform

TABLE 2.1 Laplace transform table

Item no.	$f(t)$	$F(s)$
1.	$\delta(t)$	1
2.	$u(t)$	$\frac{1}{s}$
3.	$tu(t)$	$\frac{1}{s^2}$
4.	$t^n u(t)$	$\frac{n!}{s^n + 1}$
5.	$e^{-at}u(t)$	$\frac{1}{s+a}$
6.	$\sin \omega t u(t)$	$\frac{\omega}{s^2+\omega^2}$
7.	$\cos \omega t u(t)$	$\frac{s}{s^2+\omega^2}$

Laplace transform

Laplace transform theorems

Theorem	Name
$\mathcal{L}[f(t)] = F(s) = \int_{0-}^{\infty} f(t)e^{-st}dt$	Definition
$\mathcal{L}[kf(t)] = kF(s)$	Linearity theorem
$\mathcal{L}[f_1(t) + f_2(t)] = F_1(s) + F_2(s)$	Linearity theorem
$\mathcal{L}[e^{-at}f(t)] = F(s+a)$	Frequency shift theorem
$\mathcal{L}[f(t-T)] = e^{-sT}F(s)$	Time shift theorem
$\mathcal{L}[f(at)] = \frac{1}{a}F\left(\frac{s}{a}\right)$	Scaling theorem
$\mathcal{L}\left[\frac{df}{dt}\right] = sF(s) - f(0-)$	Differentiation theorem
$\mathcal{L}\left[\frac{d^2f}{dt^2}\right] = s^2F(s) - sf(0-) - f'(0-)$	Differentiation theorem
$\mathcal{L}\left[\frac{d^nf}{dt^n}\right] = s^nF(s) - \sum_{k=1}^{n} s^{n-k}f^{k-1}(0-)$	Differentiation theorem
$\mathcal{L}\left[\int_{0-}^{t} f(\tau)d\tau\right] = \frac{F(s)}{s}$	Integration theorem
$f(\infty) = \lim_{s\to 0} sF(s)$	Final value theorem[1]
$f(0+) = \lim_{s\to\infty} sF(s)$	Initial value theorem[2]

Laplace transform

$$a_n \frac{d^n c(t)}{dt^n} + a_{n-1} \frac{d^{n-1} c(t)}{dt^{n-1}} + \ldots + a_0 c(t) = b_m \frac{d^m r(t)}{dt^m} + b_{m-1} \frac{d^{m-1} r(t)}{dt^{m-1}} + \ldots + b_0 r(t)$$

Computing the
Laplace transform

$$\Rightarrow \quad L\left\{a_n \frac{d^n c(t)}{dt^n}\right\} = a_n L\left\{\frac{d^n c(t)}{dt^n}\right\} \quad \text{Linearity}$$

$$= a_n s^n L\{c(t)\} \quad \text{Differentiation theorem}$$

$$= a_n s^n C(s) \quad \text{Definition}$$

Laplace transform

$$a_n \frac{d^n c(t)}{dt^n} + a_{n-1} \frac{d^{n-1} c(t)}{dt^{n-1}} + \ldots + a_0 c(t) = b_m \frac{d^m r(t)}{dt^m} + b_{m-1} \frac{d^{m-1} r(t)}{dt^{m-1}} + \ldots + b_0 r(t)$$

$$\Downarrow$$

$$a_n s^n C(s) + a_{n-1} s^{n-1} C(s) + \ldots + a_0 C(s) = b_m s^m R(s) + b_{m-1} s^{m-1} R(s) + \ldots + b_0 R(s)$$

$$\left(a_n s^n + a_{n-1} s^{n-1} + \ldots + a_0\right) C(s) = \left(b_m s^m + b_{m-1} s^{m-1} + \ldots + b_0\right) R(s)$$

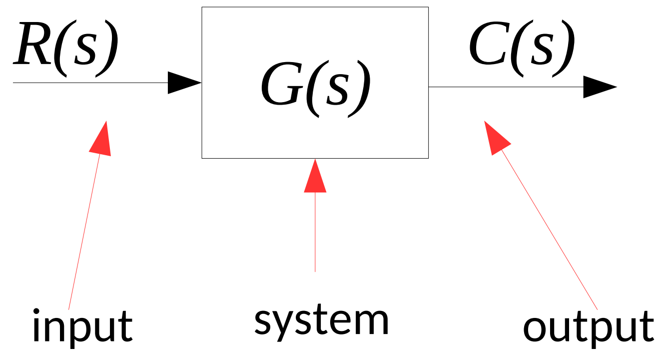
$$\frac{C(s)}{R(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_0} = G(s)$$

Transfer function

$$\frac{C(s)}{R(s)}=\frac{b_m s^m+b_{m-1}s^{m-1}+\ldots+b_0}{a_n s^n+a_{n-1}s^{n-1}+\ldots+a_0}=G(s)$$

Transfer function

$$C(s)=G(s)\cdot R(s)$$



- A relation expressed originally in terms of a differential equation is expressed as a product
- the physical nature of input/output relationship is irrelevant; only mathematical relationship matters --> **abstraction**

Transfer function

$$C(s) = G(s) \cdot R(s)$$

$$R(s) \longrightarrow \boxed{G(s)} \longrightarrow C(s)$$

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_0}$$

$$= \frac{b_m (s - z_1)(s - z_2) \ldots (s - z_m)}{a_n (s - p_1)(s - p_2) \ldots (s - p_n)}$$

z_1, z_2, ... Zeroes of $G(s)$

p_1, p_2, ... Poles of $G(s)$

Nearly all information about the system behaviour can be extracted from knowing the zeroes and poles.

Poles and zeroes are either:
- Real valued
- Complex conjugate

(meaning that the b_i and a_j are all real valued)

In "real life" systems

Partial fraction expansion

All poles of *F(s)* are real and distinct

$$F(s)=\frac{N(s)}{D(s)}=\frac{N(s)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

$$=\frac{K_1}{(s-p_1)}+\frac{K_2}{(s-p_2)}+\cdots+\frac{K_n}{(s-p_n)}$$

where $K_i=\lim_{s\to p_i}(s-p_i)F(s)$

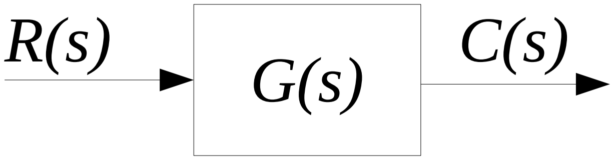
Assumption: order of $N(s)$ is smaller than the order of $D(s)$

Equivalent procedures exist when:
a) poles are real and repeated
b) poles are complex conjugated

Computation hint:
You can use Apart function in Wolfram Alpha
or the Partial Fraction Calculator widget

Nise, Norman S. 2011. Control Systems Engineering. 6. ed., International student version. Hoboken, NJ: Wiley.

Example



$$r(t)=u(t)$$

$$G(s)=\frac{1}{s+4}$$

$$c(t)=?$$

$$R(s)=\frac{1}{s}$$

$$C(s)=G(s)\cdot R(s)$$

$$=\frac{1}{s+4}\cdot\frac{1}{s}$$

Inverse L.T.

$$c(t)=\frac{1}{4}-\frac{1}{4}e^{-4t}$$

Corresponds to the pole s=0.
Related to the input signal.
Forced response

Corresponds to the pole s=-4.
Related to the system's response
If system is stable, it will decay with time.
Natural response

Stability

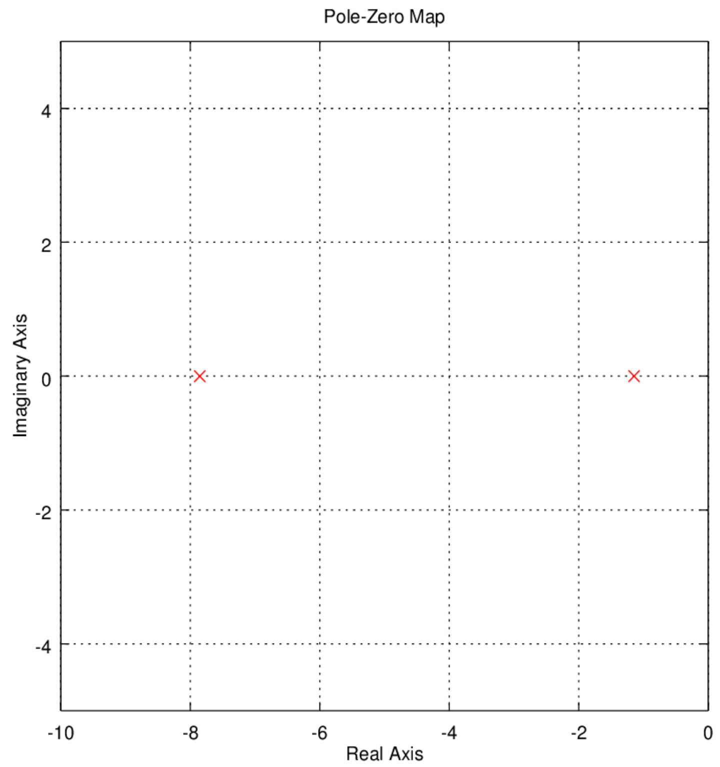
$$F_i(s) = \frac{K_i}{(s - p_i)}$$

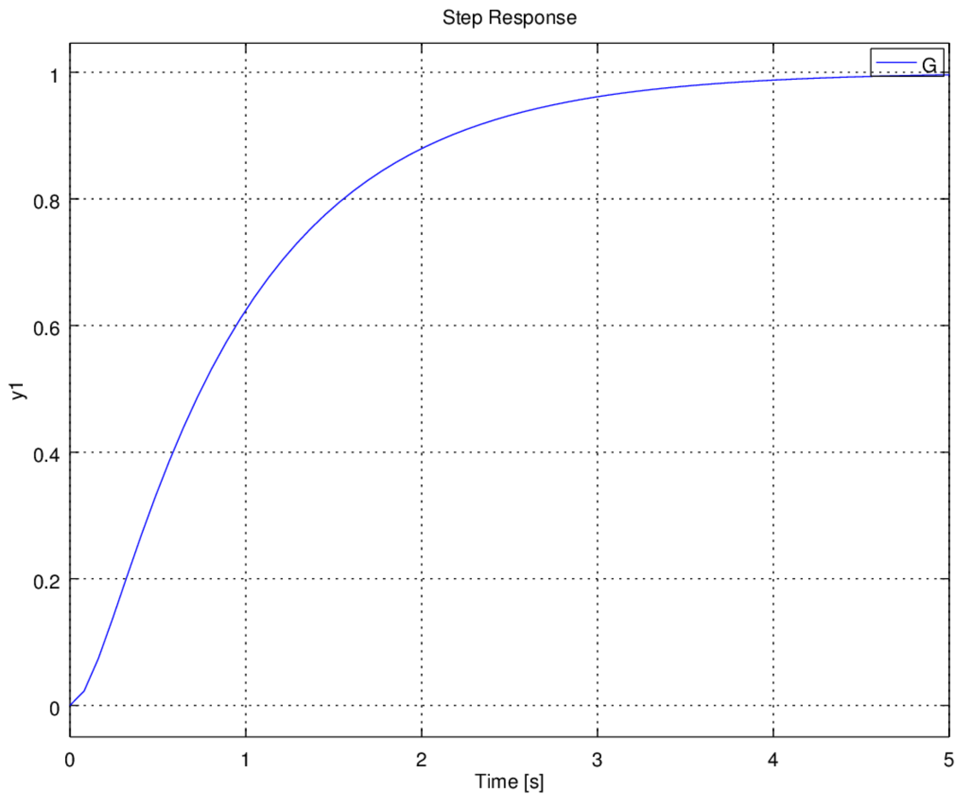
$$f_i(t) = K_i e^{p_i t}$$

Poles and step response

$$G_1(s)=\frac{9}{s^2+9s+9}$$

Overdamped

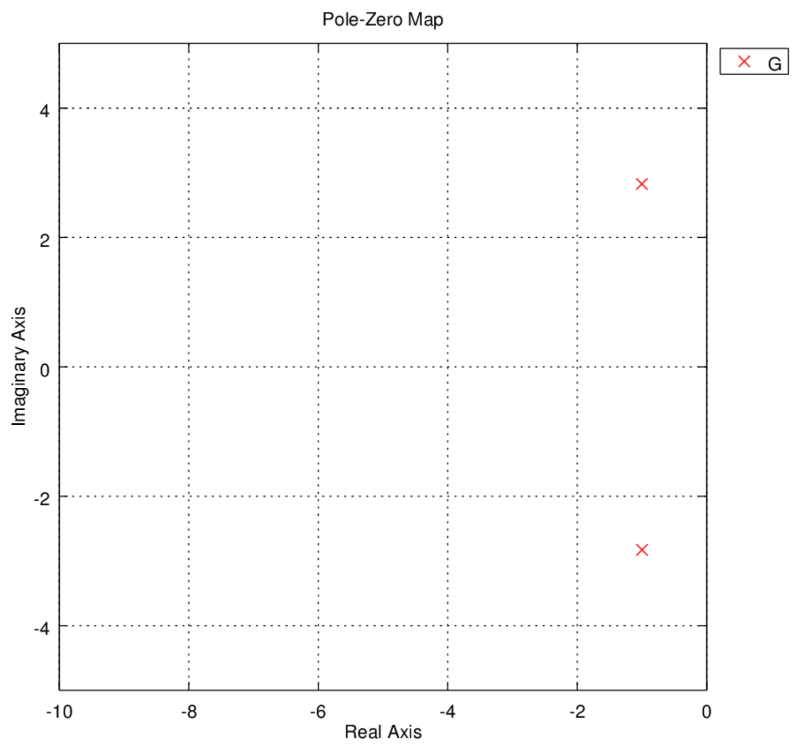


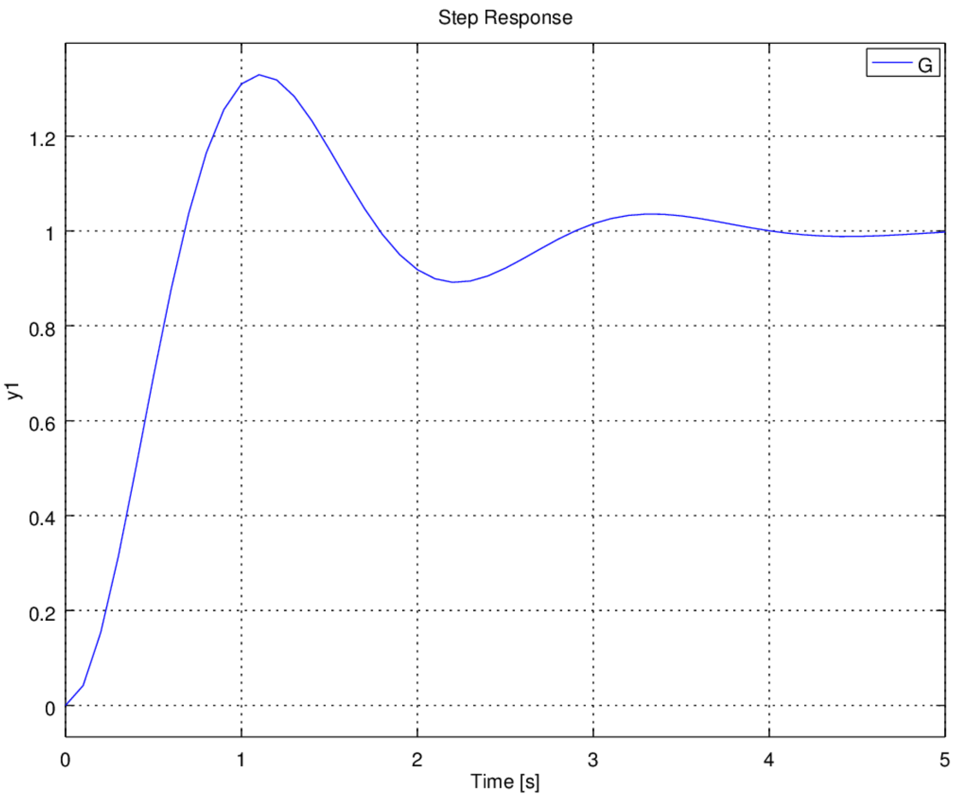


Poles and step response

$$G_2(s) = \frac{9}{s^2 + 2s + 9}$$

Underdamped

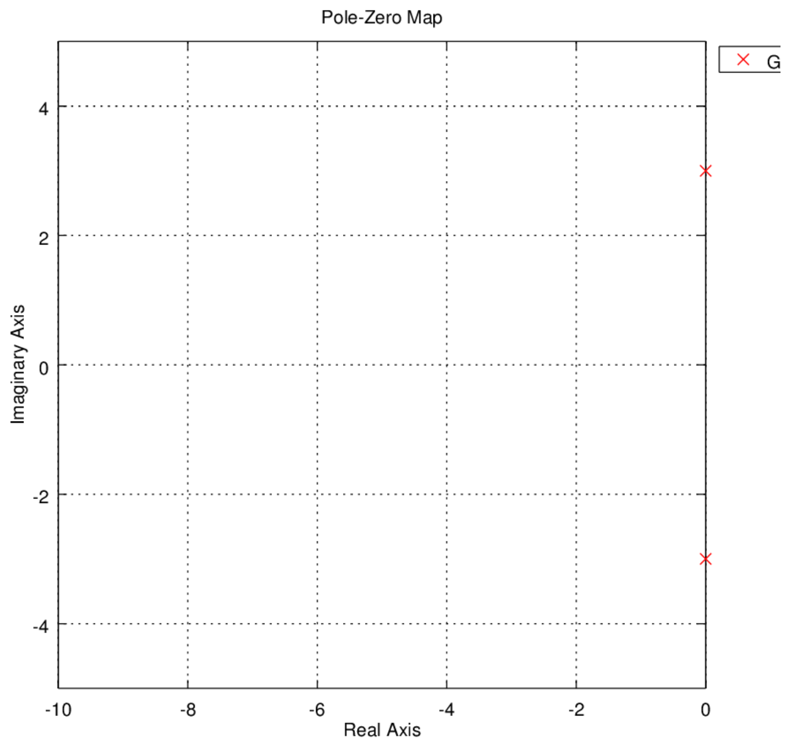


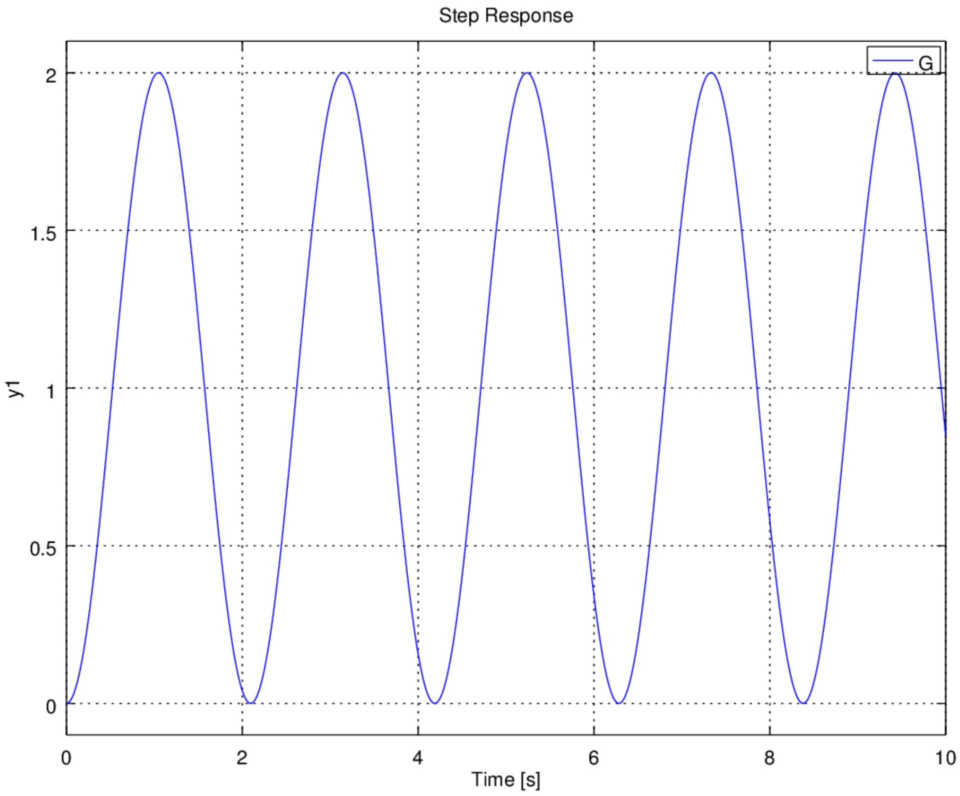


Poles and step response

Undamped

$$G_3(s)=\frac{9}{s^2+9}$$

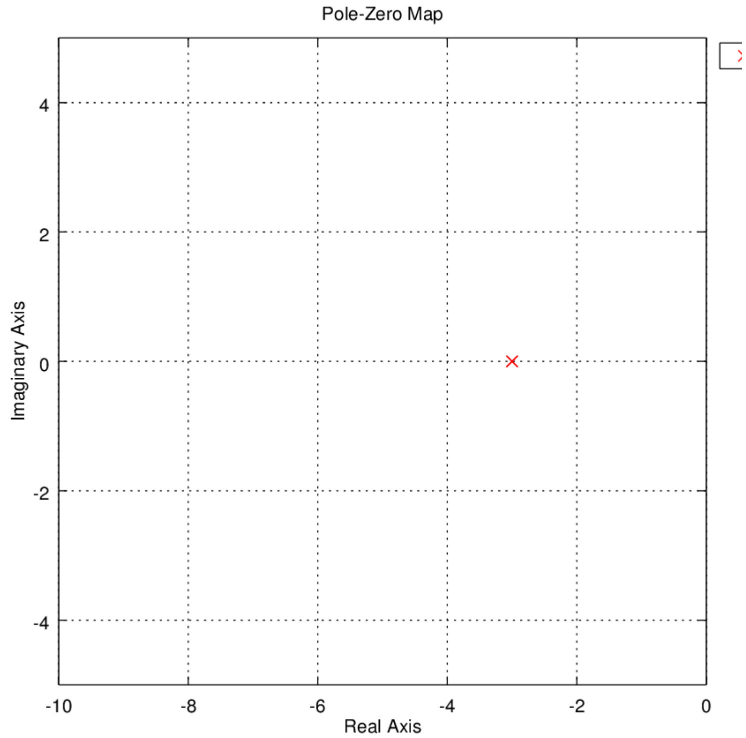


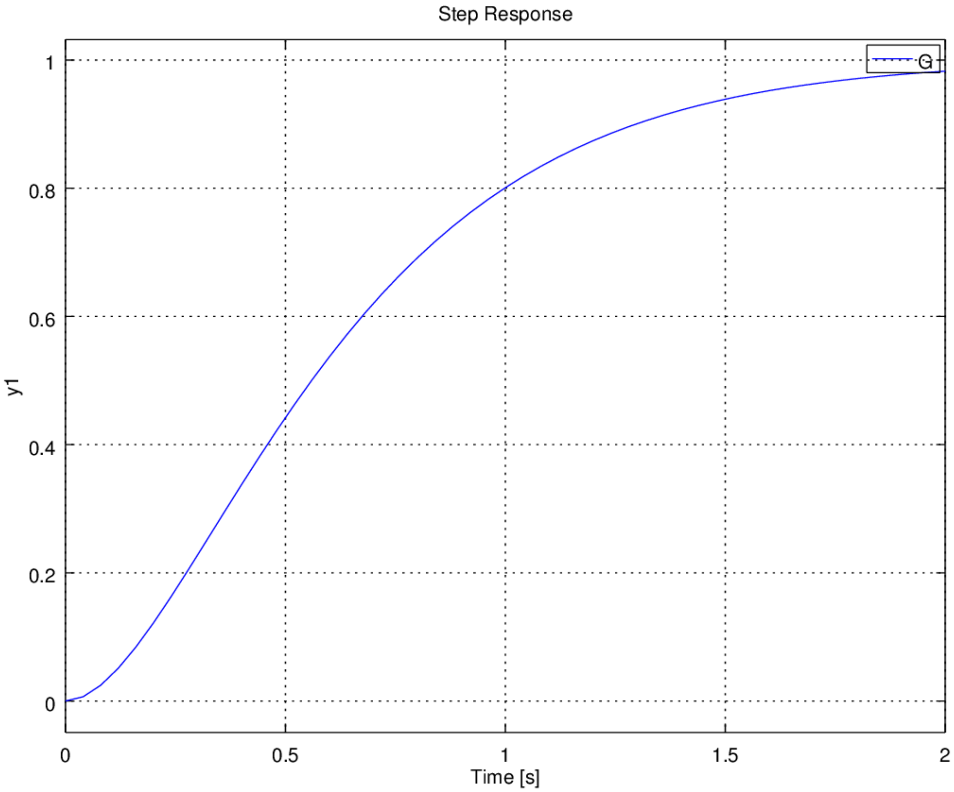


Poles and step response

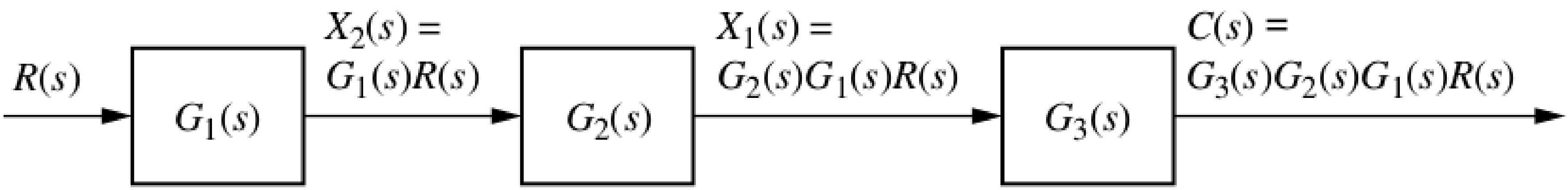
$$G_4(s)=\frac{9}{s^2+6s+9}$$

Critically damped

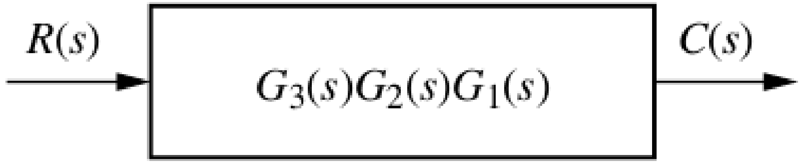




Block diagram algebra

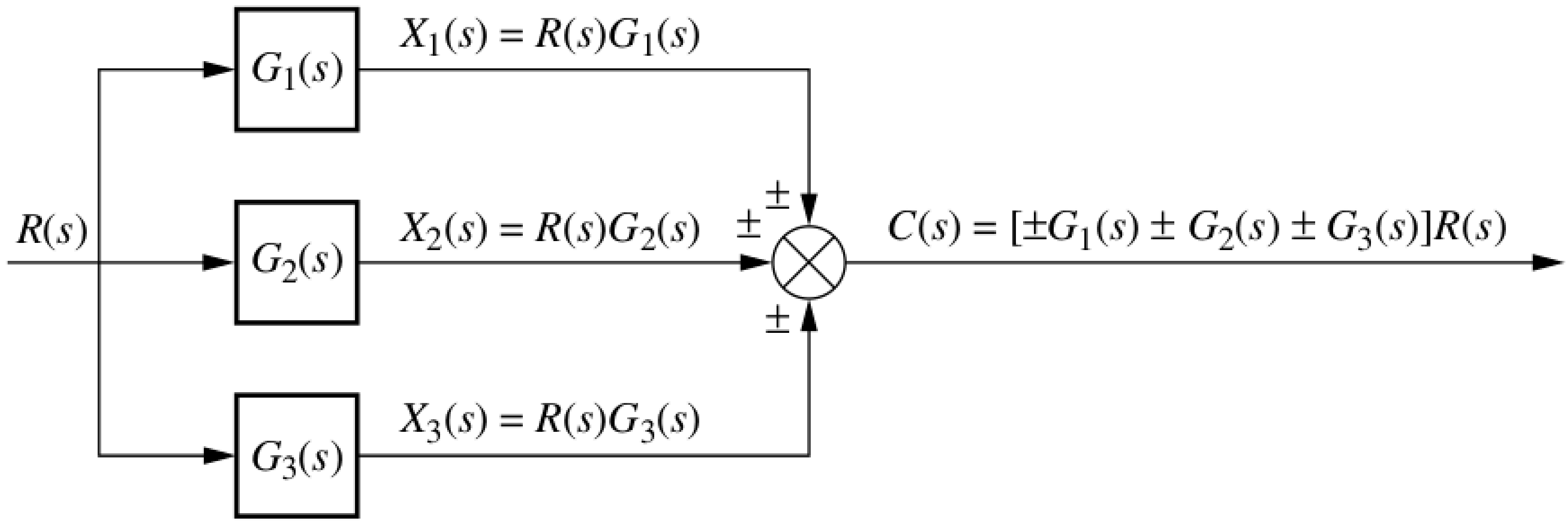


(*a*)

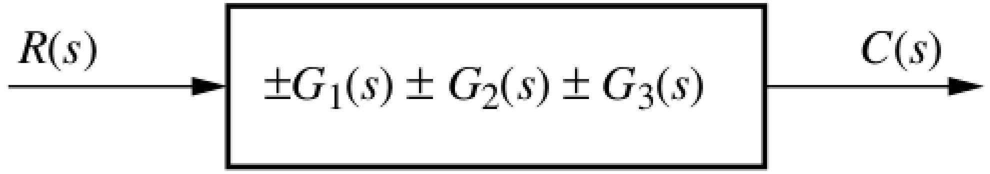


(*b*)

Block diagram algebra

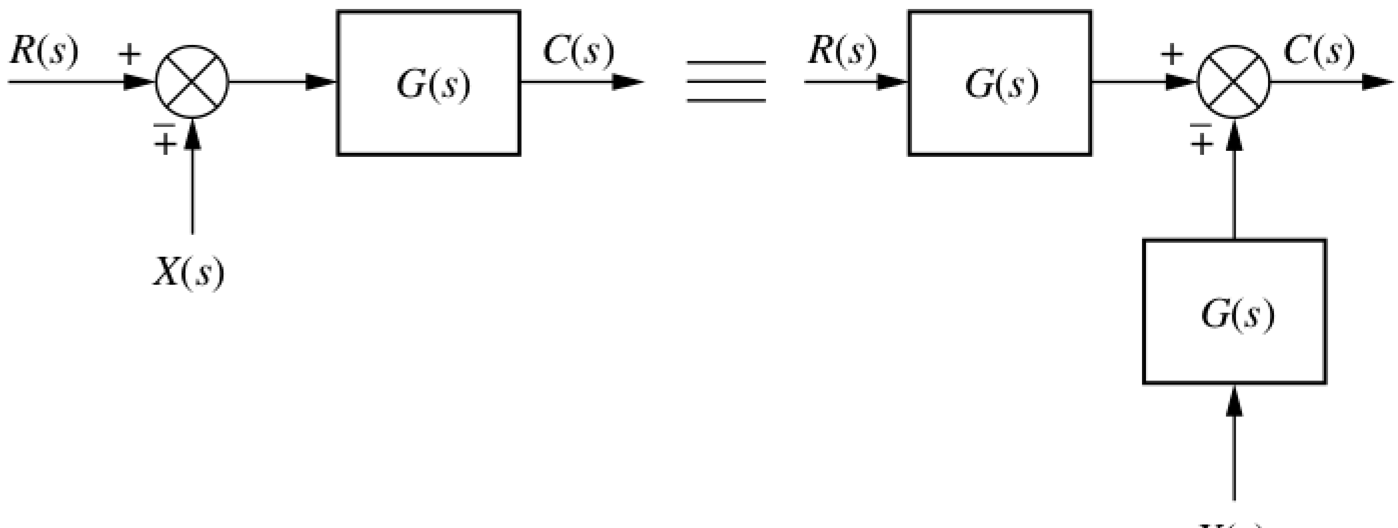


(*a*)

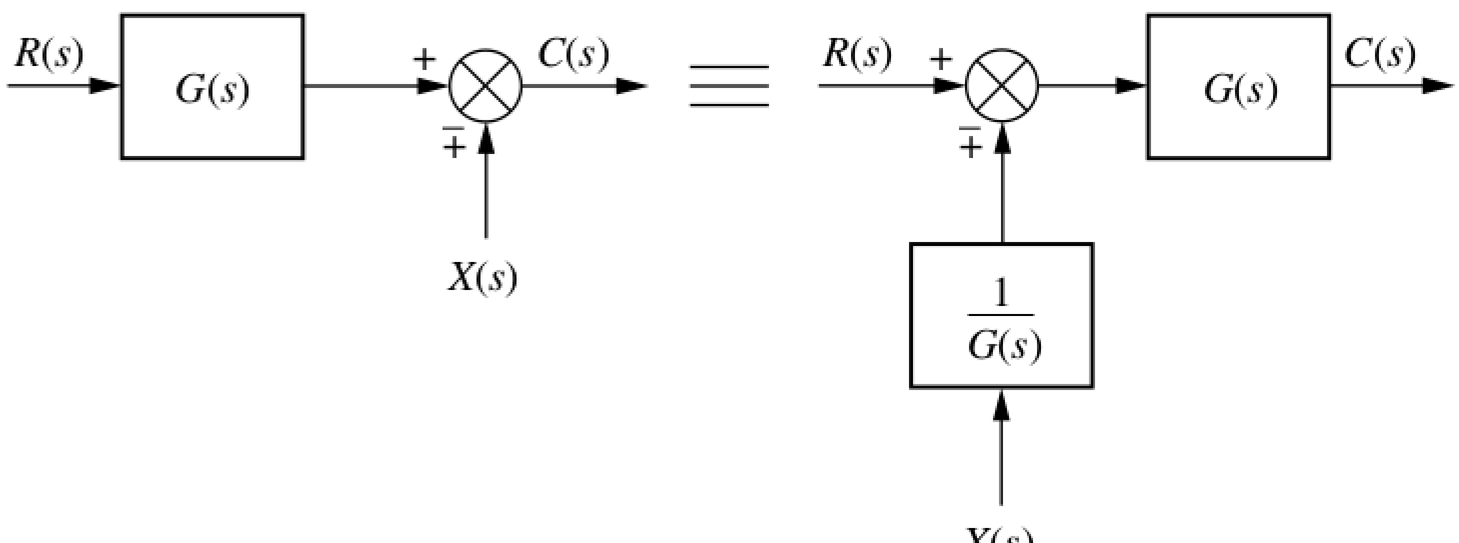


(*b*)

Block diagram algebra

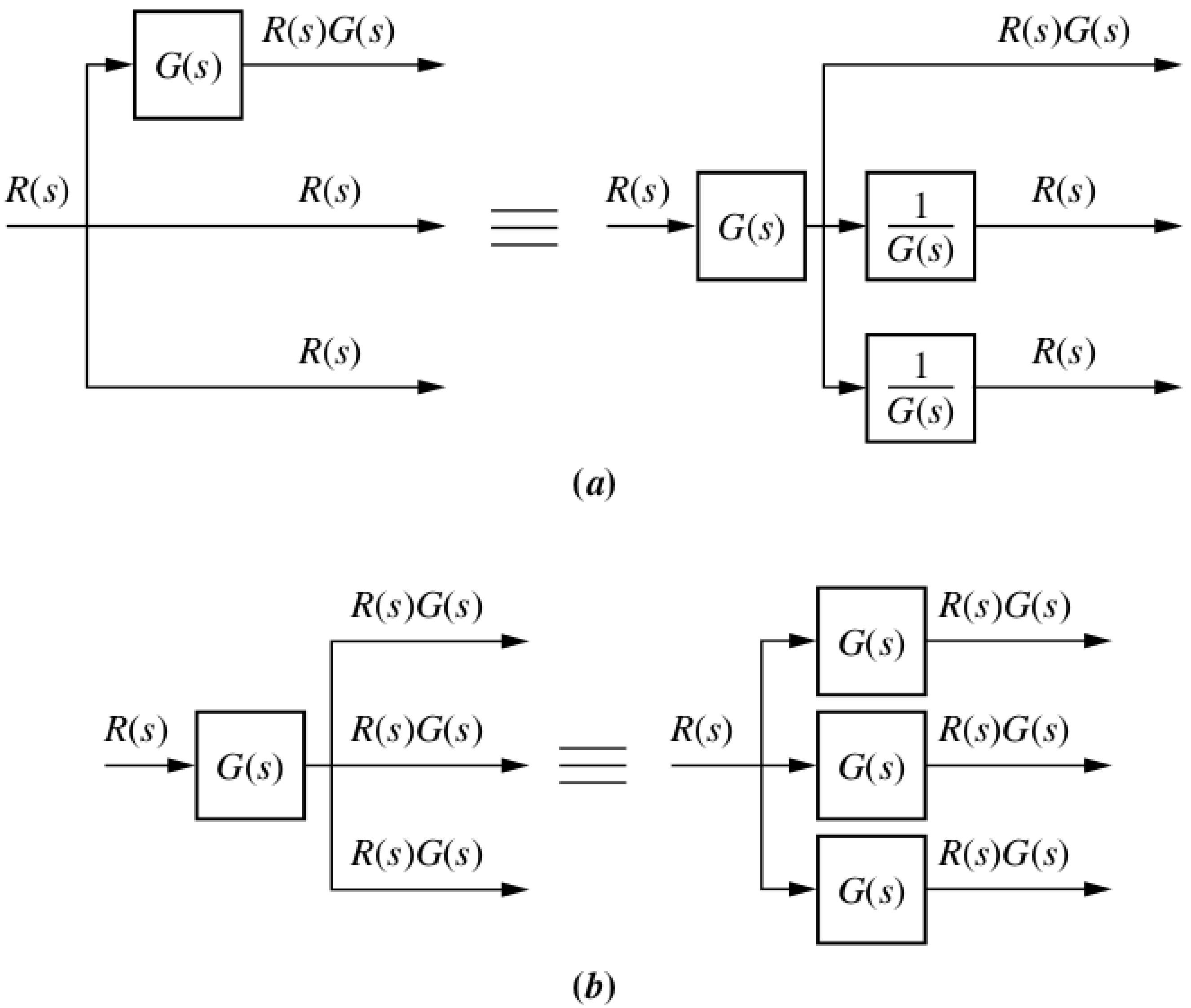


(a)



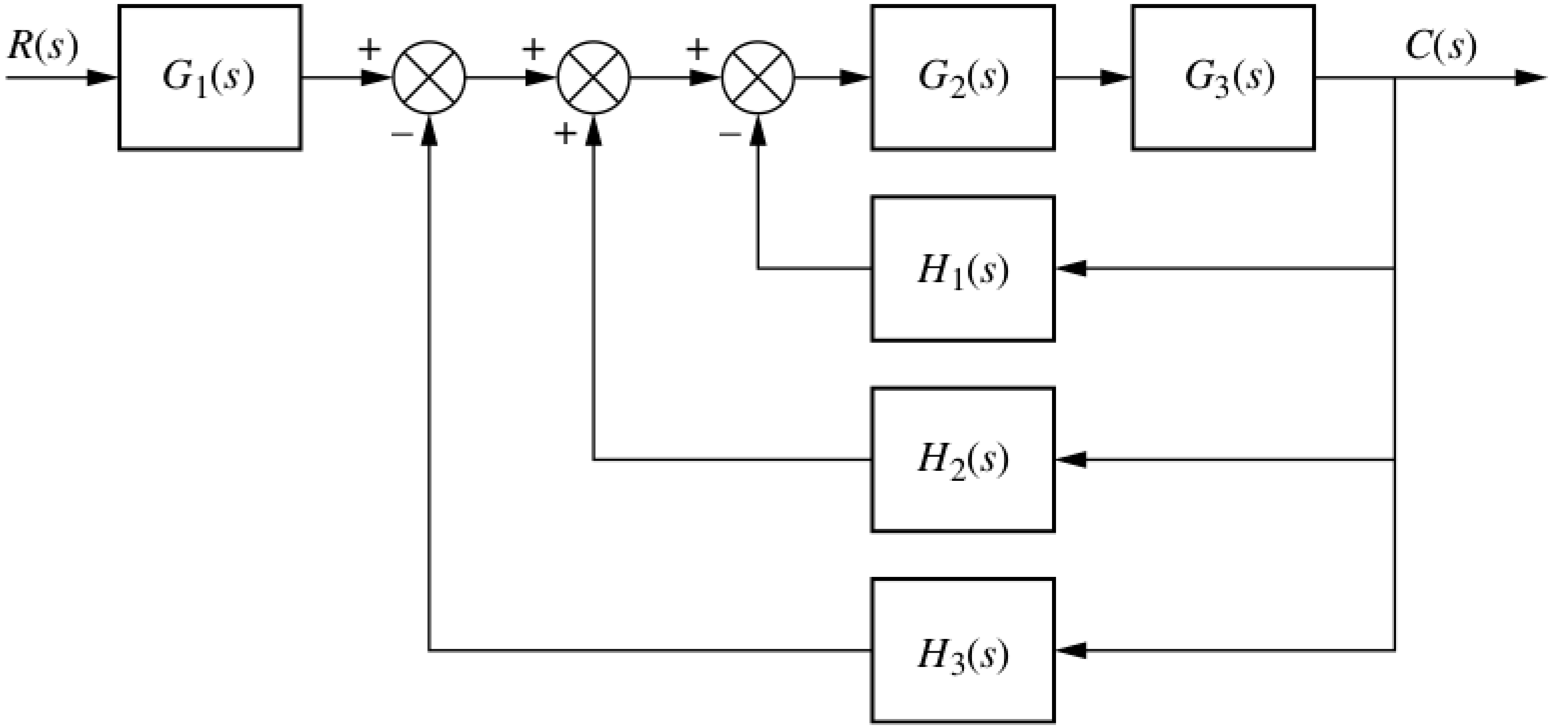
(b)

Block diagram algebra



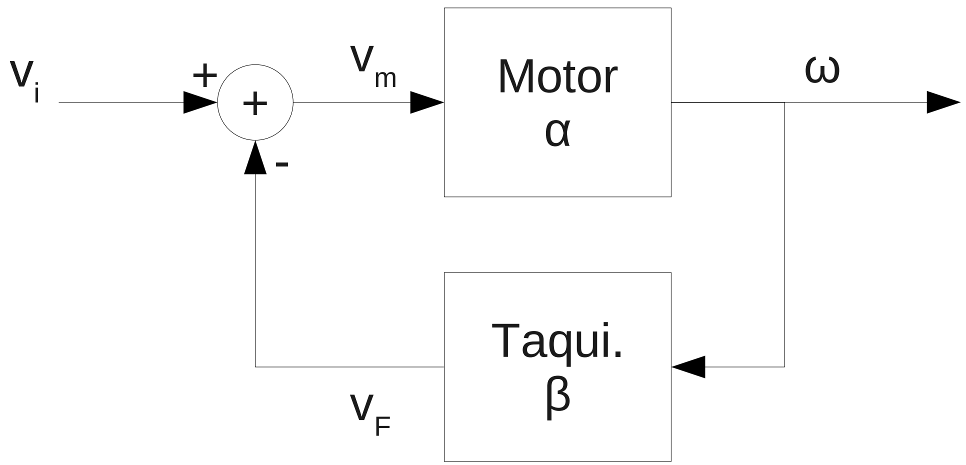
Block diagram algebra

PROBLEM: Reduce the block diagram shown in Figure 5.9 to a single transfer function.

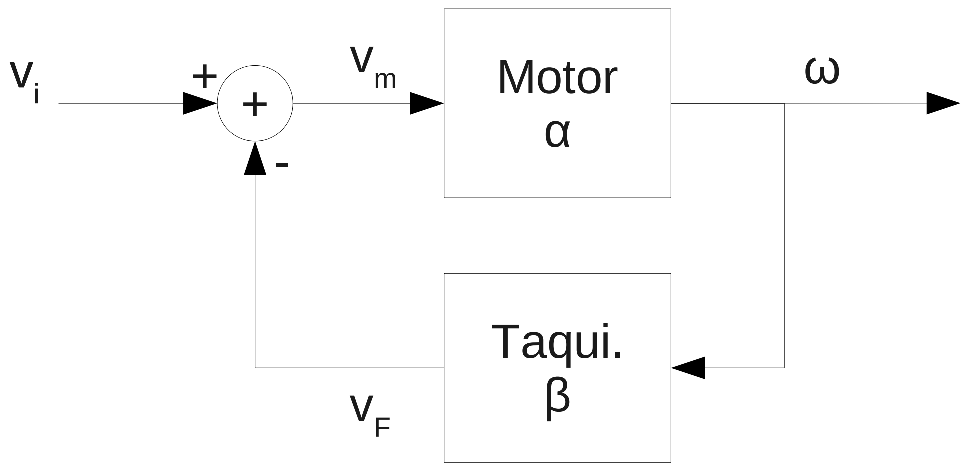


feedback

- **a complex system is represented as a collection of interconnected set of simpler systems**
  - each simple system has a known transfer function



feedback

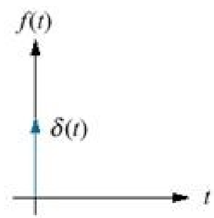
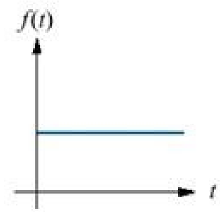
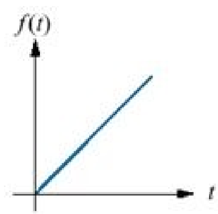
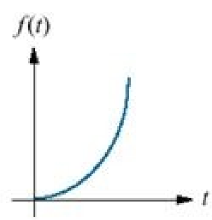
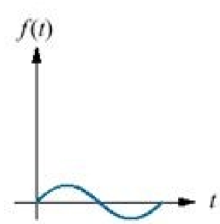


$$\begin{cases} v_m = v_i - v_T \\ \omega = \alpha\, v_m \\ v_F = \beta\, \omega \end{cases} \Rightarrow \omega = \frac{\alpha}{1+\alpha\beta} v_i$$

$$\lim_{\alpha \to \infty} \frac{\alpha}{1+\alpha\beta} = \frac{1}{\beta}$$

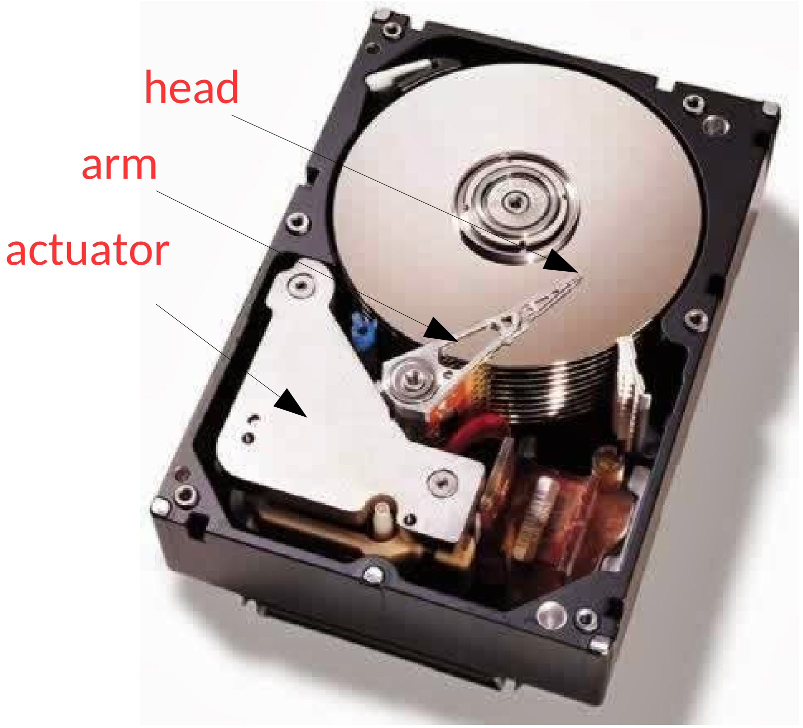
- **for high values of α, the output value will depend mainly on the *feedback* loop**

Test waveforms

Input	Function	Description	Sketch	Use
Impulse	$\delta(t)$	$\delta(t) = \infty$ for $0- < t < 0+$ $= 0$ elsewhere $\int_{0-}^{0+} \delta(t)\,dt = 1$		Transient response Modeling
Step	$u(t)$	$u(t) = 1$ for $t > 0$ $= 0$ for $t < 0$		Transient response Steady-state error
Ramp	$tu(t)$	$tu(t) = t$ for $t \geq 0$ $= 0$ elsewhere		Steady-state error
Parabola	$\frac{1}{2}t^2u(t)$	$\frac{1}{2}t^2u(t) = \frac{1}{2}t^2$ for $t \geq 0$ $= 0$ elsewhere		Steady-state error
Sinusoid	$\sin \omega t$			Transient response Modeling Steady-state error

Example of a control system

- **Objective: to impose a given value at a system's output, by acting in its input**



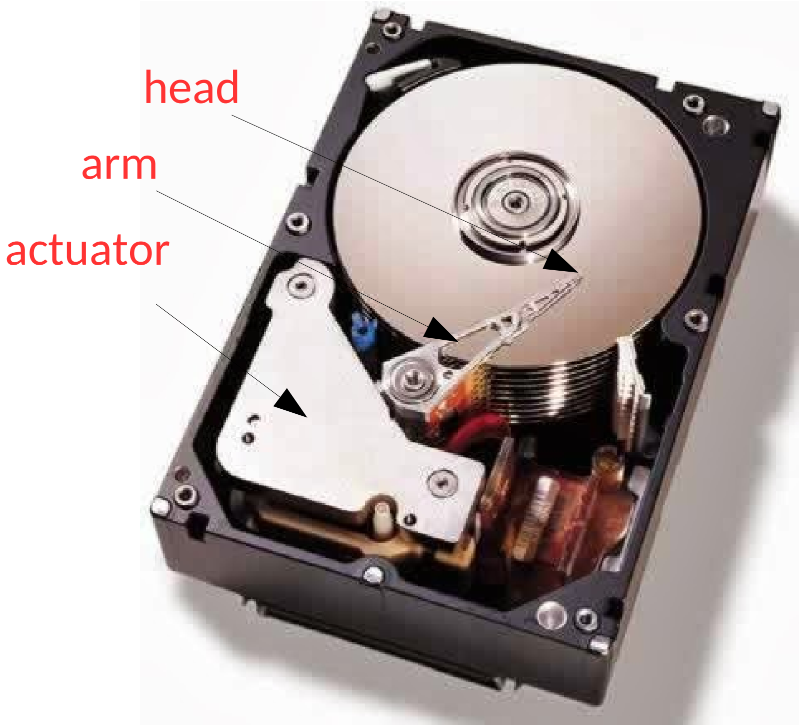
The head position will be determined by a voltage applied to the arm actuator.

Usually, we are interested in having a reference signal $r(t)$ that has a simple relation to the controlled variable $c(t)$ (proportional, if possible...)

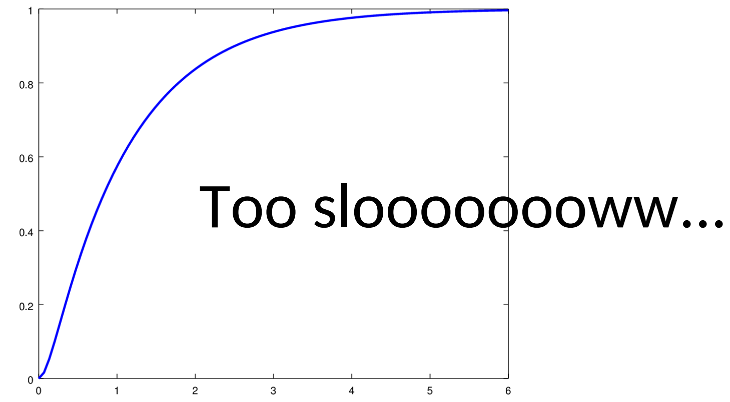
http://www.bsierad.com/assembling-process-and-function-hdd-hard-drive-parts/

Example of a control system

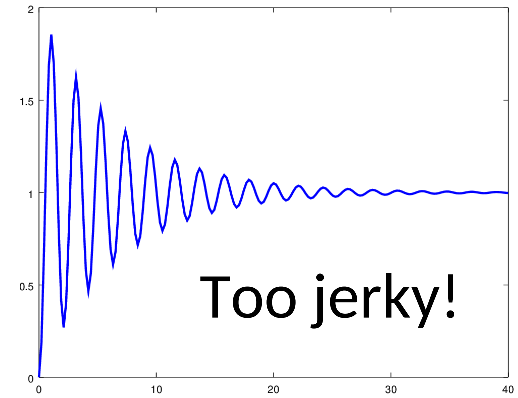
- **Objective: to impose a given value at a system's output, by acting in its input**



It is not good if the system reacts like this:



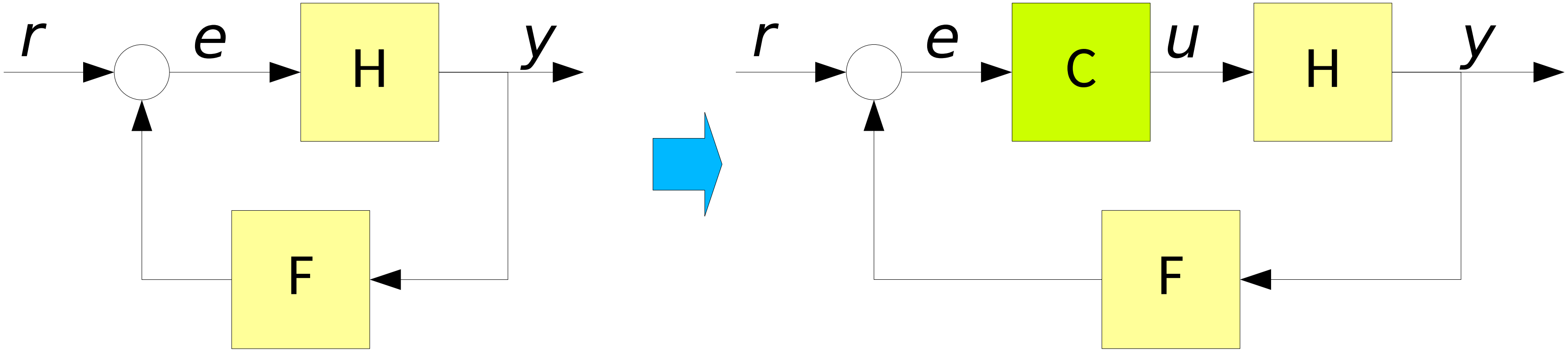
Or like this:



In both cases, it takes too long to reach the desired position.

controller

- **Controller C: included to improve the system response**



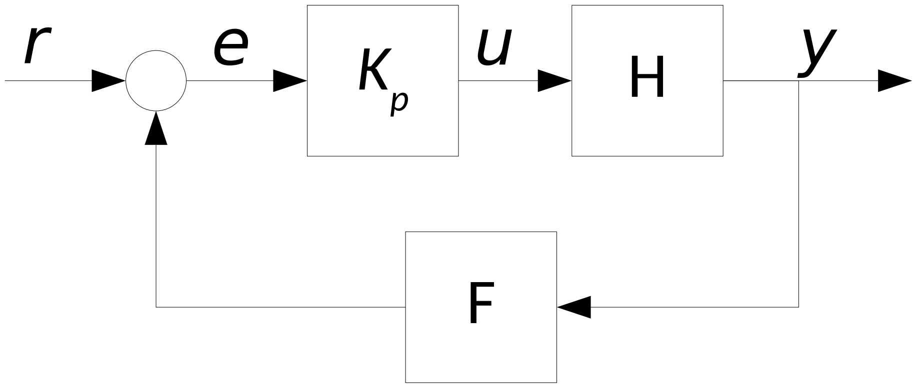
e: error

P controller

- **P = “proportional”. Simplest form of controller**

  $u = K_p \, e$

- **Gain: K_p**



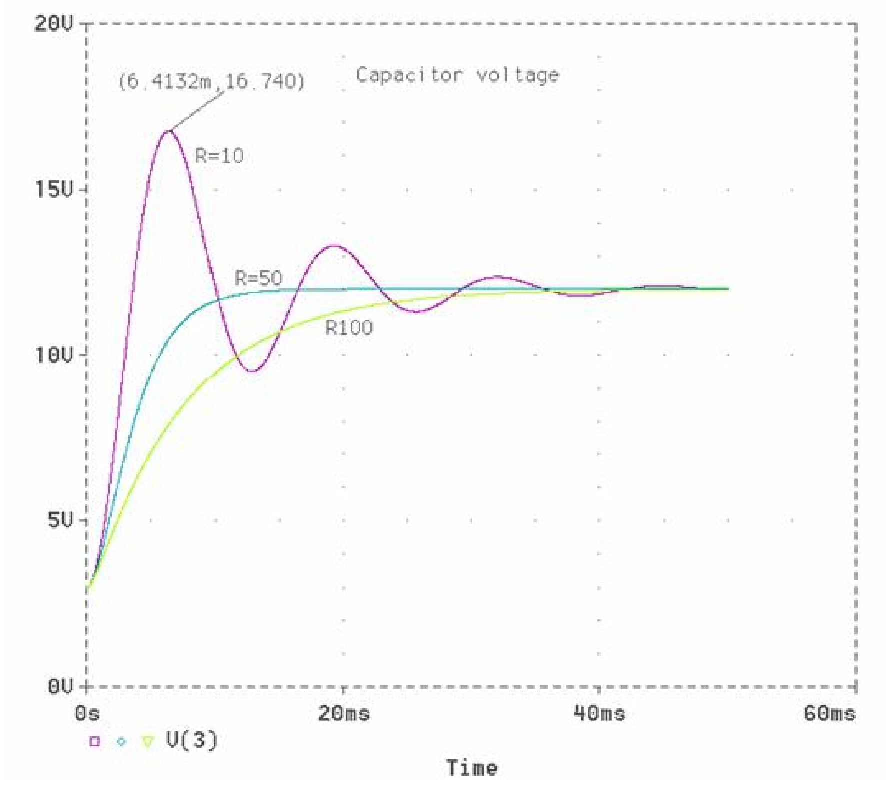
The larger K_p, the smaller *e* for the same output.

- Kp low: soft system
  - it takes a large value of error for system to react
- Kp high: hard system
  - strong reaction, even with small values of *e*.

$u \neq 0$ iif $e \neq 0$

To reduce error, a high value of Kp is required

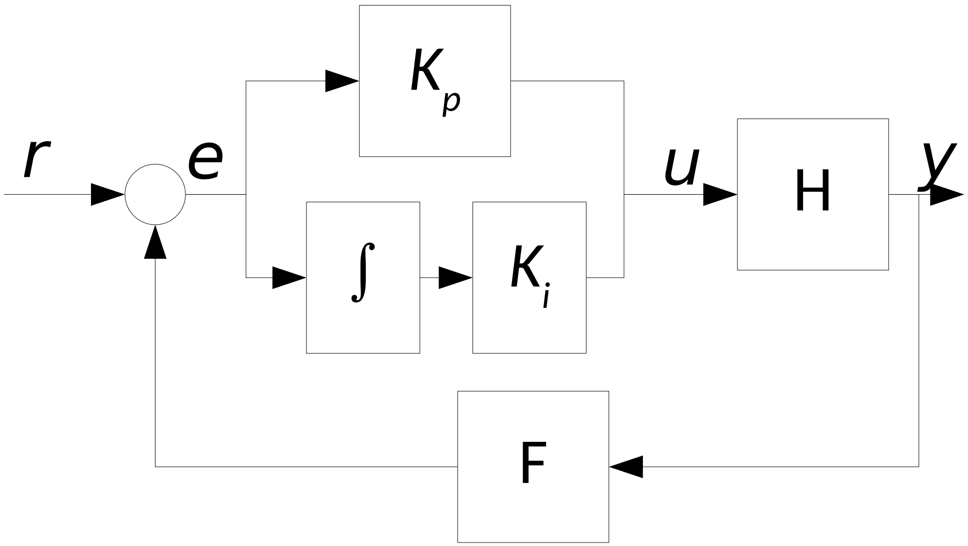
P controller



- **Increasing gain Kp reduces error, but...**
- **High values of gain Kp may cause the system to be unstable**

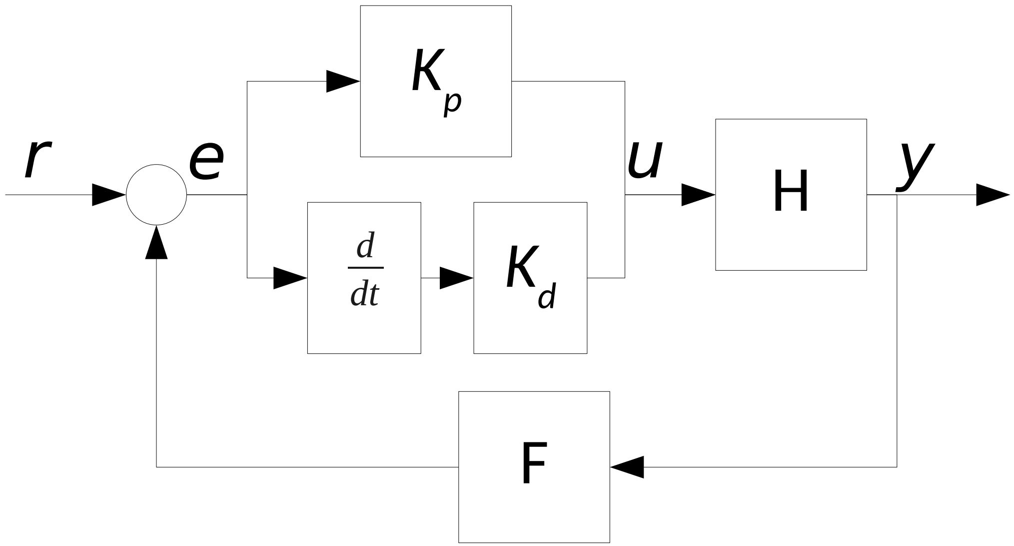
PI controller

- **PI: Proportional + Integral**
  - Adding a term $K_i \int e\, dt$.
  - PI controller allows for systems with e=0
  - Problem: inertia (memory effect)
    - with rapid changes in the input, u may be at a value when the error e would require to be different



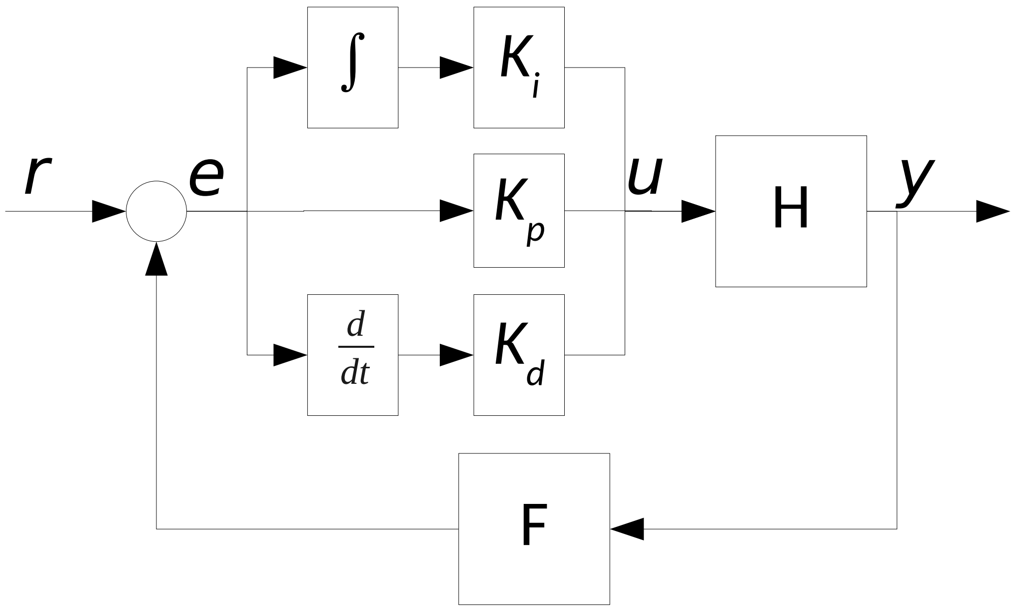
PD controller

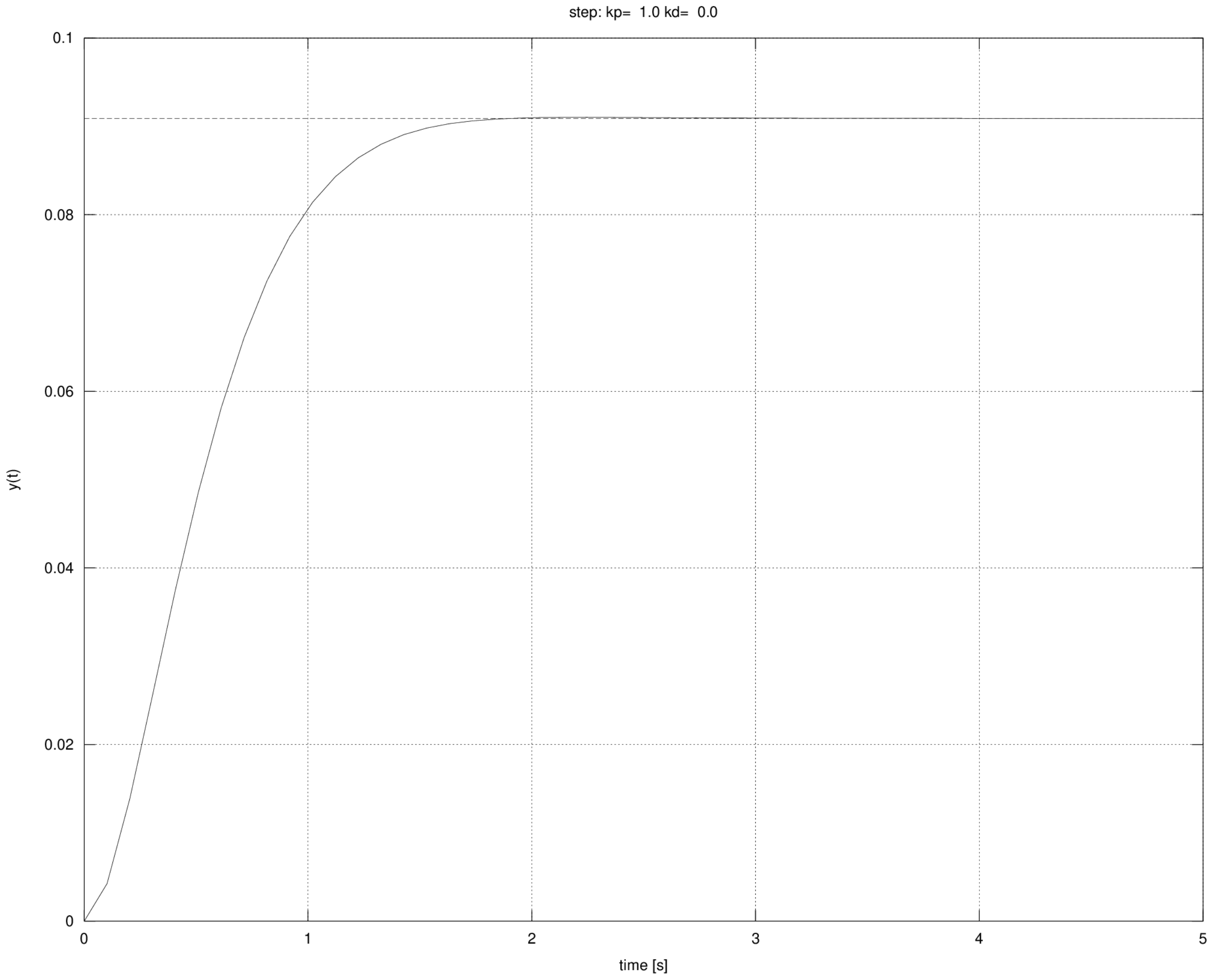
- **PD = Proportional + Derivative**
  - adding a term proportional to the error derivative
  - In general, it has the effect of reducing oscillations (damper)



PID controller

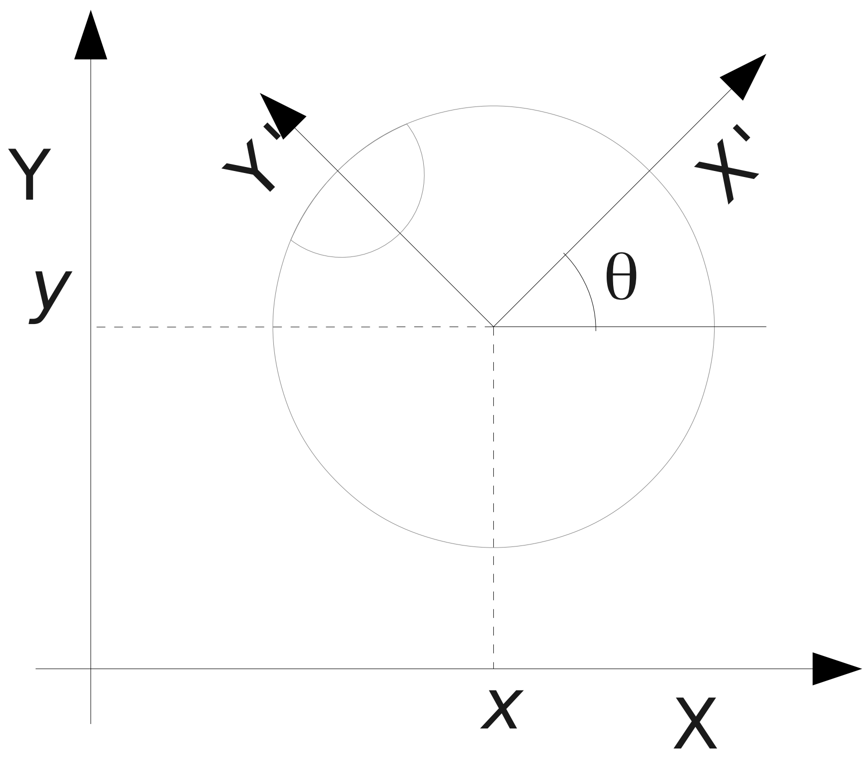
- **PID = Proportional + Integral + Derivative**
  - Reunion of previous controllers
  - One of the most popular controllers




step: kp= 1.0 kd= 0.0
y(t)
time [s]
0.1
0.08
0.06
0.04
0.02
0
0
1
2
3
4
5

control systems of CAMBADA robots

position control in a mobile robot

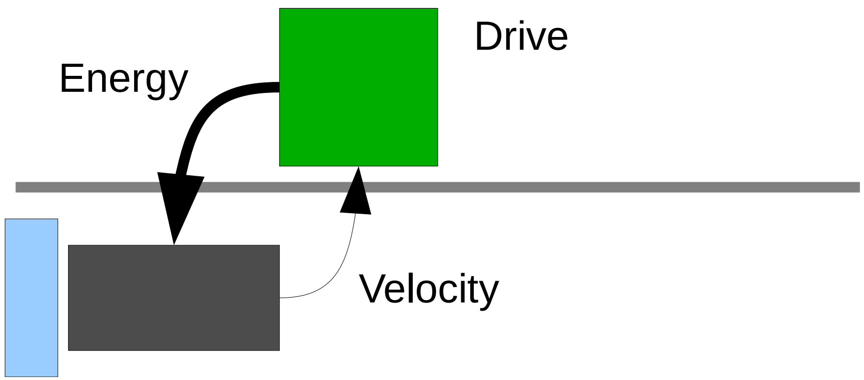


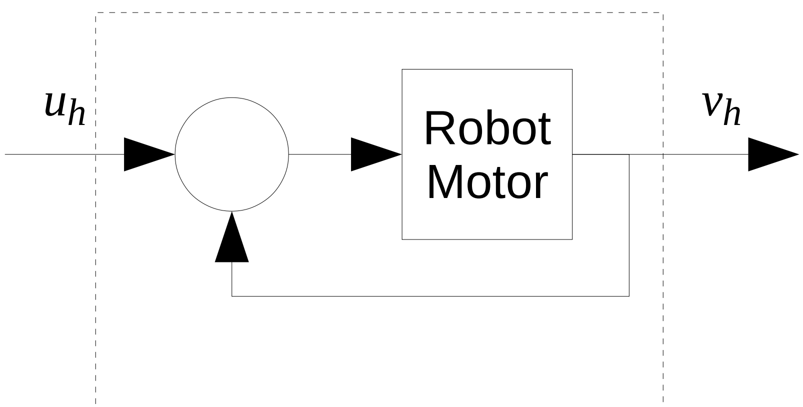
- **Robot location in the plane**
  - Pose $\begin{bmatrix} x \\ y \\ \Theta \end{bmatrix}$

- **system is defined by the pose and 1st derivative**

$$\begin{bmatrix} x \\ y \\ \Theta \\ \dot{x} \\ \dot{y} \\ \dot{\Theta} \end{bmatrix}$$

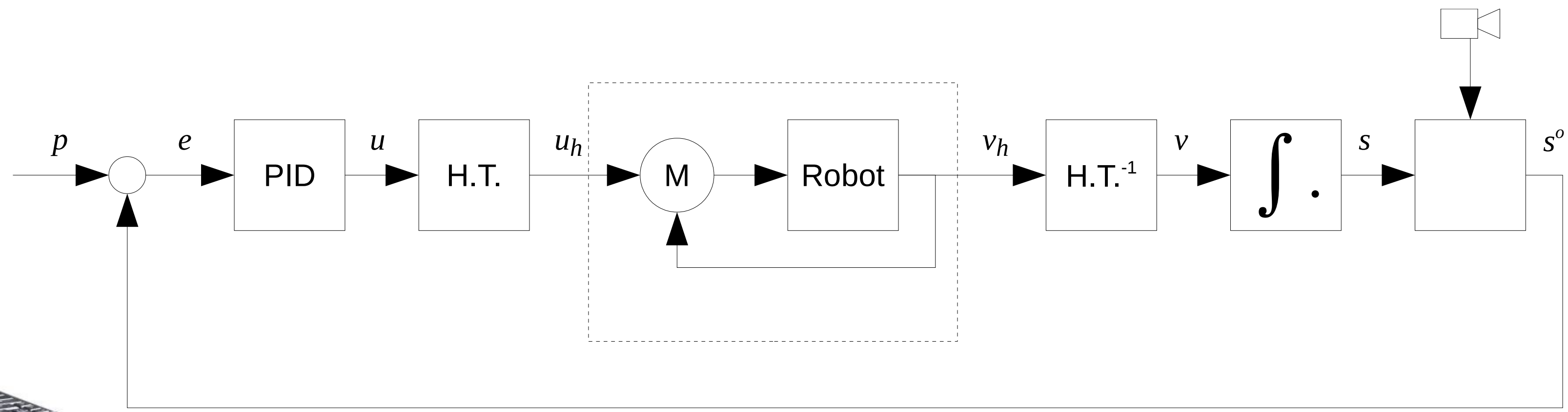
lower level

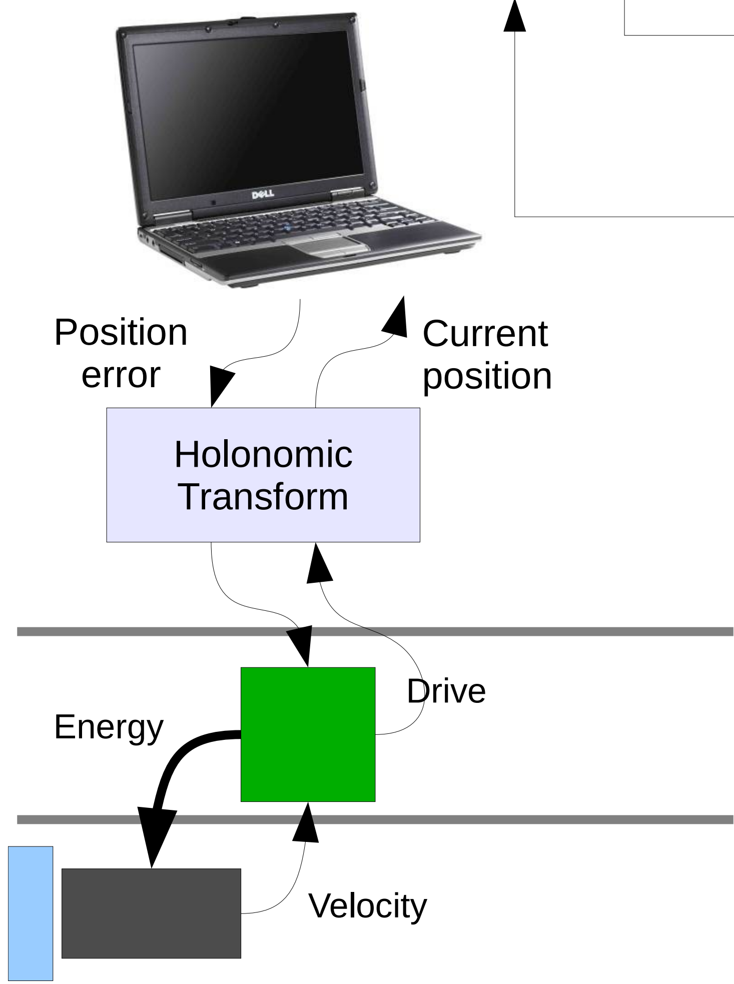




- **Motor and *drive* implement a local velocity feedback loop**
- **Motor drive:**
  - Receives velocity command
  - Modulates energy delivered to the motor
  - Receives velocity feedback

upper level



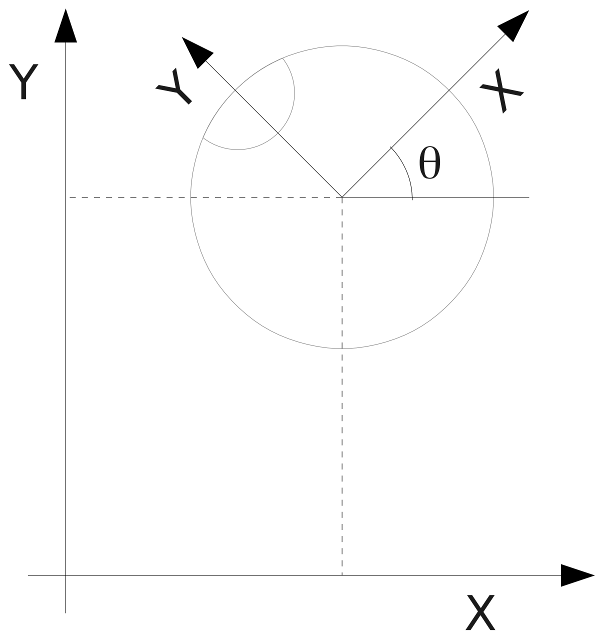


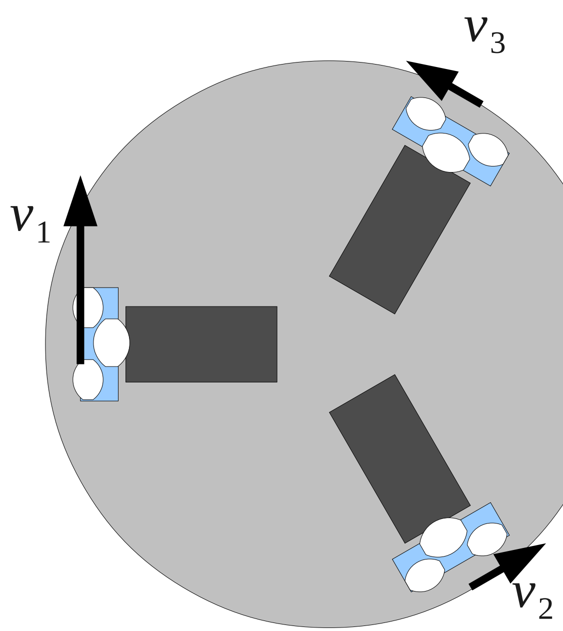
- **PC implements a position control loop in SW**
  - defines set-point *p*
  - receives current position s^0
  - computes error, applies PID and sends control signal *u*

holonomic movement

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\Theta} \end{bmatrix} \rightarrow \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

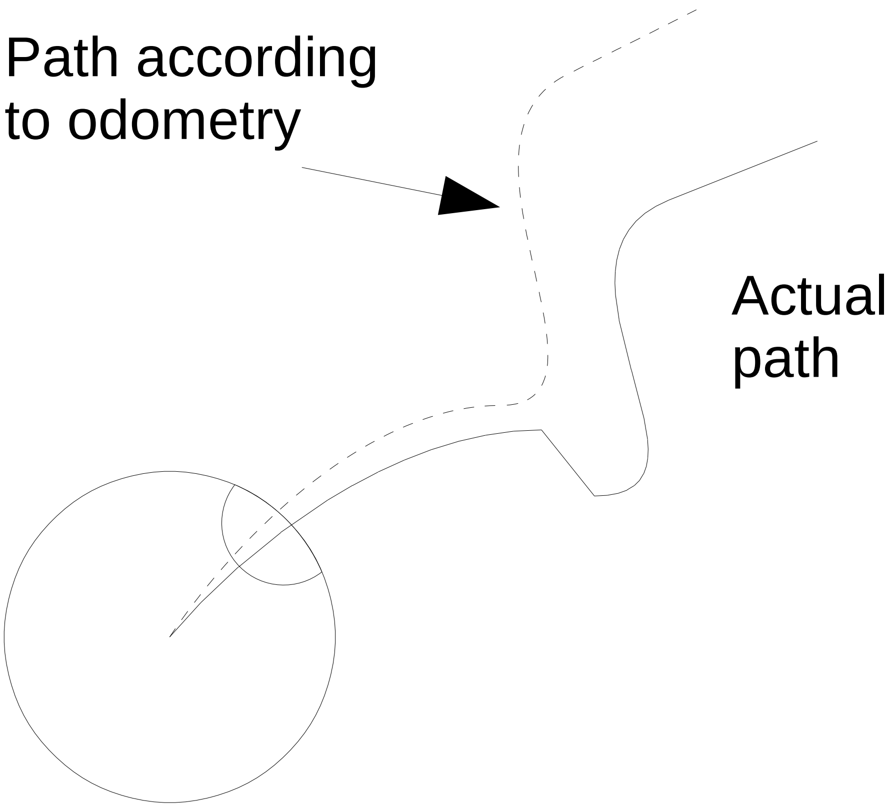
Convert a set of 3 motor speeds to linear X, Y and rotating speed and vice-versa





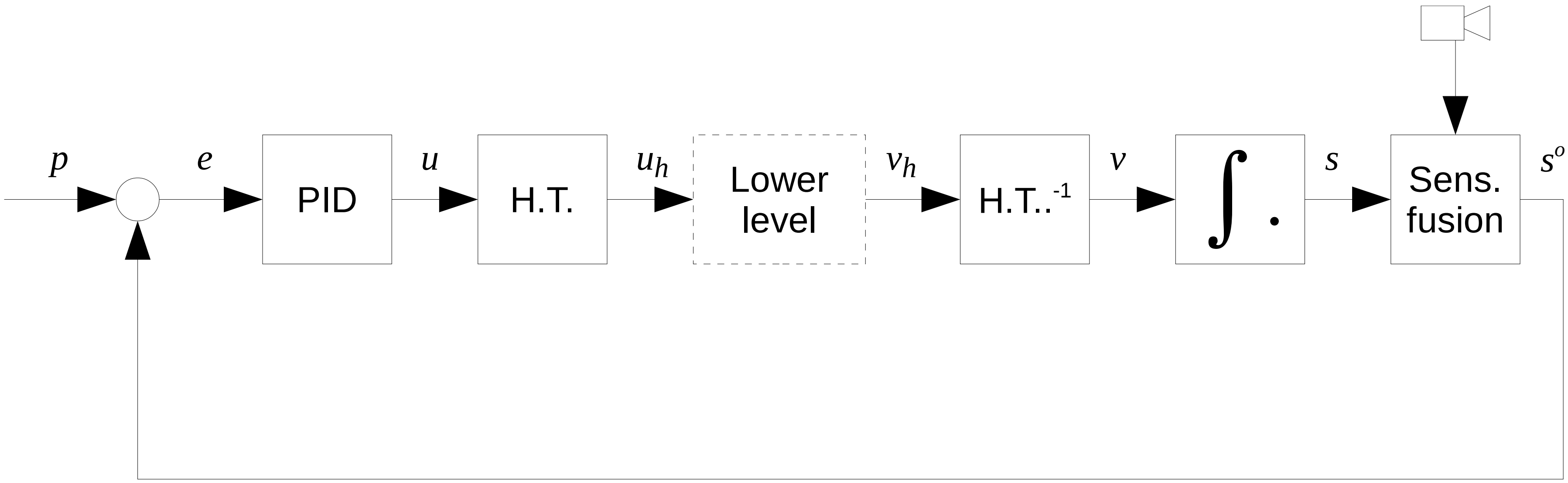
Odometry: to estimate the robot's displacement by the motors rotation (and holonomic conversion)

sensorial fusion



- **Sensorial fusion allows a more reliable position feedback**
  - e.g. robot position is given by:
    - odometry
    - camera image

control structure in CAMBADA robots



- **this control structure will try to place the robot at point *p***
- **robot control (and command) is done computing a sequence of points *p* over time**
  - navigation output