RobotDoc

BASIC MOTOR CONTROL

We'd like to...

- Describe a single joint of an otherwise complex robot
- Describe a controller for the single joint
- Describe how to deal with the math of multiple joints
- Describe which approximations we're taking in this "simple" description of the robot
- Show that we could do better if we're ready to do more complex stuff

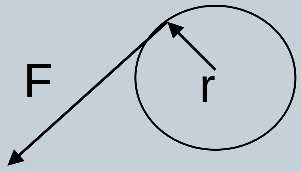
Notation

$$F = \frac{d}{dt}(mv) = m\ddot{x}$$

Since links are physical objects with mass

$$\tau = J\ddot{\vartheta}$$

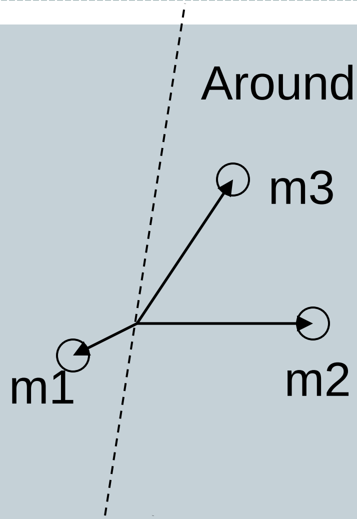
J = moment of inertia



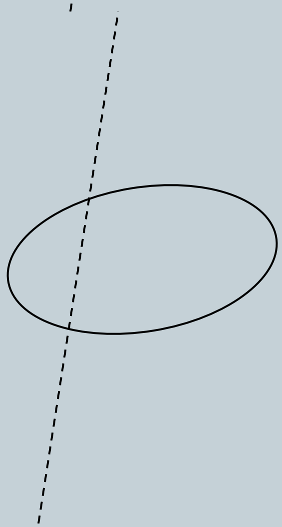
$$\tau = F \times r$$

Moment of inertia

Around an axis



$$J = \sum_{i=1}^{N} m_i r_i^2$$



density

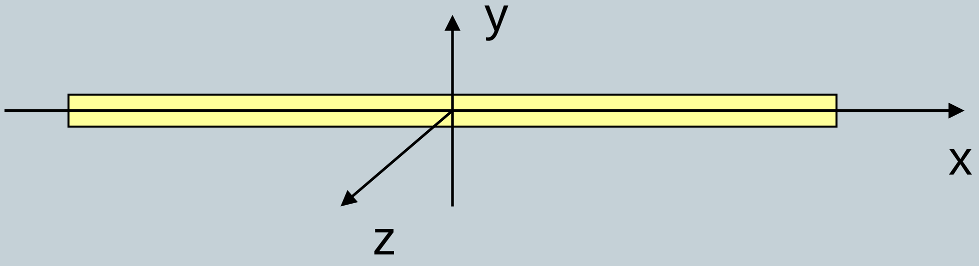
$$J = \int_{volume} \rho r^2 dV$$

Parallel axis theorem

$$J = J_c + Mr^2$$

Through the center of gravity

Example



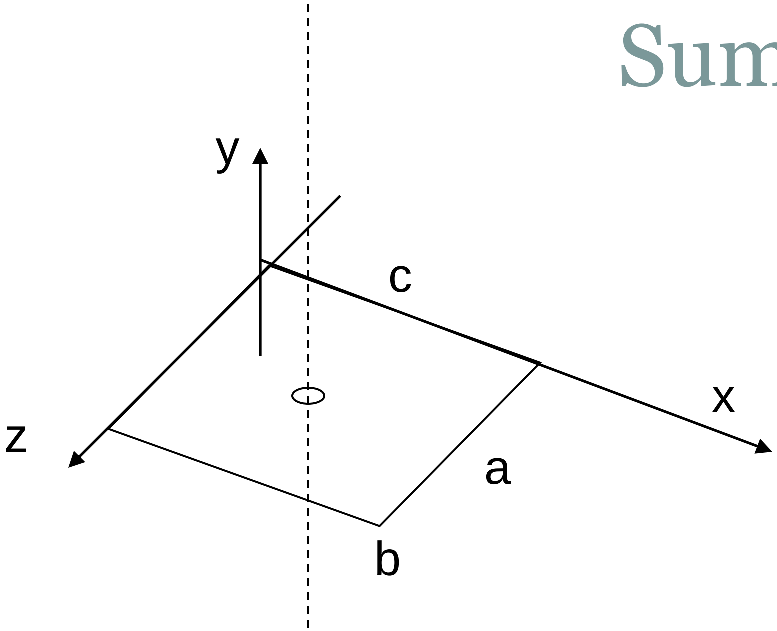
$$Mass = M, \rho = M/l$$

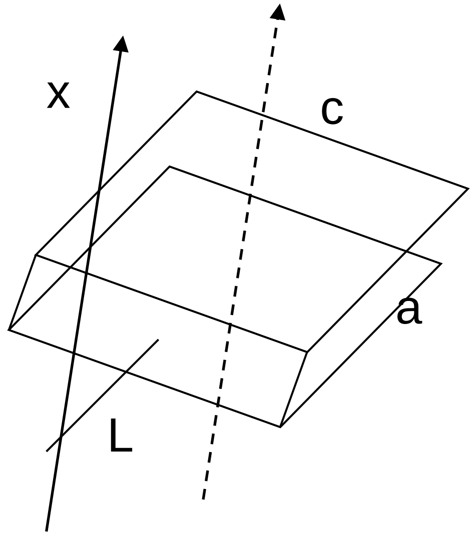
$$J_x = 0$$

$$J_y = \rho \int r^2 dV = \rho \int_{-l/2}^{l/2} x^2 dx = \rho \frac{1}{3} x^3 \Big|_{-l/2}^{l/2} = \frac{Ml^2}{12}$$

$$J_{y=-l/2} = \frac{Ml^2}{12} + M\frac{l^2}{4} = M\frac{l^2}{3}$$

Sum of J





$$J_x = \frac{M}{12}(a^2 + b^2)$$

$$J_y = \frac{M}{12}(a^2 + c^2)$$

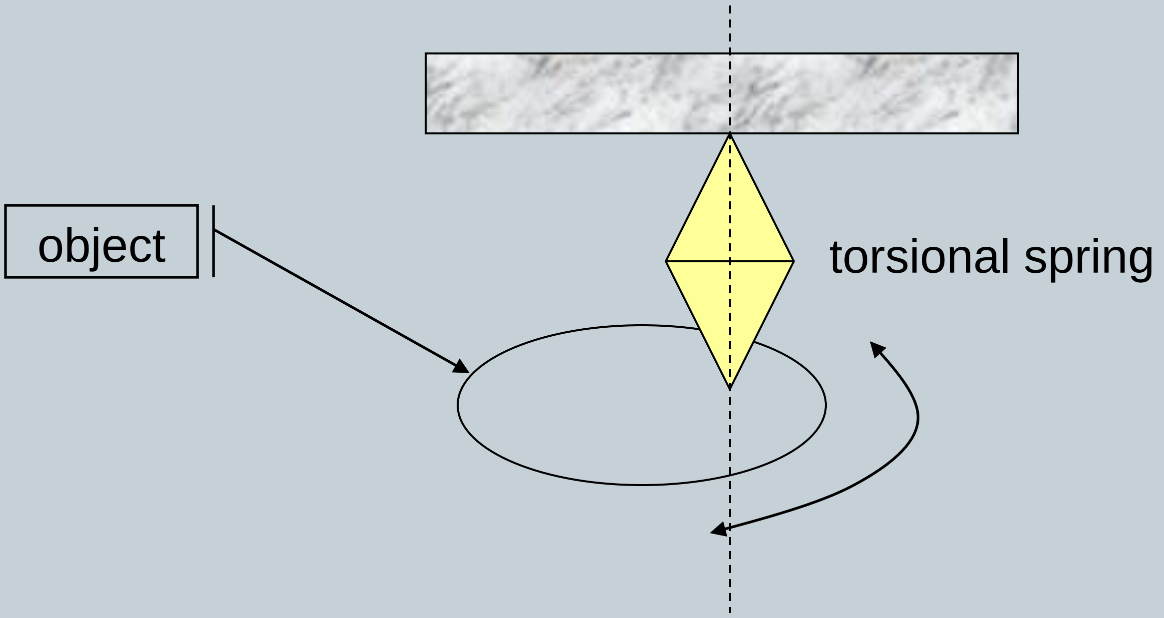
$$J_z = \frac{M}{12}(b^2 + c^2)$$

e.g. $\rightarrow J_{top-x} = \frac{M_{top}}{12}(a^2 + c^2) + M_{top}(\frac{a}{2} + L)^2$

$\downarrow$

$$J_{hand-x} = J_{top-x} + J_{side-x} + J_{bottom-x}$$

Experimental estimation of *J*

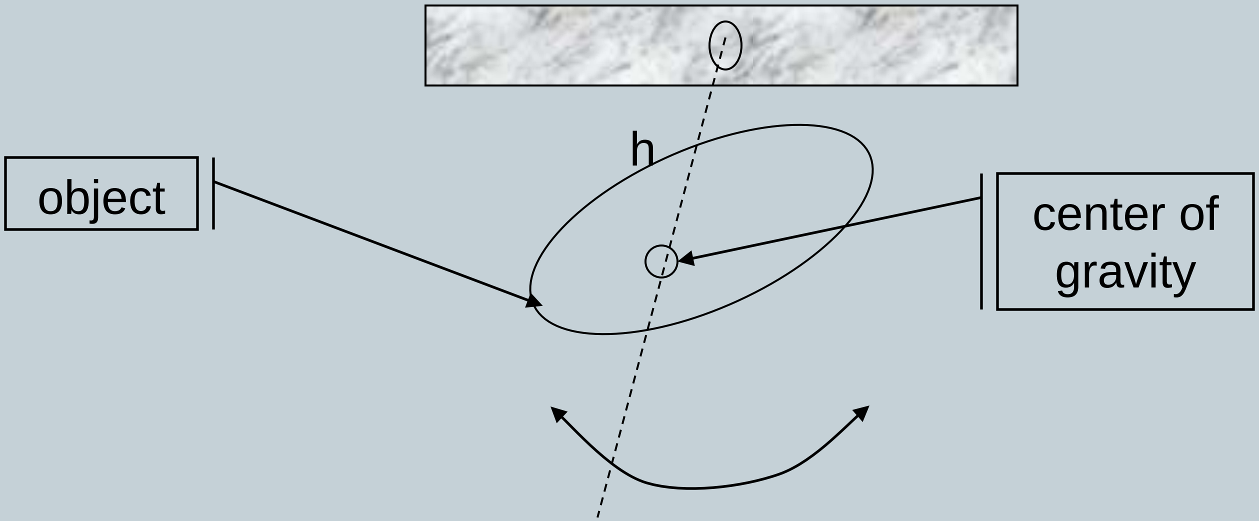


Use a photodiode and a computer to measure the frequency

Requires calibration from known *J*

$$f = \frac{1}{2\pi}\sqrt{\frac{K}{J}}$$

Experimental estimation of J



$$f \approx \frac{1}{2\pi}\sqrt{\frac{Mgh}{J}}$$

Work and power

$$E = const \quad \textbf{\textit{if}} \quad \sum F_{ext} = 0$$

$$W = \int_{s1}^{s2} F ds \qquad W = \Delta E, E = \text{energy}$$

$$K = \frac{1}{2} m v^2$$ **kinetic energy**

$$P = \frac{dW}{dt}$$ Power $\rightarrow$ $$P = Fv$$

Rotational case

$$E = const$$ ***if*** $$\sum \tau_{ext} = 0$$

$$W = \int_{\vartheta 1}^{\vartheta 2} \tau d\vartheta$$

$$W = \Delta E, E = \text{energy}$$

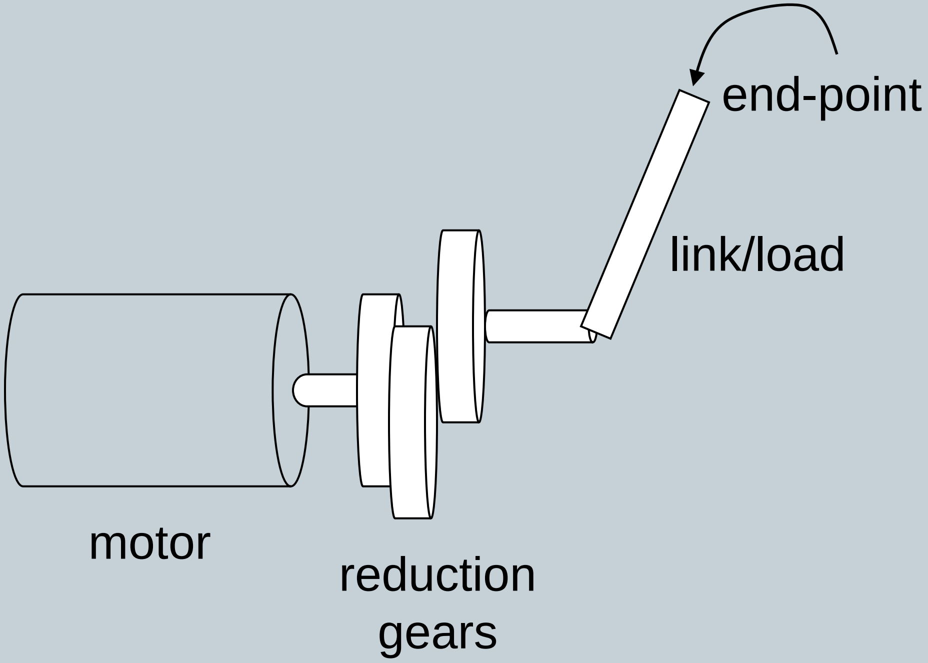
$$K = \frac{1}{2} J\omega^2$$

kinetic energy

$$P = \frac{dW}{dt}$$

Power $\rightarrow$ $$P = \tau\omega$$

Single joint model



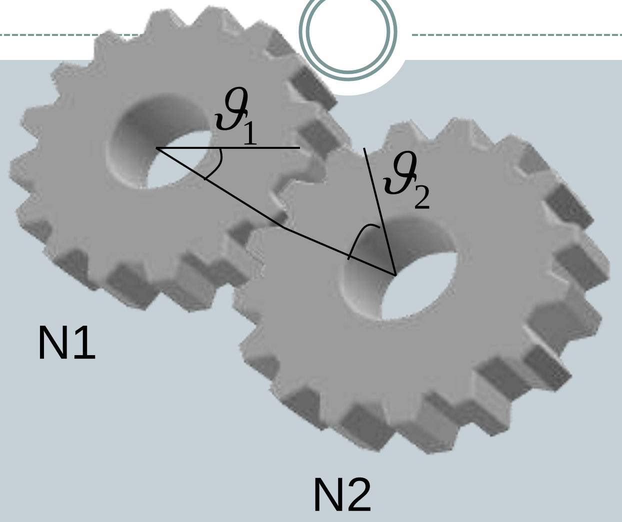
Motor

- Let's imagine for now that it is something that generates a given torque

Mechanical transmission

- Gears
- Belts
- Lead screws
- Cables
- Cams
- etc.

Gears



- Distance traveled is the same:

$$r_1\vartheta_1 = r_2\vartheta_2$$

- Because the size of teeth is the same:

$$\frac{N_1}{r_1} = \frac{N_2}{r_2}$$

Furthermore…

$$r_1\vartheta_1 = r_2\vartheta_2$$

$$\frac{N_1}{r_1} = \frac{N_2}{r_2}$$

- No loss of energy

$$\tau_1\vartheta_1 = \tau_2\vartheta_2$$

Combining…

$$\frac{N_1}{N_2} = \frac{r_1}{r_2} = \frac{\vartheta_2}{\vartheta_1} = \frac{\tau_1}{\tau_2} = \frac{\omega_2}{\omega_1} = \frac{\alpha_2}{\alpha_1}$$

# of teeth

Inverse relationship between speed and torque

input

$$\tau_2 = \tau_1 \frac{N_2}{N_1} \qquad TR = \frac{N_1}{N_2}$$

output

mechanical parameter

Equivalent J

$$\ddot{\vartheta}_1 J_1 \Leftarrow \tau_1 = \tau_2 \frac{N_1}{N_2} = \ddot{\vartheta}_2 J_2 \frac{N_1}{N_2}$$

$$J_1 = \frac{\ddot{\vartheta}_2}{\ddot{\vartheta}_1} J_2 \frac{N_1}{N_2} \Rightarrow \left(\frac{N_1}{N_2} \right)^2 J_2$$

$$J_1 = TR^2 J_2$$

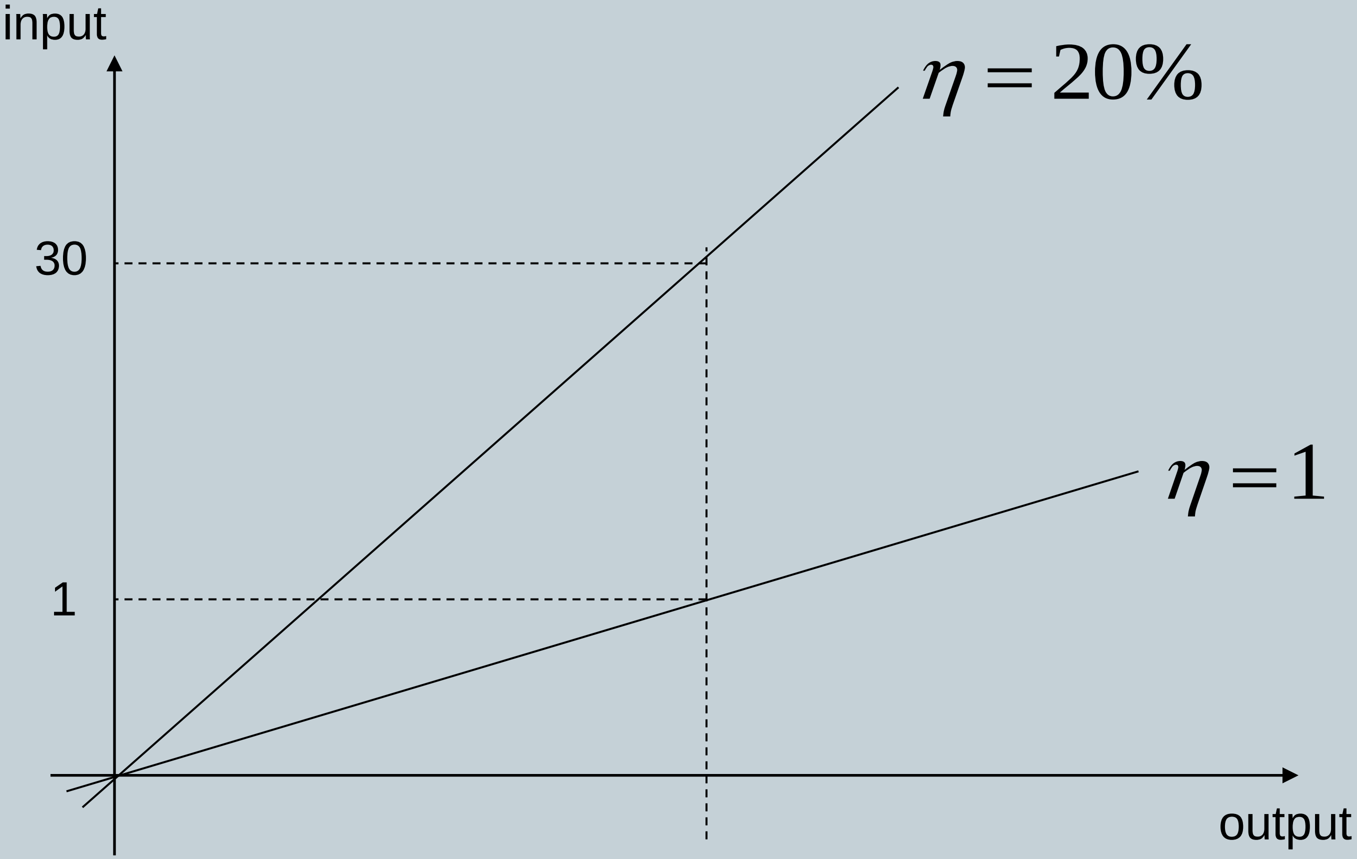
- J as seen from the motor

In reality

$$\tau_2 = \tau_1 \frac{1}{TR} \eta$$

- Where η is the efficiency of the mechanism (from 0 to 1)
- η is related to power, speed ratio doesn't change
- η is also the ratio of input power vs. power at the output

For example



Example

Specifications

reduction ratio (nominal)	weight without motor g	length without motor L2 mm	length with motor 1319 T L1 mm	length with motor 1331 T L1 mm	length with motor 1336 U L1 mm	output torque continuous operation M max. mNm	output torque intermittent operation M max. mNm	direction of rotation (reversible)	efficiency %
3,71:1	17	20,9	34,1	45,9	50,9	200	300	=	90
14 :1	20	25,0	38,2	50,0	55,0	300	450	=	80
43 :1	24	29,2	42,4	54,2	59,2	300	450	=	70
66 :1	24	29,2	42,4	54,2	59,2	300	450	=	70
134 :1	27	33,3	46,5	58,3	63,3	300	450	=	60
159 :1	27	33,3	46,5	58,3	63,3	300	450	=	60
246 :1	27	33,3	46,5	58,3	63,3	300	450	=	60
415 :1	30	37,4	50,6	62,4	67,4	300	450	=	55
592 :1	30	37,4	50,6	62,4	67,4	300	450	=	55
989 :1	30	37,4	50,6	62,4	67,4	300	450	=	55
1 526 :1	30	37,4	50,6	62,4	67,4	300	450	=	55

Motion conversion

- Start with

$$\tau_2 = \frac{N_2}{N_1}\tau_1$$

- Design TR, more torque (usually)

$$TR < 1$$

$$N_2 > N_1$$

$$J_1 < J_2 \Leftrightarrow \omega_2 < \omega_1$$

Viscous friction

- Easy:

$$\tau_{viscous} = B_2 \dot{\vartheta}_2$$

$$\tau_{eq\_viscous} = TR \cdot \tau_{viscous} = TR \cdot B_2 \dot{\vartheta}_2$$

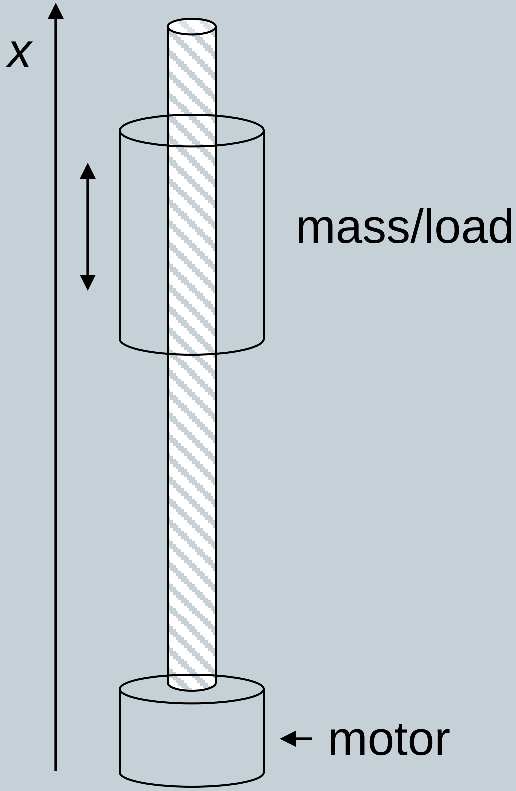
$$B_{eq} \dot{\vartheta}_1 = TR \cdot B_2 \dot{\vartheta}_2 \Rightarrow B_{eq} = TR^2 B_2$$

- Coulomb friction:

$$\tau_{eq} = TR \cdot F_c \operatorname{sgn}(\dot{\vartheta}_2)$$

Lead screw

- Rotary to linear motion conversion (P=pitch in #of turns/mm or inches)



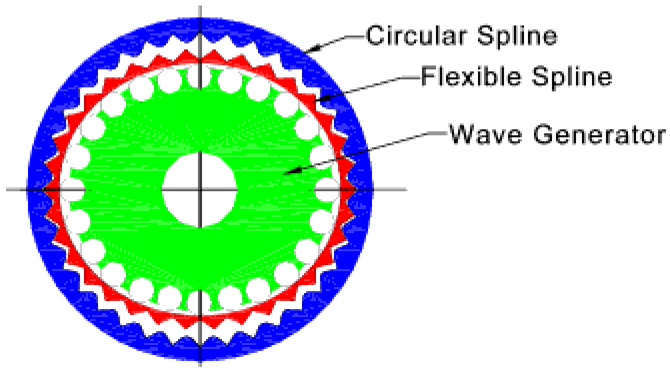
$$\vartheta[rad] = 2\pi P x$$

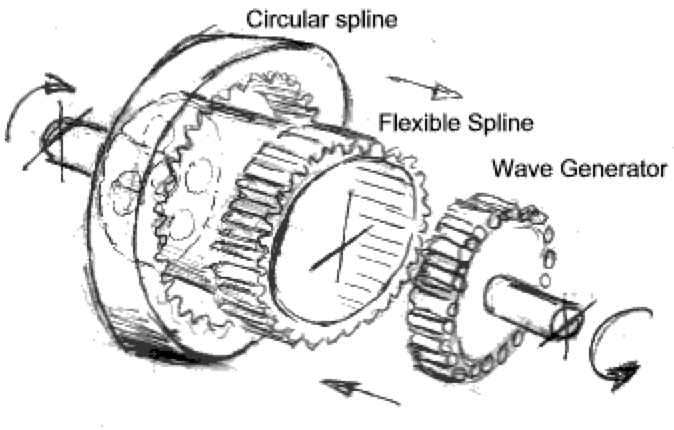
$$\dot{\vartheta} = 2\pi P \dot{x}$$

$$E_{rot} = E_{lin} \Rightarrow \frac{1}{2} M_{load} v^2 = \frac{1}{2} J \omega^2 \Rightarrow$$

$$\Rightarrow J = \frac{M_{load}}{(2\pi P)^2}$$

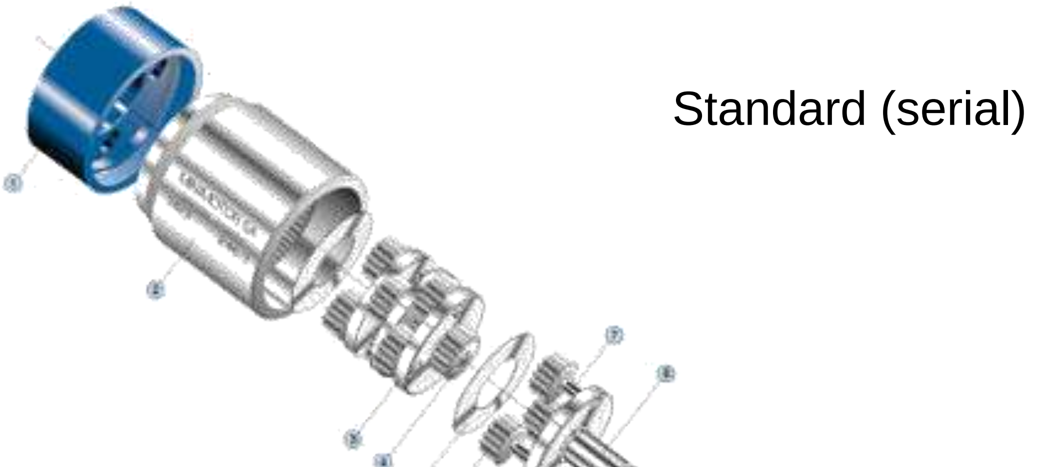
Harmonic drives



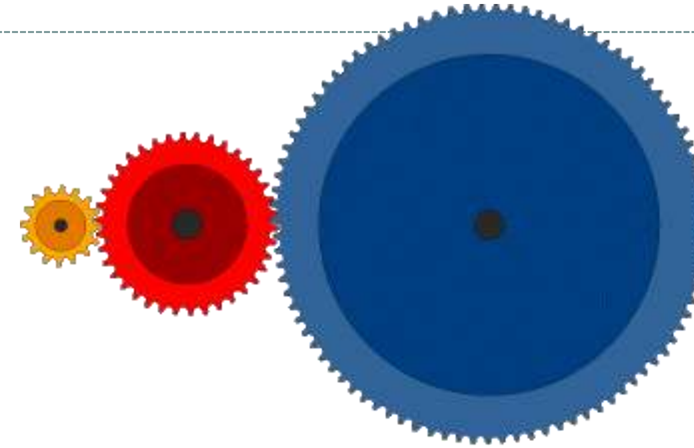


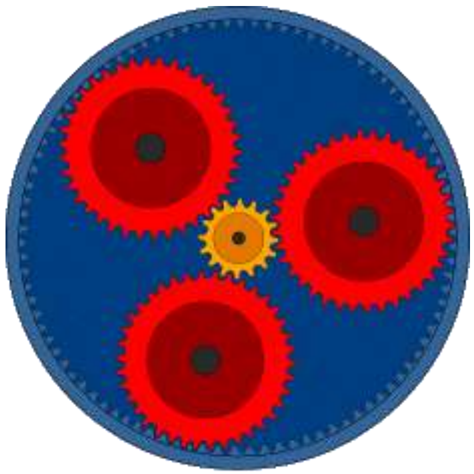
From the harmonic drive website
http://www.harmonicdrive.de

Gearhead (for real)



Standard (serial)





Planetary

Example

- Designing the single joint
  - Given:

$$\ddot{\vartheta}_{\max} \Rightarrow \tau = J_{eq}\ddot{\vartheta} \Rightarrow \tau_{\max} = J_{eq}\ddot{\vartheta}_{\max} = J_{load}TR^2\ddot{\vartheta}_{\max}$$

- Then taking into account some more realistic components:

$$\tau_{\max} = J_{load}\frac{TR^2}{\eta}\ddot{\vartheta}_{\max}$$

Example (continued)

$$\tau_{\max} = J_{load} \frac{TR^2}{\eta} \ddot{\vartheta}_{\max}$$

$$P = \tau_{\max} \dot{\vartheta} \Rightarrow \text{given } \dot{\vartheta}_{\max} \Rightarrow \text{get } P$$

motor power, from catalog

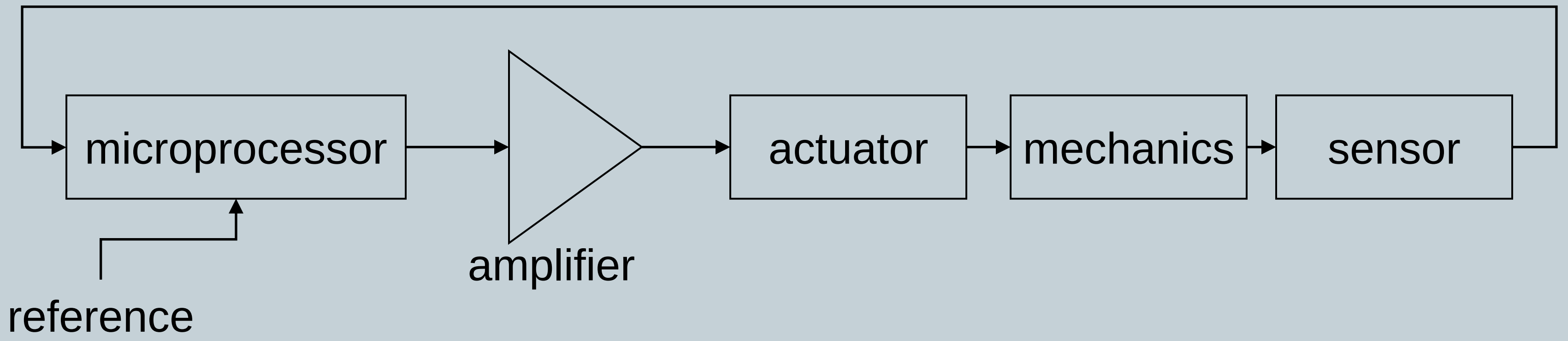
This guarantees that the motor can still deliver maximum torque at maximum speed

More on real world components

- Efficiency
- Eccentricity
- Backlash
- Vibrations

- To get better results during design mechanical systems can be simulated

Control of a single joint



Components

- Digital microprocessor:
  - Microcontroller, processor + special interfaces
- Amplifier (drives the motor)
  - Turns control signals into power signals
- Actuator
  - E.g. electric motor
- Mechanics/load
  - The robot!
- Sensors
  - For intelligence

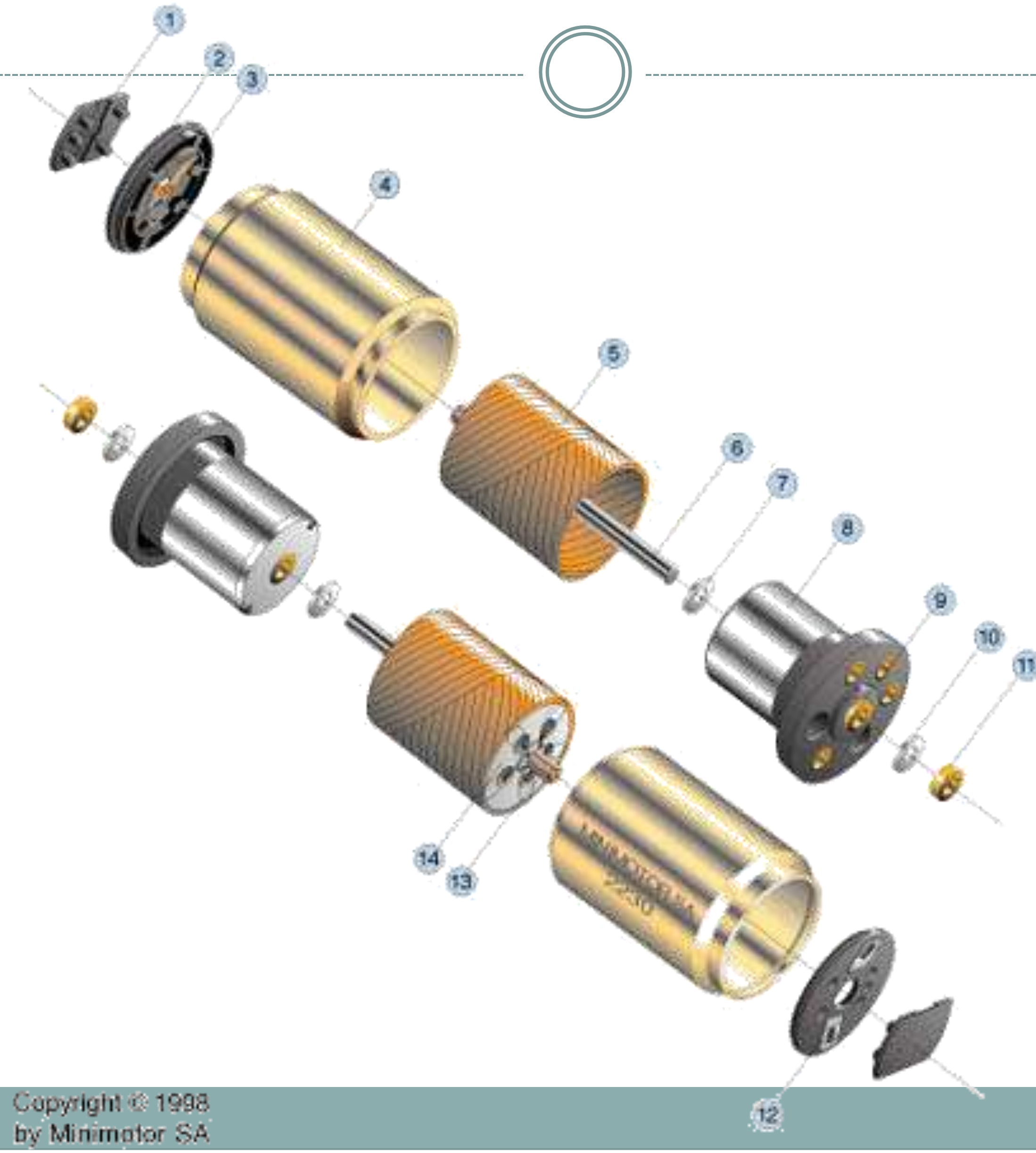
Actuators

- Various types:
  - AC, DC, stepper, etc.
  - DC
    - Brushless
    - With brushes
- We'll have a look at the DC with brushes, simple to control, widely used in robotics

DC-brushless

Copyright © 1998
by Minimotor SA

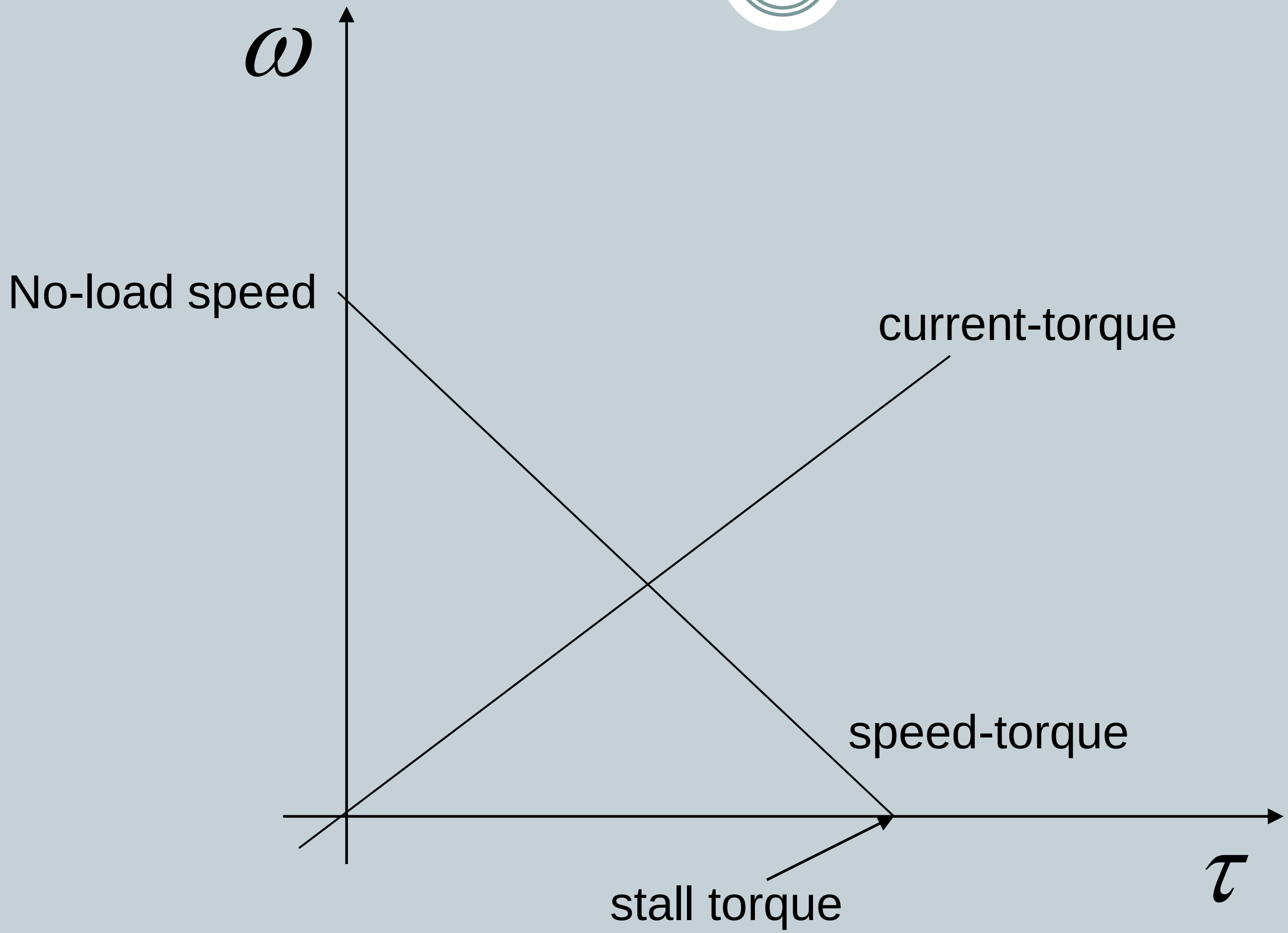
DC with brushes



Modeling the DC motor

- Speed-torque and torque-current relationships are linear

In particular



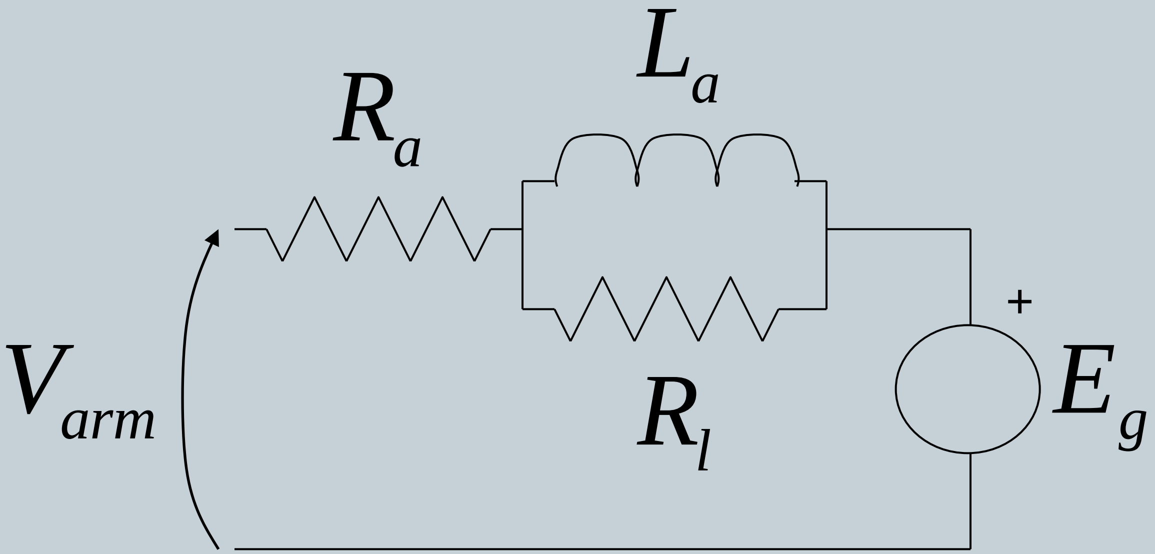
Real numbers!

http://www.minimotor.ch

Series 1331 … SR

		1331 T		006 SR	012 SR	024 SR	
1	Nominal voltage	U_N		6	12	24	Volt
2	Terminal resistance	R		2,83	13,7	52,9	Ω
3	Output power	$P_{2\,max.}$		3,11	2,57	2,66	W
4	Efficiency	$\eta_{max.}$		81	80	80	%
5	No-load speed	n_o		10 600	9 900	10 400	rpm
6	No-load current (with shaft ø 1,5 mm)	I_o		0,0220	0,0105	0,0055	A
7	Stall torque	M_H		11,20	9,90	9,76	mNm
8	Friction torque	M_R		0,12	0,12	0,12	mNm
9	Speed constant	k_n		1 790	835	439	rpm/V
10	Back-EMF constant	k_E		0,56	1,20	2,28	mV/rpm
11	Torque constant	k_M		5,35	11,4	21,8	mNm/A
12	Current constant	k_I		0,187	0,087	0,046	A/mNm
13	Slope of n-M curve	$\Delta n/\Delta M$		946	1 000	1 070	rpm/mNm
14	Rotor inductance	L		70	310	1 100	µH
15	Mechanical time constant	τ_m		7	7	7	ms
16	Rotor inertia	J		0,71	0,67	0,63	gcm²
17	Angular acceleration	$\alpha_{max.}$		160	150	160	·10³rad/s²
18	Thermal resistance	$R_{th\,1} / R_{th\,2}$	6 / 25				K/W
19	Thermal time constant	τ_{w1} / τ_{w2}	5 / 190				s
20	Operating temperature range:						
	– motor		– 30 … + 85 (optional – 55 … + 125)				°C
	– rotor, max. permissible		+ 125				°C

Electrical diagram



$$E_g = \omega(t) K_E$$

Meaning of components

R_a	Armature resistance (including brushes)
V_{arm}	Armature voltage
R_l	Losses due to magnetic field
E_g	Back EMF produced by the rotation of the armature in the field
L_a	Coil inductance

We can write…

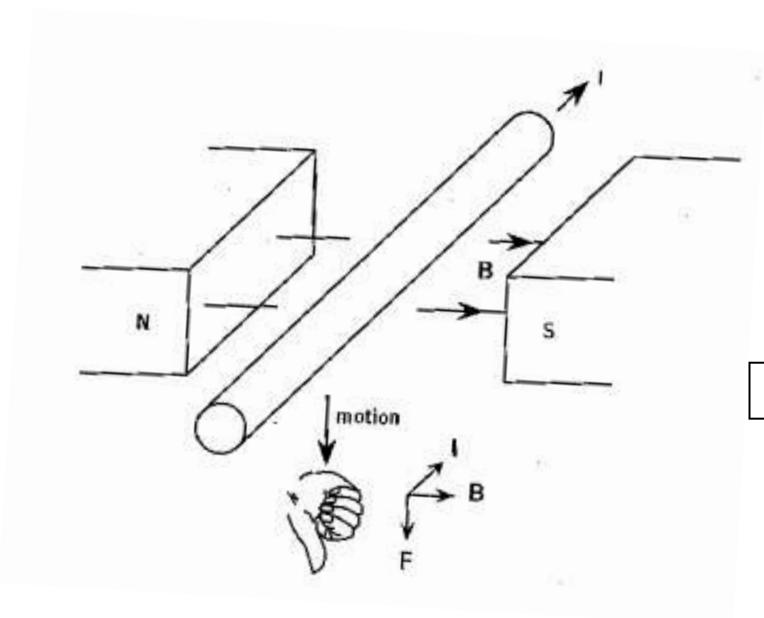
$$V_{arm} = R_a I_a + L_a \dot{I}_a + \omega(t) K_E$$

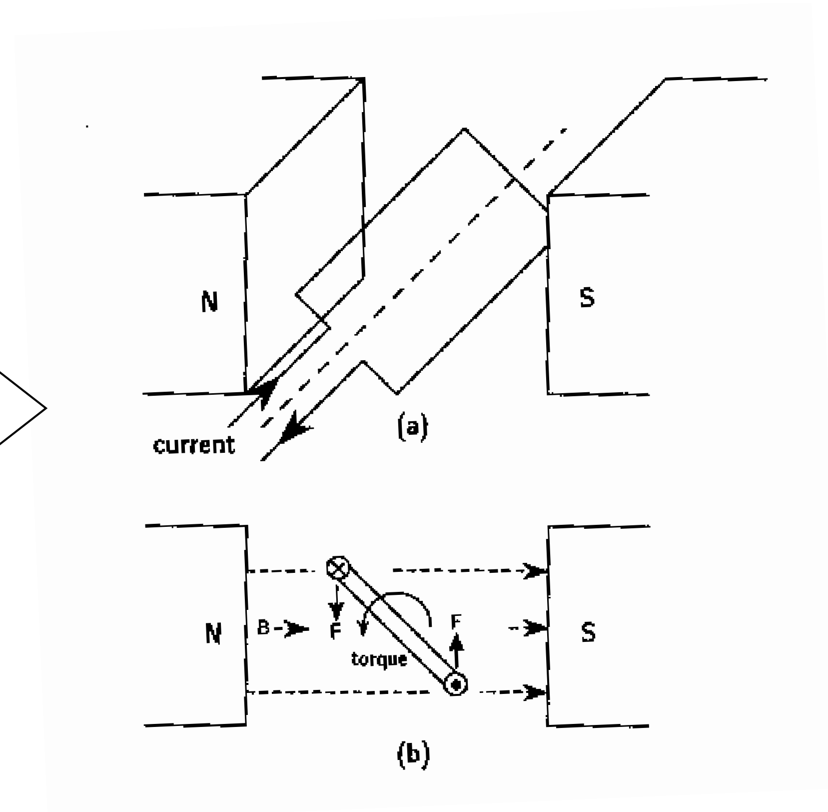
$$\text{for} \quad R_l << R_a$$

which is the case at the frequency of interest, and we also have…

$$\tau = K_T I_a$$

On torque and current

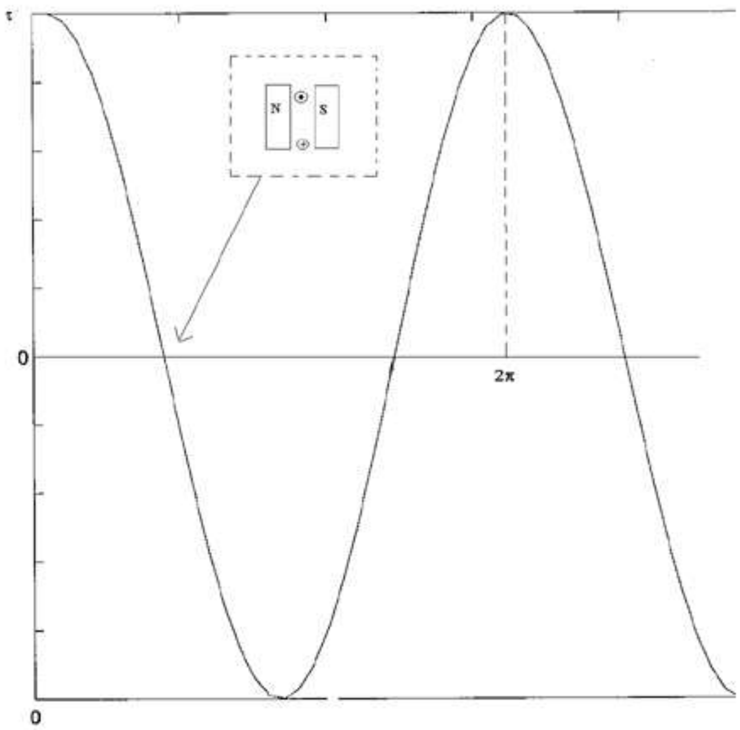


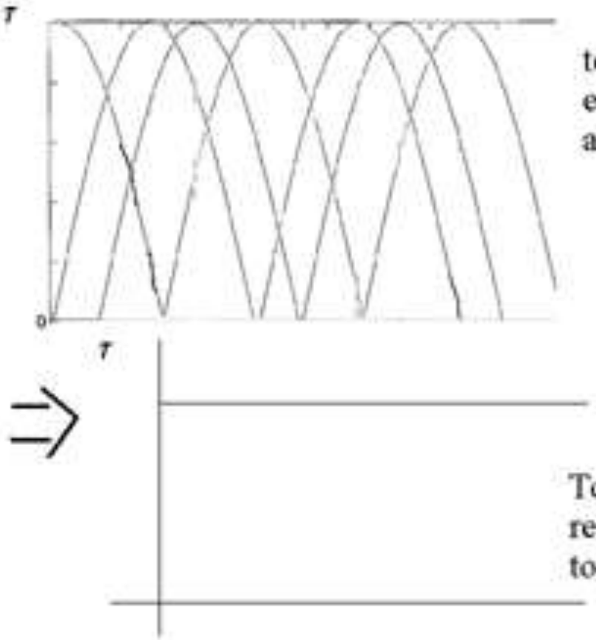


$$F = i\vec{L} \times \vec{B}$$

$$F_{lorentz} = q\vec{v} \times \vec{B}$$

Thus for many coils…





torque from each loop added

Torque is related to the total current

Back to motor modeling…

$$\tau = (J_M + J_L)\dot{\omega}(t) + B\omega(t) + \tau_f + \tau_{gr}$$

τ	Torque generated
J_M	Inertia of the motor
J_L	Inertia of the load
τ_f	Friction
τ_{gr}	Gravity

Furthermore…

$$V_{arm} = R_a I_a + L_a \dot{I}_a + \omega(t) K_E$$

$$\tau = K_T I_a$$

$$\tau = (J_M + J_L)\dot{\omega}(t) + B\omega(t) + \tau_f + \tau_{gr}$$

Consequently

$$\begin{bmatrix} \dot{I}_a \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} {R_a}/{L_a} & {K_E}/{L_a} \\ {K_T}/{J_M + J_L} & {B}/{J_M + J_L} \end{bmatrix} \cdot \begin{bmatrix} I_a \\ \omega \end{bmatrix} + \begin{bmatrix} {-V_{arm}}/{L_a} \\ {\tau_f + \tau_{gr}}/{J_M + J_L} \end{bmatrix}$$

- A linear system of two equations (differential)
- Q: can you write a transfer function from these equations?
- Q: can you transform the equations into a block diagram?

By Laplace-transforming

$$V_{arm}(s) = R_a I_a(s) + L_a I_a(s)s + \omega(s)K_E \Rightarrow I_a(s) = \frac{V_{arm}(s) - \omega(s)K_E}{R_a + L_a s}$$

$$\tau = K_T I_a$$

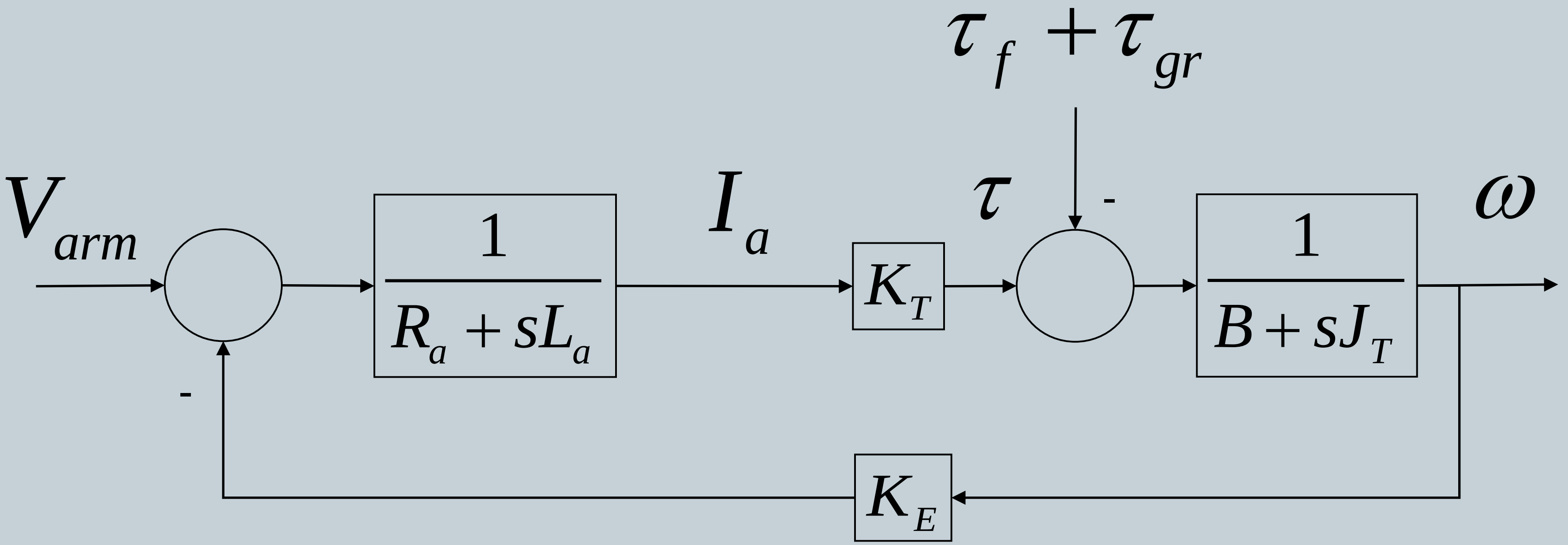
$$K_T \frac{V_{arm}(s) - \omega(s)K_E}{R_a + L_a s} = (J_M + J_L)\omega(s)s + B\omega(s) + \tau_f + \tau_{gr}$$

and finally

$$\frac{\omega(s)}{V_{arm}(s)} = \frac{K_T / L_a J_T}{s^2 + [(R_a J_T + L_a B)/L_a J_T]s + (K_T K_E + R_a B)/L_a J_T}$$

- Considering gravity and friction as additional inputs

Block diagram

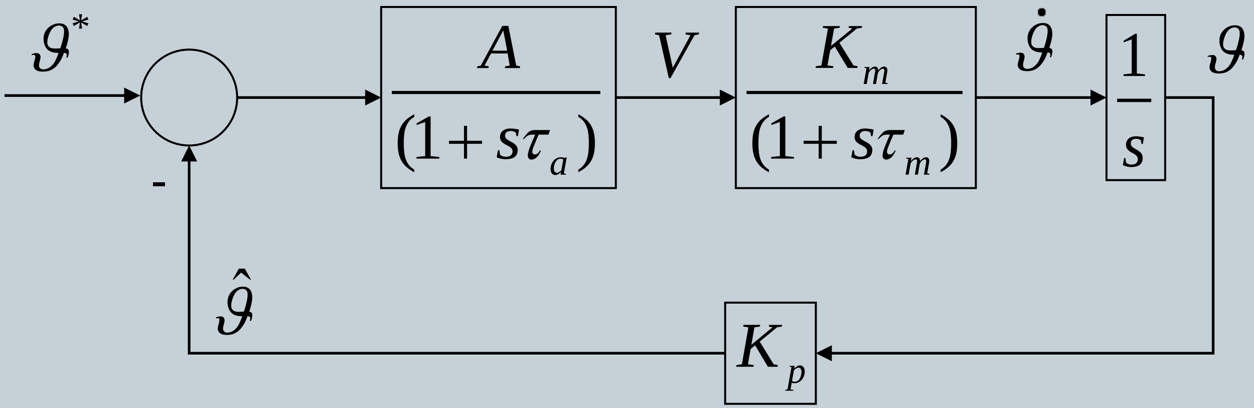


Analysis tools

- Control: determine V_a so to move the motor as desired

- Root locus
- Pole placement
- Frequency response
- Etc.

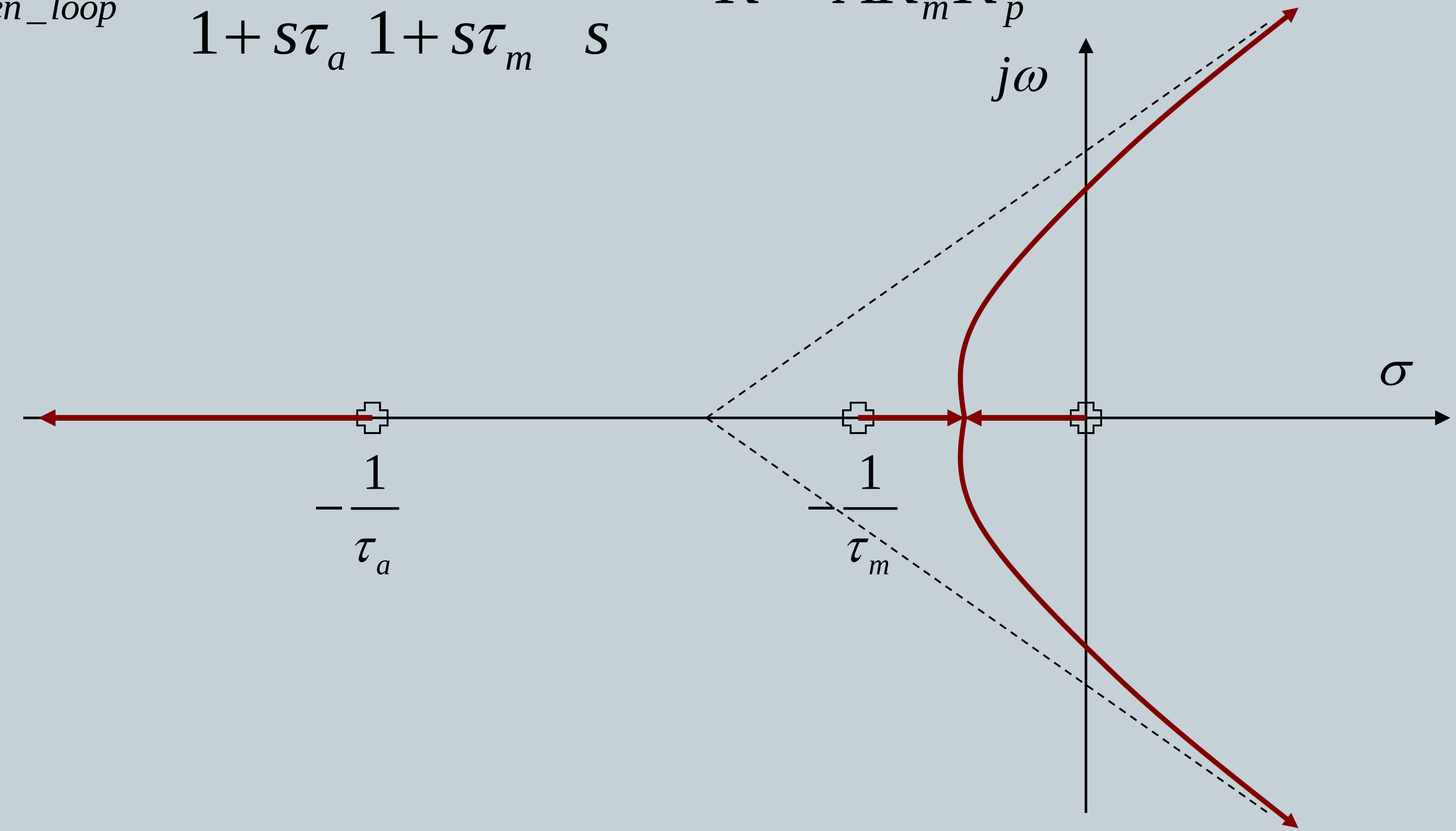
First block diagram



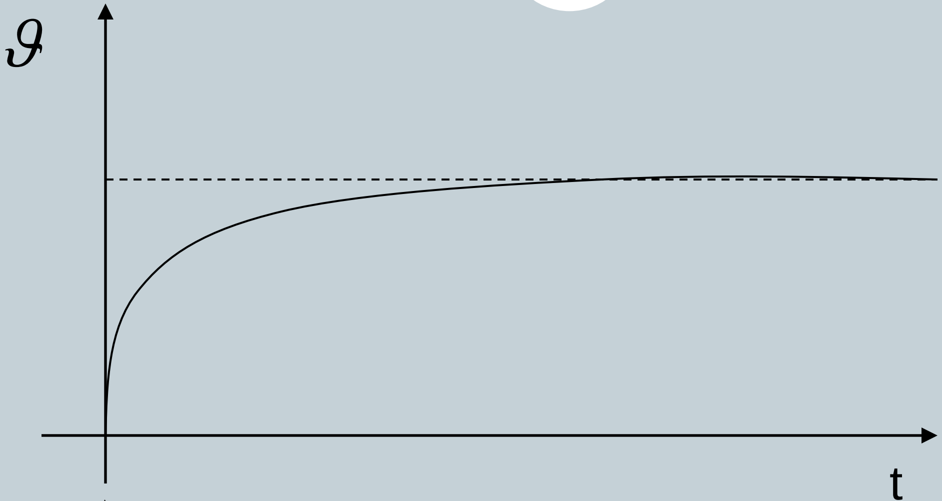
$$H_{open\_loop} = \frac{A}{1+s\tau_a}\frac{K_m}{1+s\tau_m}\frac{K_p}{s}$$

Root locus

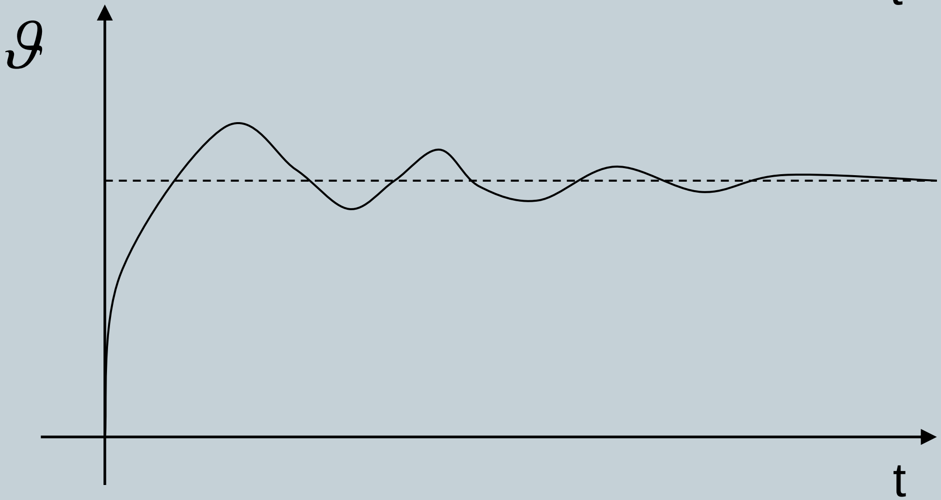
$$H_{open\_loop} = \frac{A}{1+s\tau_a}\frac{K_m}{1+s\tau_m}\frac{K_p}{s} \qquad K = AK_mK_p$$



Changing K

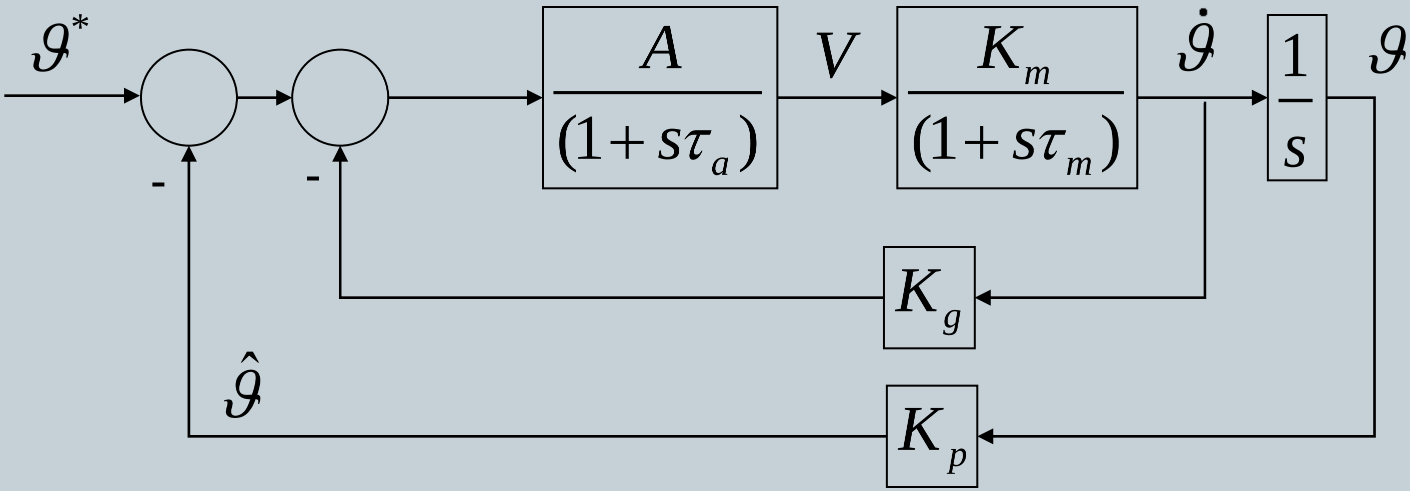


small K



higher K

Let's add something, second diagram



$$H_{open\_loop} = \frac{AK_m(K_p + sK_g)}{(1+s\tau_a)(1+s\tau_m)s}$$

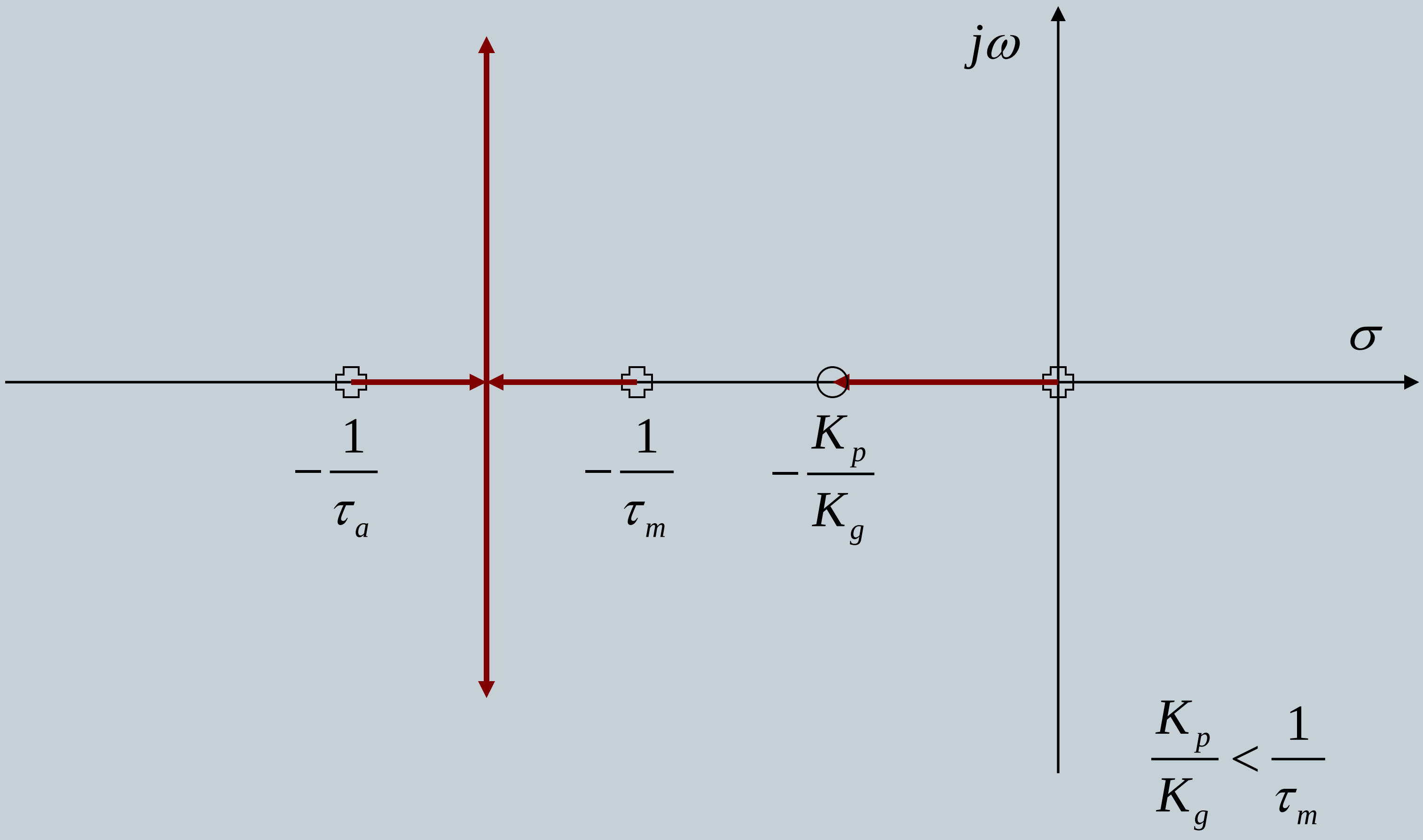
Analysis

$$H_{open\_loop} = \frac{AK_m K_p (1 + s\, {}^{K_g}\!/_{K_p})}{(1 + s\tau_a)(1 + s\tau_m)s}$$

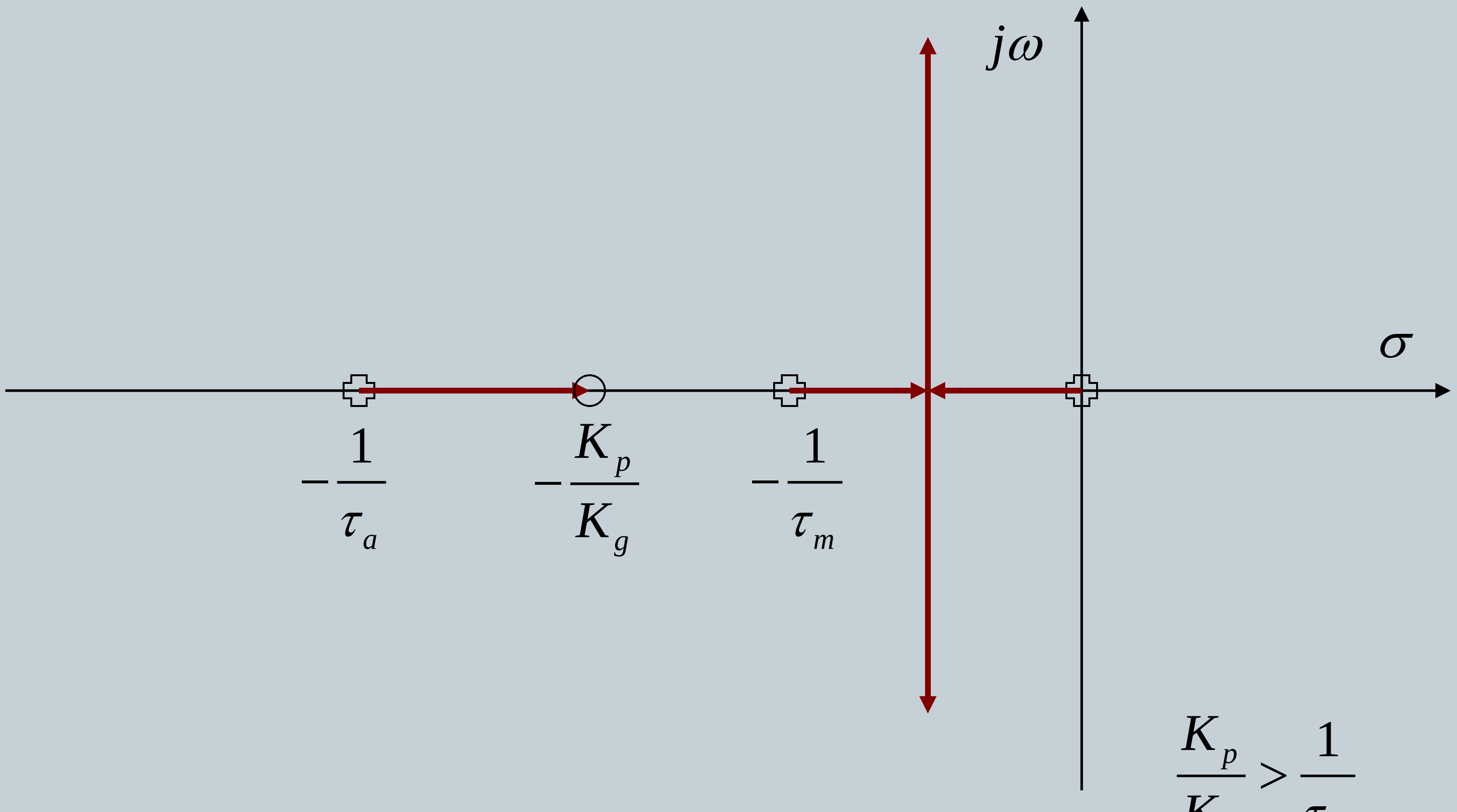
$$K = AK_m K_p$$

$$Z_{feedback} = \frac{K_g}{K_p}$$

Root locus (case 1)

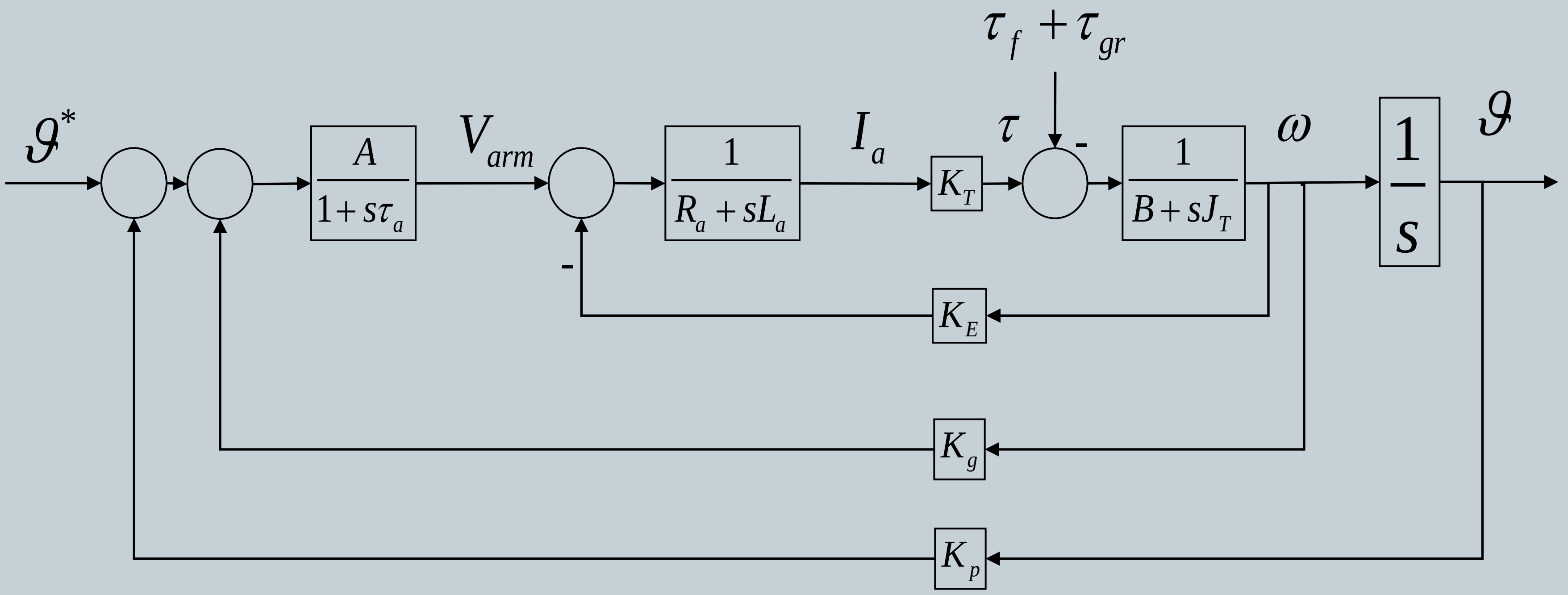


Root locus (case 2)



$$\frac{K_p}{K_g} > \frac{1}{\tau_m}$$

Overall…



ϑ^*
$\frac{A}{1+s\tau_a}$
V_{arm}
-
$\frac{1}{R_a+sL_a}$
I_a
K_T
τ
$\tau_f+\tau_{gr}$
-
$\frac{1}{B+sJ_T}$
ω
$\frac{1}{s}$
ϑ
K_E
K_g
K_p

Error and performance

$$\vartheta = \frac{\vartheta_d}{s}$$

$$M(s) = \frac{K_T}{(R_a + sL_a)(B + sJ_T) + K_E K_T}$$

closed loop (velocity)

$$\omega(s) = \frac{\frac{A}{1+s\tau_a} M(s)}{1 + \frac{A}{1+s\tau_a} M(s) K_g}$$

$$\vartheta(s) = \frac{1}{s}\omega(s)$$

closed loop (position)

$$\vartheta(s) = \frac{\frac{1}{s}\omega(s)}{1 + \frac{1}{s}\omega(s) K_p}$$

finally

$$\lim_{s\to 0} sH(s) = \lim_{t\to\infty} h(t)$$

$$\Rightarrow \lim_{s\to 0} s\frac{\vartheta_d}{s}\vartheta(s) = \lim_{s\to 0}\frac{s\frac{1}{s}\frac{\vartheta_d}{s}\omega(s)}{1+\frac{1}{s}\omega(s)K_p} = \frac{\vartheta_d}{K_p}$$

- For zero error *K* must be 1 or the control structure must be different

Same line of reasoning

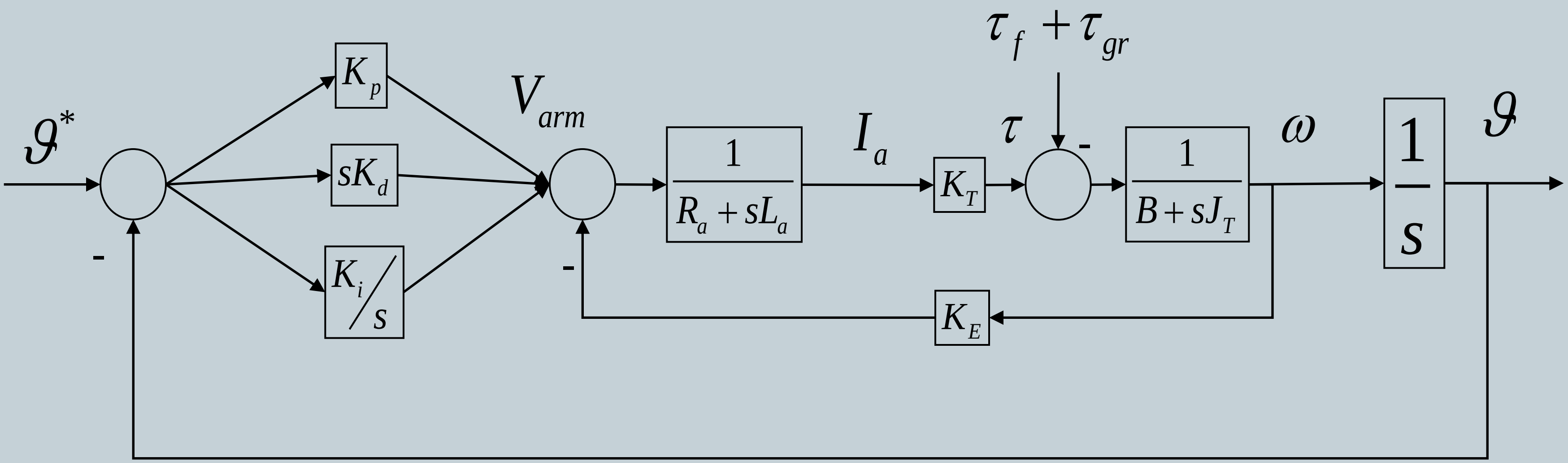
$$\vartheta_{final} = -\frac{\tau_{gr} R_a}{A K_T K_p}$$

- Final value due to friction and gravity

$$\left| \frac{\tau_{gr} R_a}{A K_T K_p} \right| \leq \vartheta_{\max} \Rightarrow K_p \geq \frac{\tau_{gr} R_a}{A K_T \vartheta_{\max}}$$

$$K_{p\min} = \frac{\tau_{gr} R_a}{A K_T \vartheta_{\max}}$$

PID controller



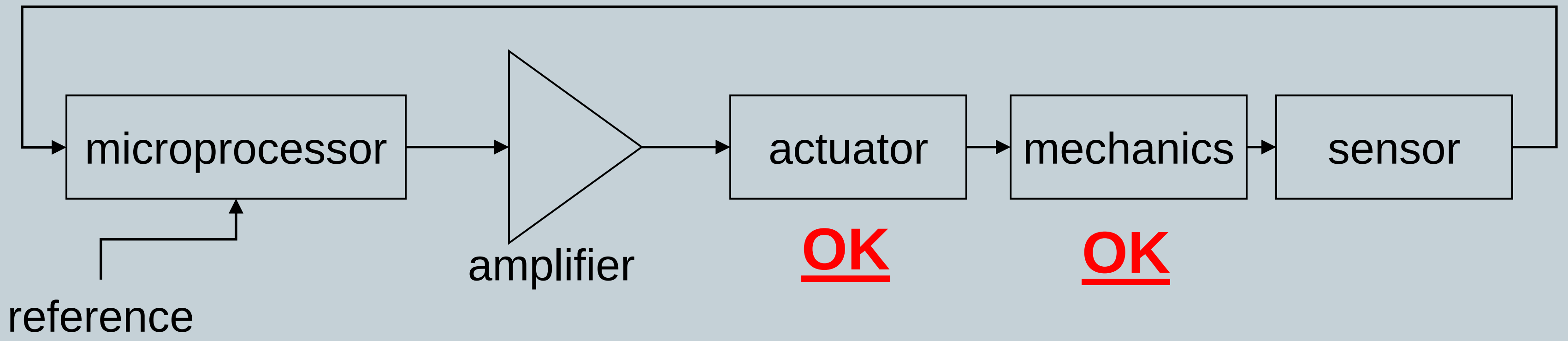
PID controller

- We now know why we need the proportional
- We also know why we need the derivative
- Finally, we add the integral
  - Integrates the error, in practice needs to be limited

Interpreting the PID

- Proportional: to go where required, linked to the steady-state error
- Derivative: damping
- Integral: to reduce the steady-state error

Global view

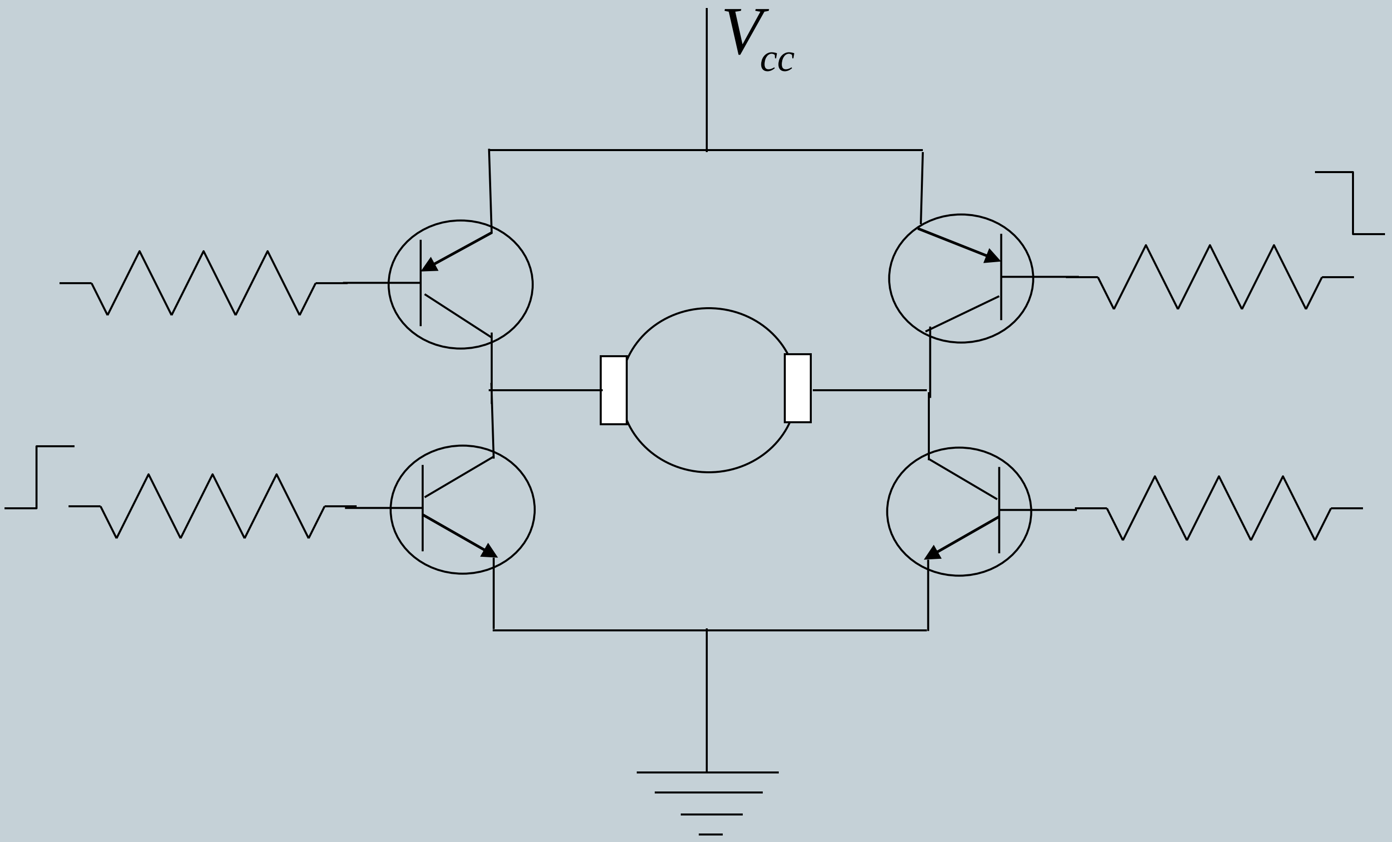


microprocessor
actuator
mechanics
sensor
reference
amplifier
OK
OK

About the amplifiers

- Linear amplifiers
  - H type
  - T type
- PWM (switching) amplifiers

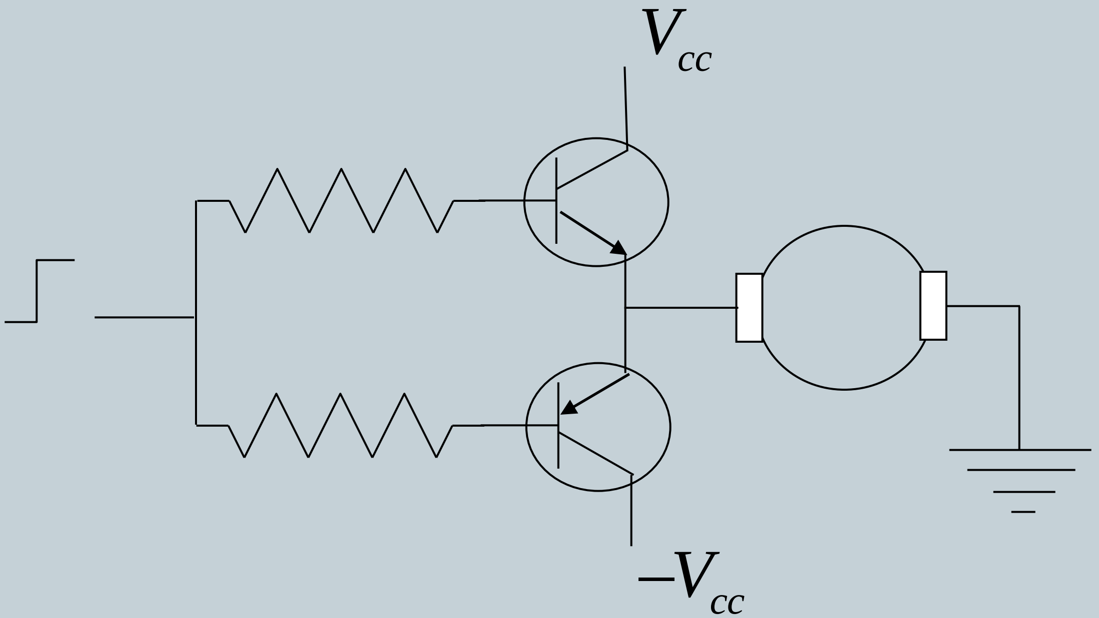
Let's consider the linear as a starting point



H-type

- The motor doesn't have a reference to ground (floating)
- It's difficult to get feedback signals (e.g. to measure the current flowing through the motor)

T-type



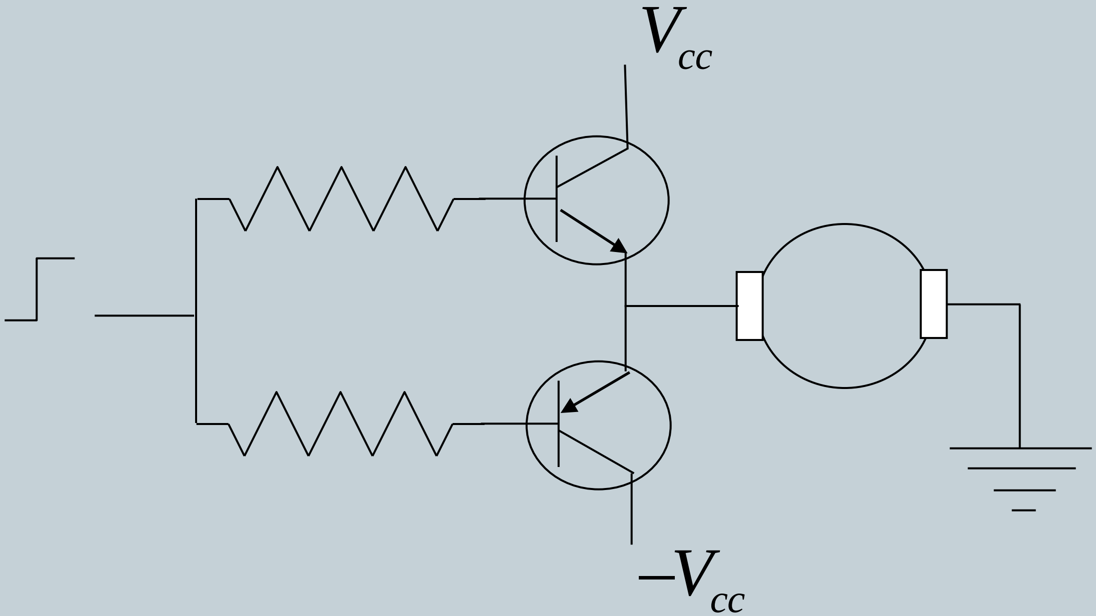
On the T-type

- Bipolar DC supply
- Dead band (around zero)
- Need to avoid simultaneous conduction (short circuit)

Things not shown

- Transistor protection (currents flowing back from the motor)
- Power dissipation and heat sink
  - Cooling
- Sudden stop due to obstacles
  - High currents → current limits and timeouts

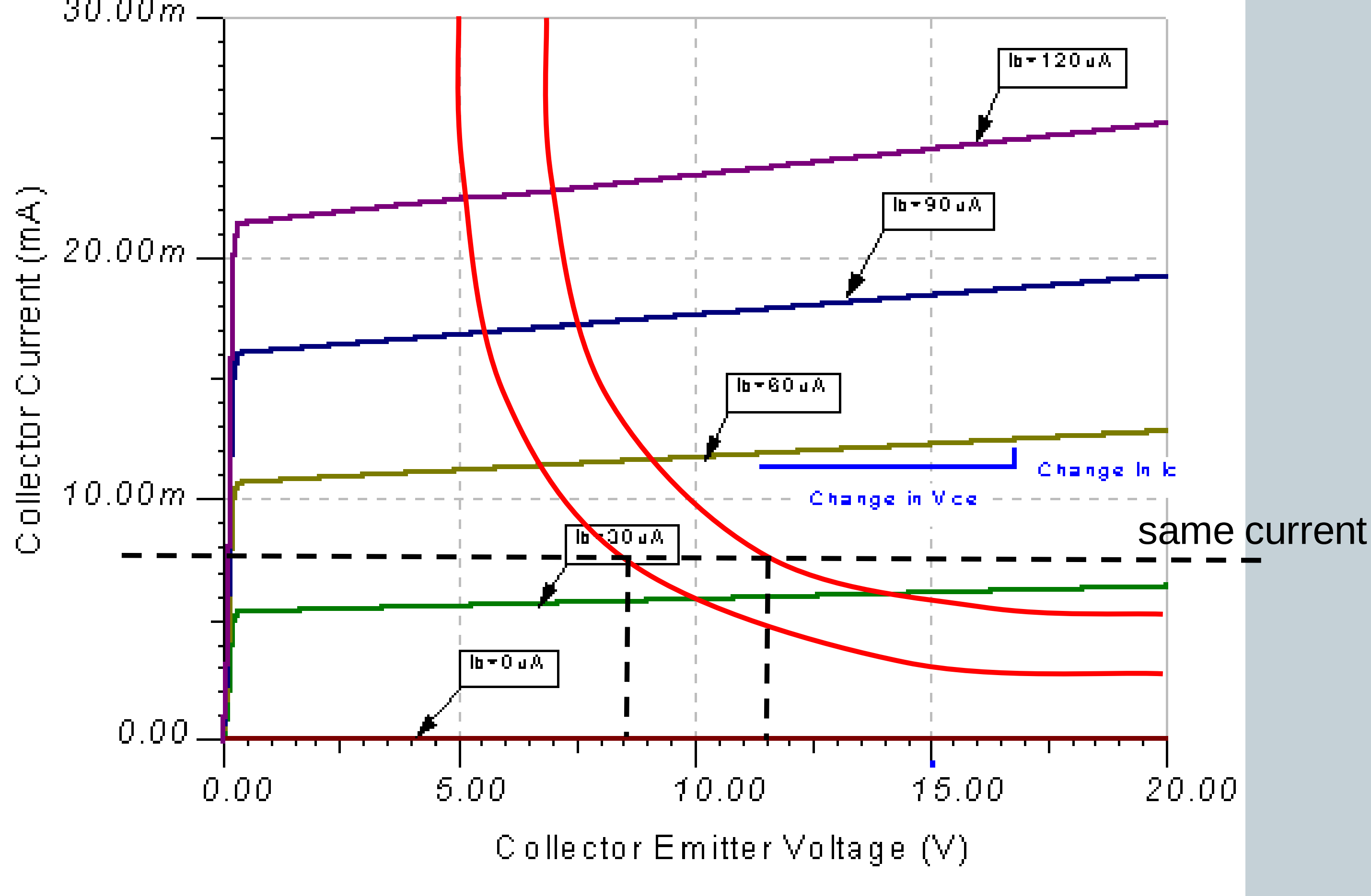
T-type



$$I_c \approx \frac{V_{cc}}{R_{transisor} + R_{motor}}$$

PWM amplifiers

$$P = V_{ce} I_c$$



PWM signal

$$P = V_{ce} I_c$$

- Transistors either “on” or “off”
  - When off, current is very low, little power too
  - When on, V is low, working point close to (or in) saturation, power dissipation is low

Comparison

- 12W for a 6A current using a switching amplifier
- 72W for a corresponding linear amplifier

Why does it work?

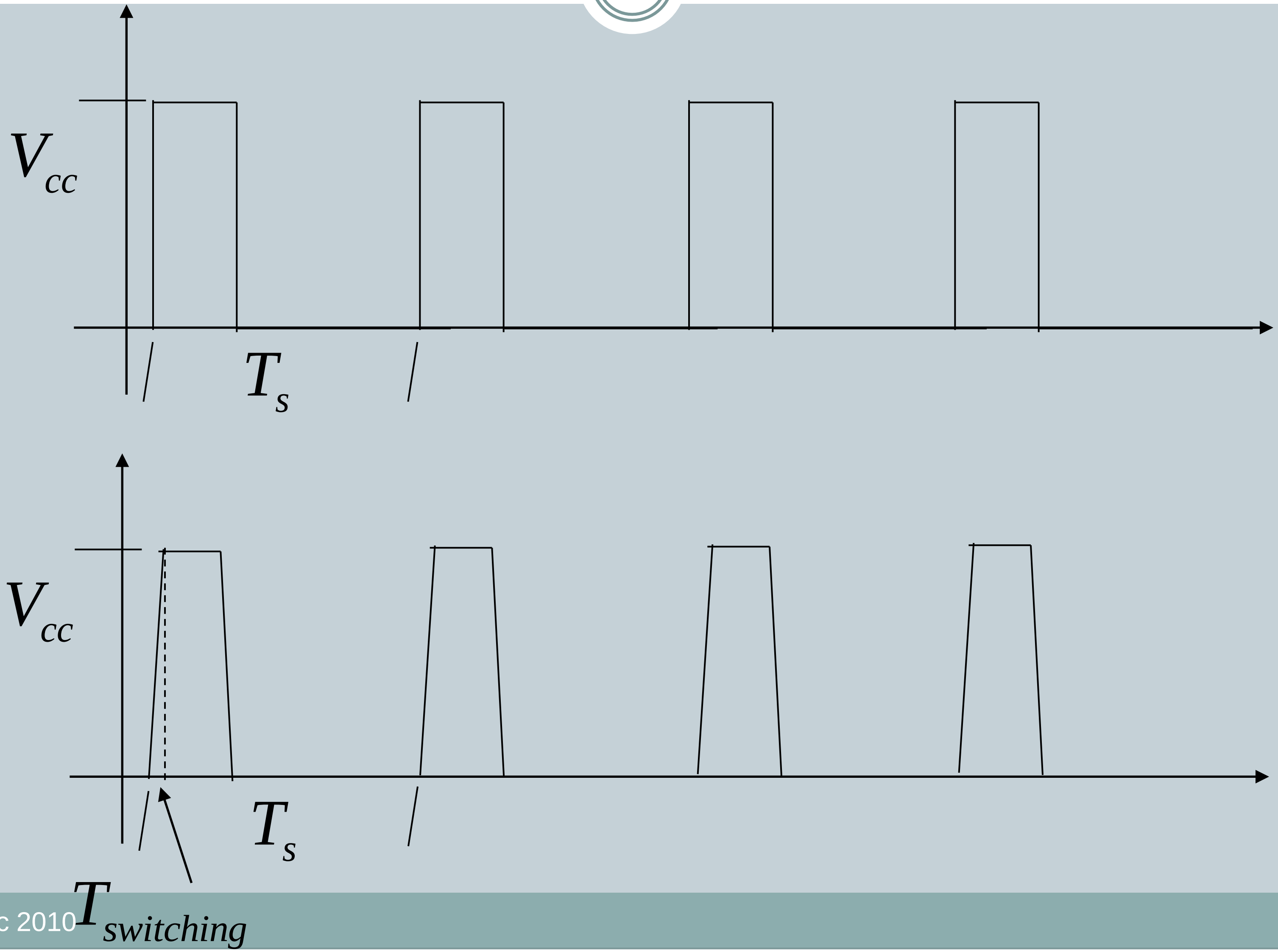
$$\frac{\omega(s)}{V_{arm}(s)} = \frac{K_T / L_a J_T}{s^2 + [(R_a J_T + L_a B)/L_a J_T]s + (K_T K_E + R_a B)/L_a J_T}$$

- In practice the motor transfer function is a low-pass filter

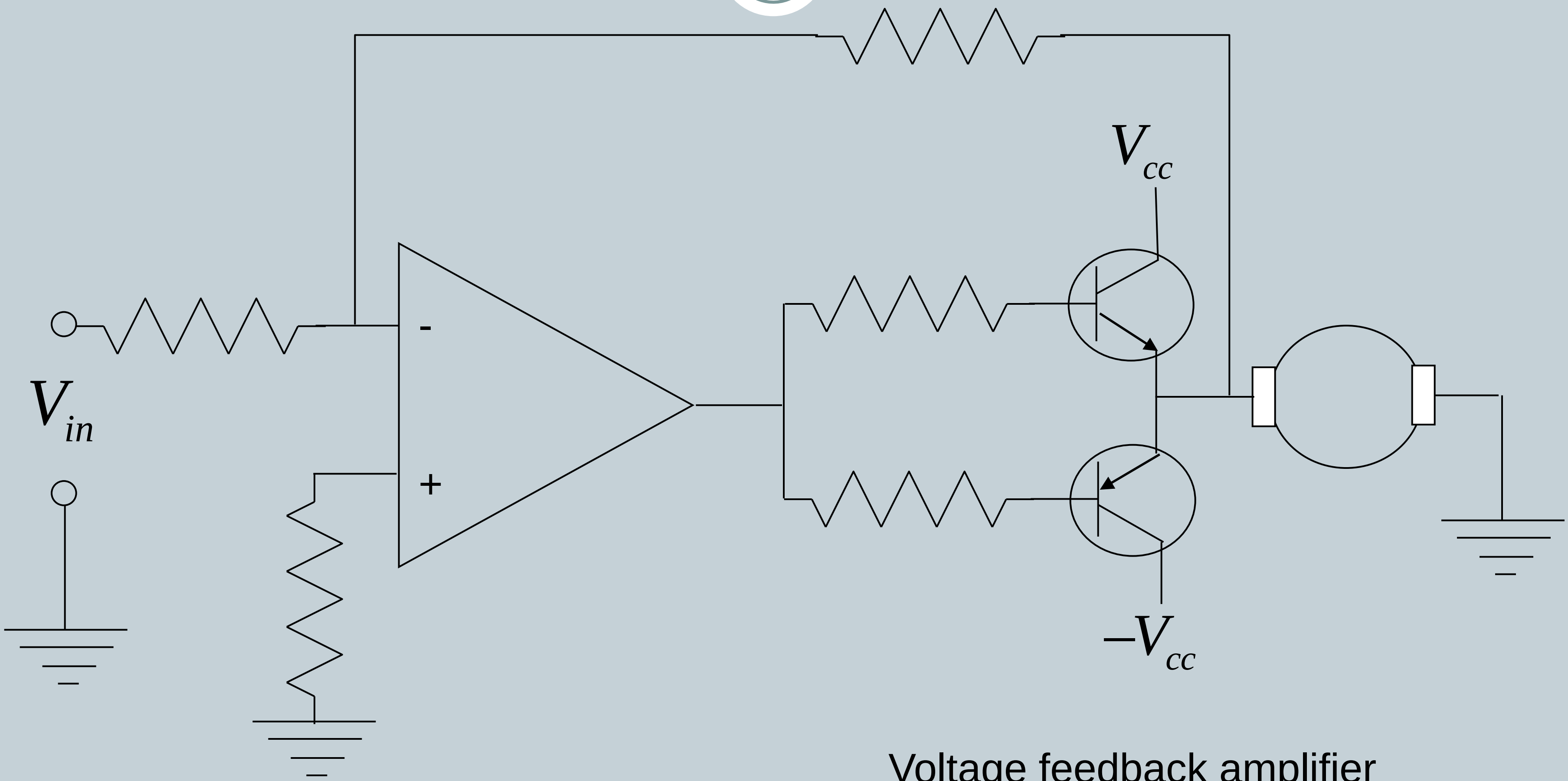
$$T_s \text{ with } f_s >> f_e \ (f_s > 100 f_e)$$

- Switching frequency must be high enough (s=switching, e=electric pole)

PWM signal

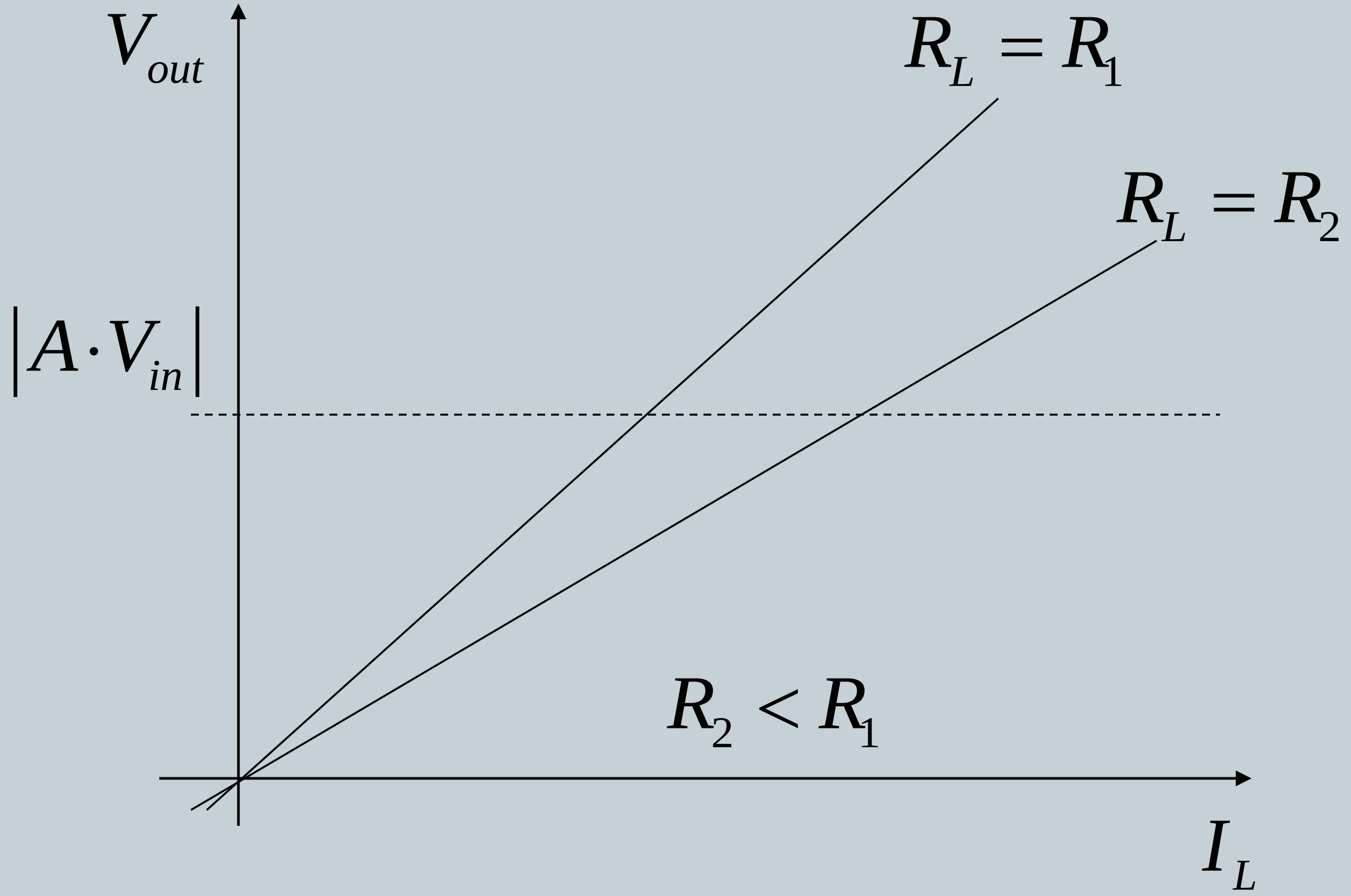


Feedback in servo amplifiers

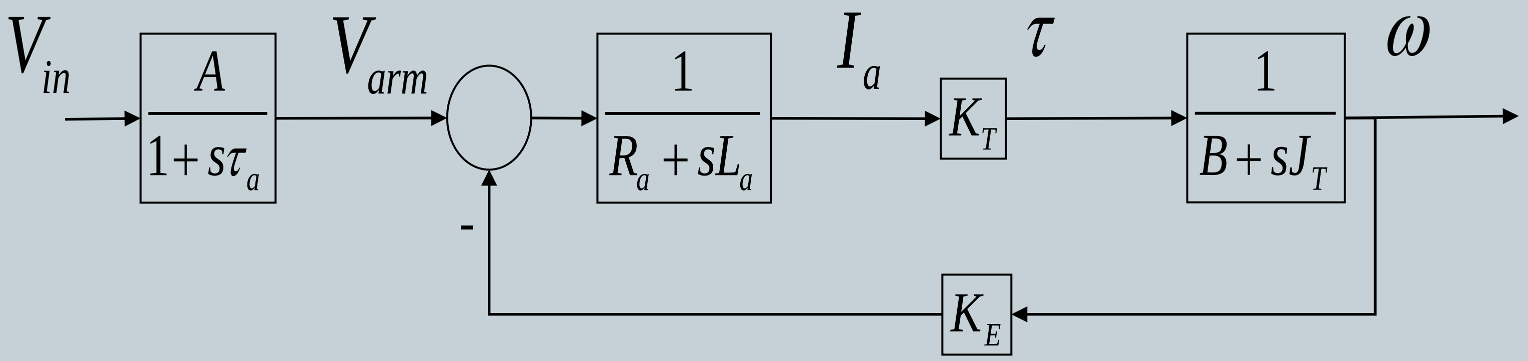


Voltage feedback amplifier

Operating characteristic

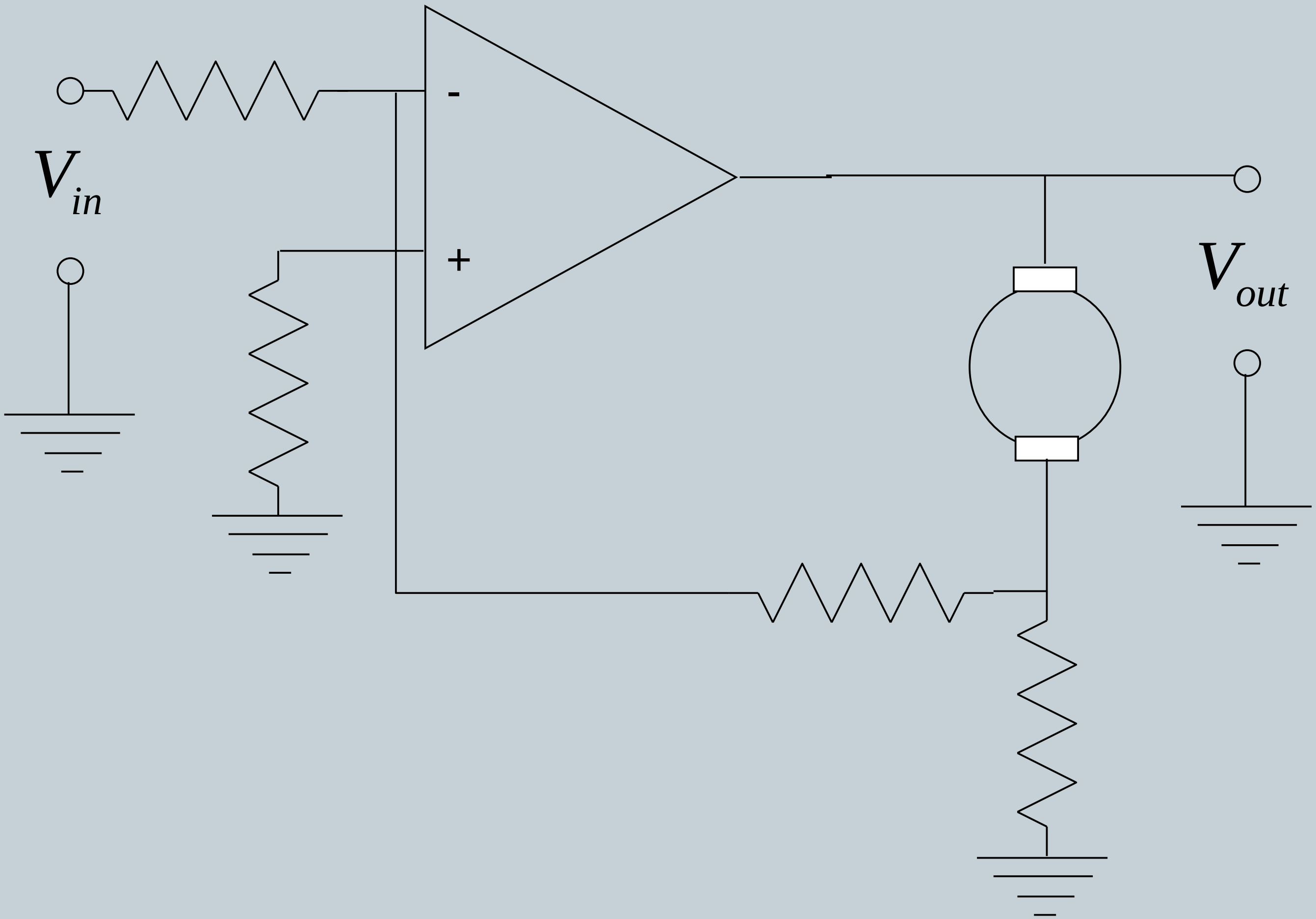


We've already seen this

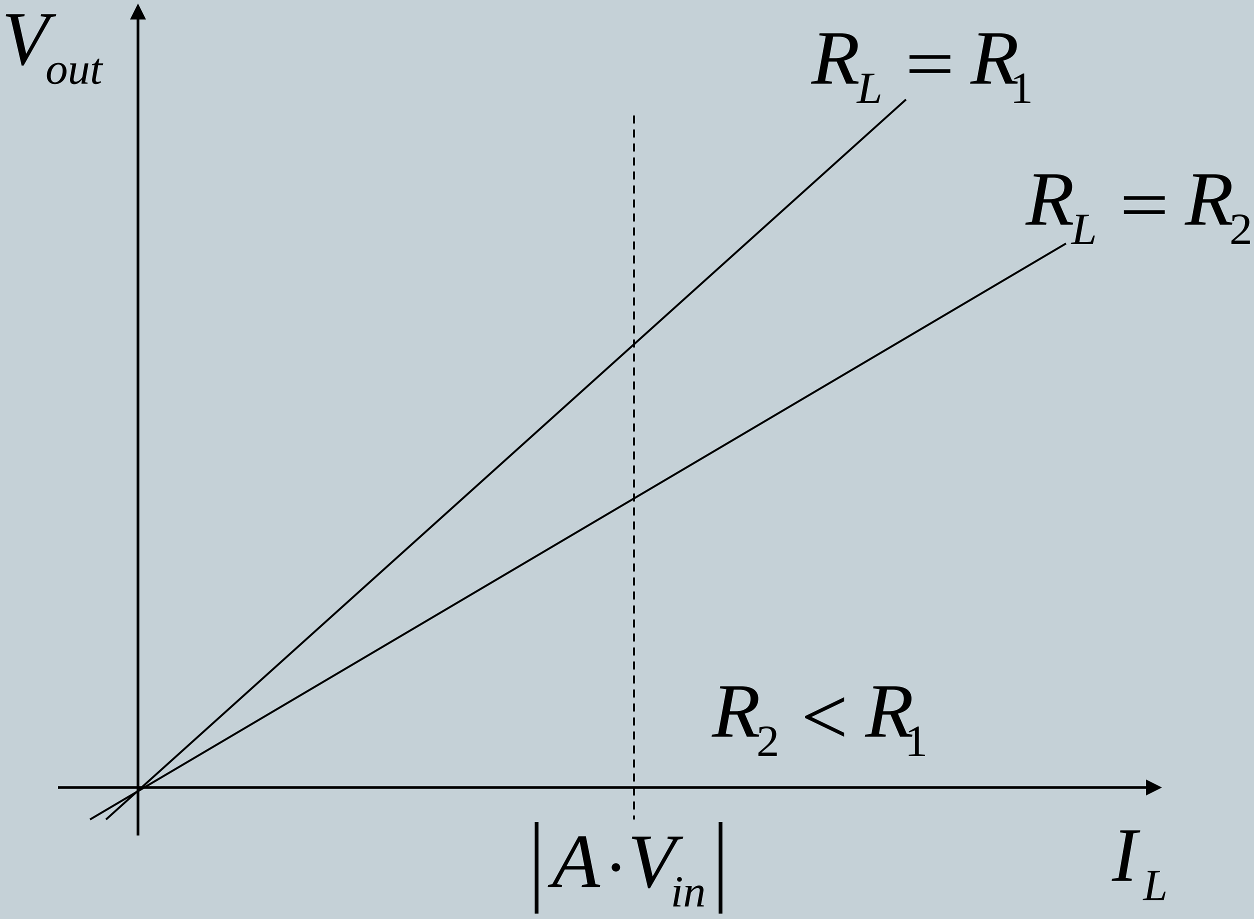


$$\frac{\omega(s)}{V_{in}(s)} = \frac{K_T/L_aJ_T}{s^2+[(R_aJ_T+L_aB)/L_aJ_T]s+(K_TK_E+R_aB)/L_aJ_T}\frac{A_v}{(1+s\tau_a)}$$

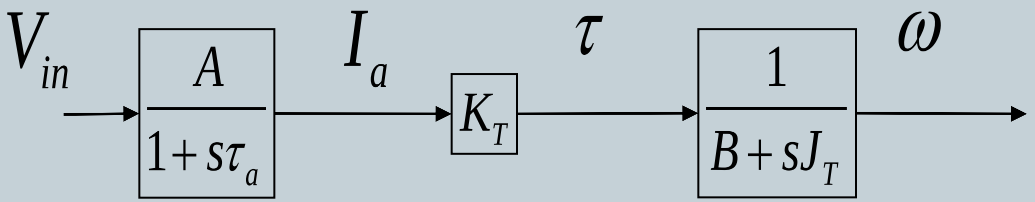
Current feedback



Current feedback

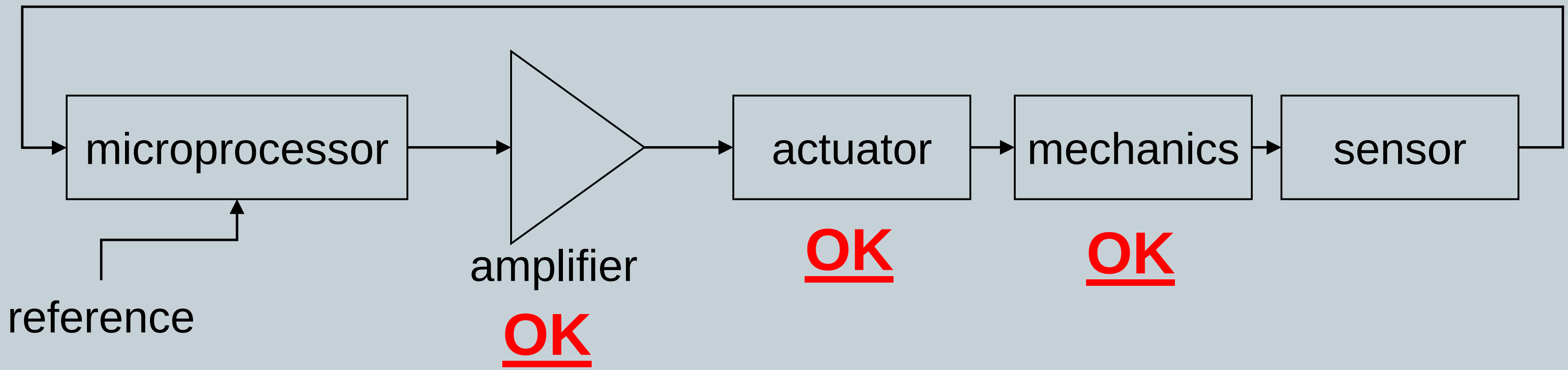


Motor driven by a current amplifier



$$\frac{\omega(s)}{V_{in}(s)} = \frac{K_T A_i}{(sJ_T + B)(1 + s\tau_a)}$$

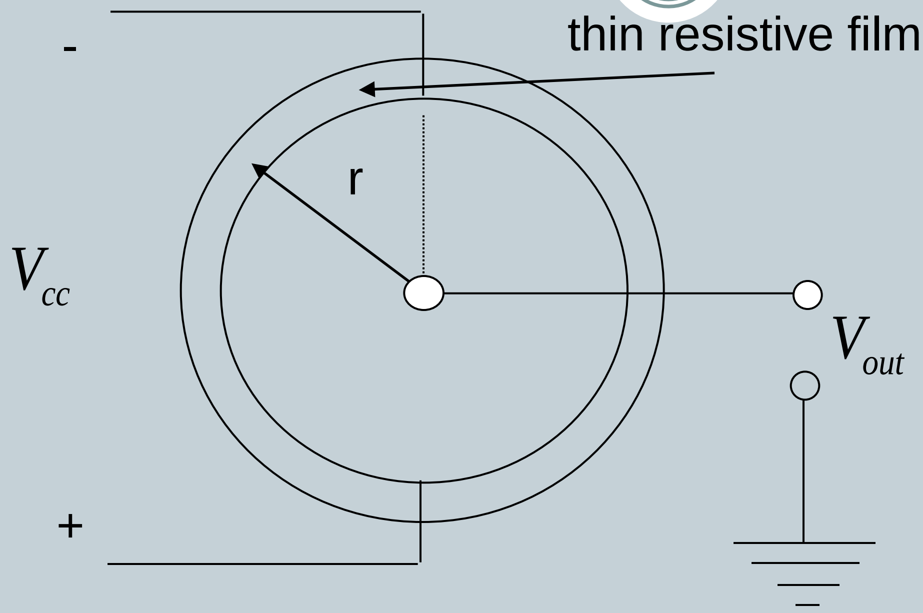
Back to the global view



Sensors

- Potentiometers
- Encoders
- Tachometers
- Inertial sensors
- Strain gauges
- Hall-effect sensors
- and many more…

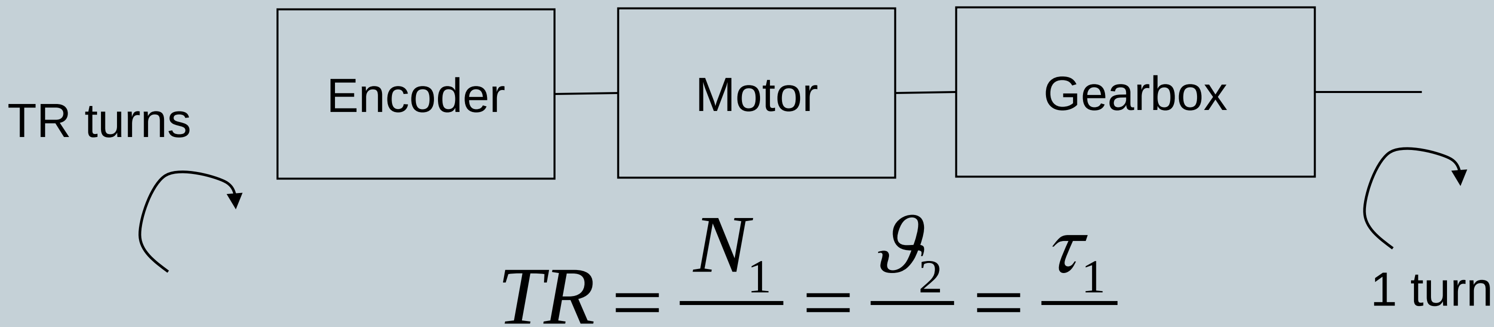
Potentiometer



$$V_{out} = \frac{r}{R} V_{cc}$$

- Simple but noisy
- Requires A/D conversion
- Absolute position (good!)

Note



$$TR = \frac{N_1}{N_2} = \frac{\vartheta_2}{\vartheta_1} = \frac{\tau_1}{\tau_2}$$

$$\tau_2 = \frac{N_2}{N_1}\tau_1 \Rightarrow (\text{most of the time})\, N_2 > N_1$$

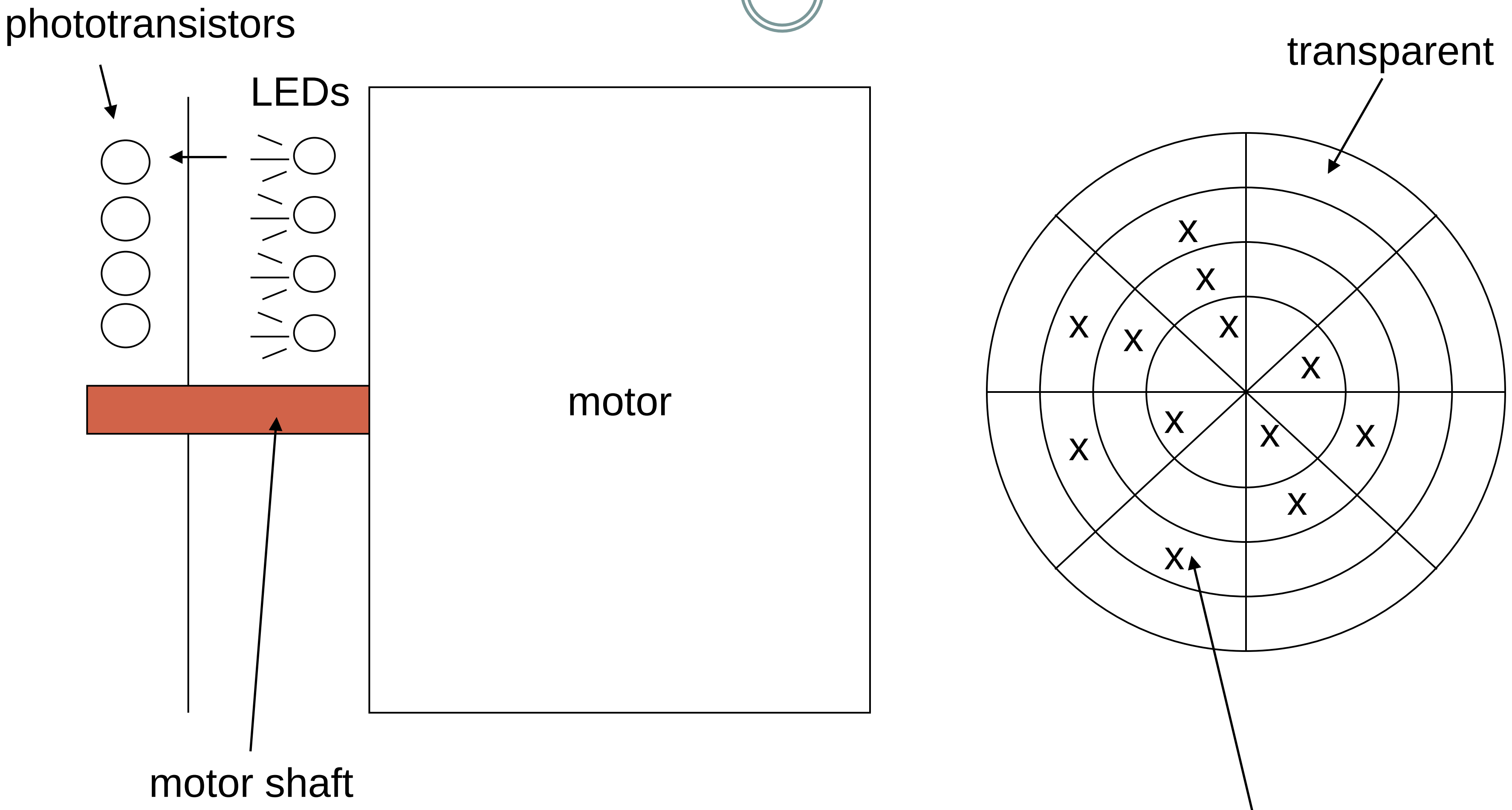
$$\vartheta_2 = \frac{N_1}{N_2}\vartheta_1$$

- The resolution of the sensor multiplied by TR

Encoder

- Absolute
- Incremental

Absolute encoder

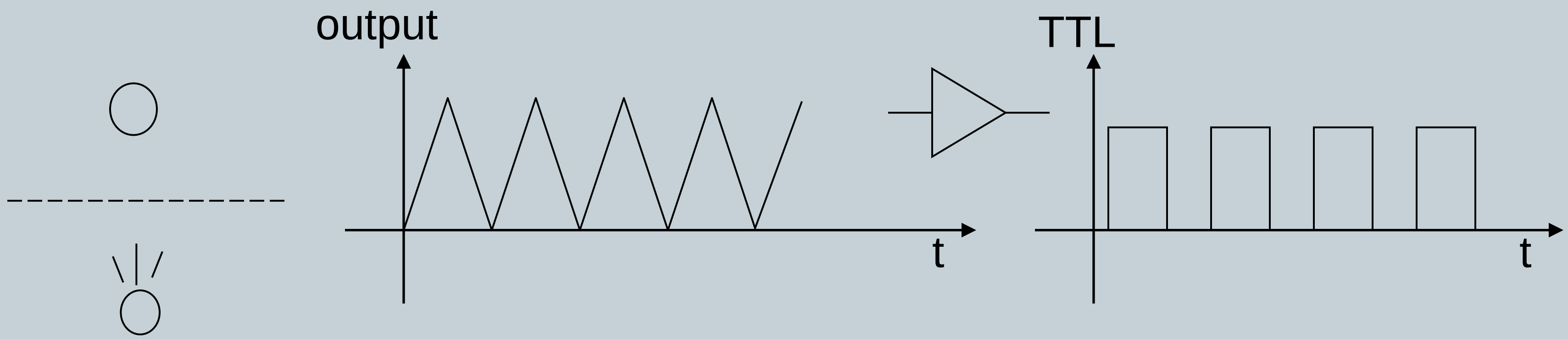


13 bits required for 0.044 degrees

Incremental encoder

- Disk single track instead of multiple
- No absolute position
- Usually an index marks the beginning of a turn

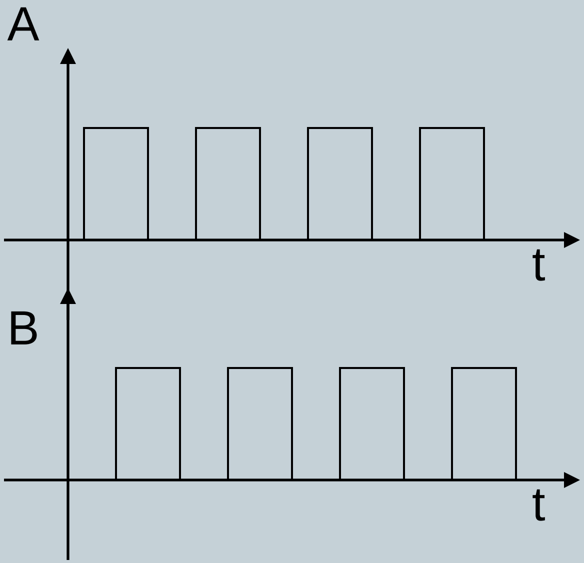
Incremental encoder

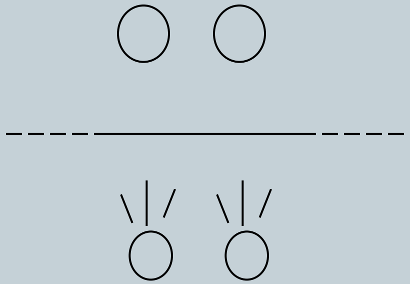


- Sensitive to the amount of light collected
- The direction of motion is not measured

Two-channel encoder

- 2 channels 90 degrees apart (quadrature signals) allow measuring the direction of motion





Moreover

- There are “differential” encoders
  - Taking the difference of two sensors 180 degrees apart
- Typically
  - A, B, Index channel
  - A, B, Index (differential)
- A “counter” is used to compute the position from an incremental encoder

Increasing resolution

- Counting UP and DOWN edges
  - X2 or X4 circuits

Absolute position

- A potentiometer and incremental encoder can be used simultaneously: the potentiometer for the "absolute" reference, and the encoder because of good resolution and robustness to noise

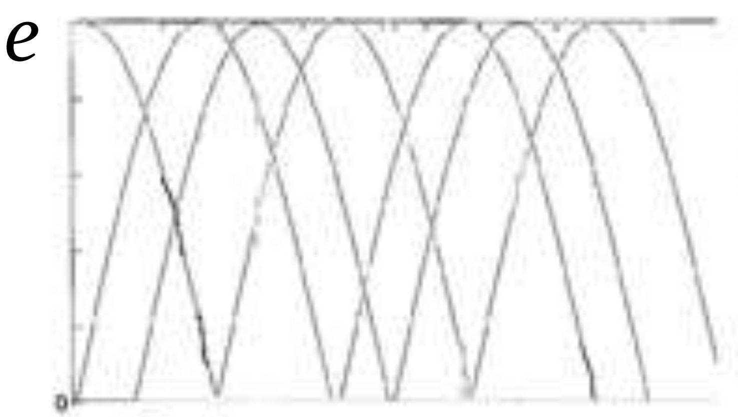
Analog locking

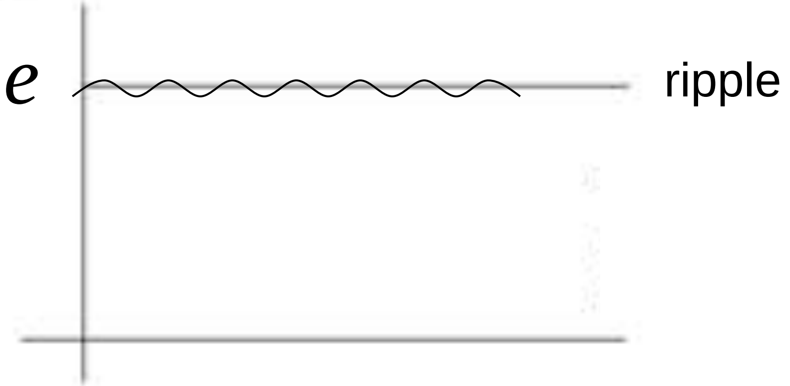
- Use digital encoder as much as possible
  - Get to zero error or so using the digital signal
- When close to zeroing the error:
  - Switch to analog: use the analog signal coming from the photodetector (roughly sinusoidal/triangular)
  - Much higher resolution, precise positioning

Tachometer

- Use a DC motor
  - The moving coils in the magnetic field will get an induced EMF
- In practice is better to design a special purpose “DC motor” for measuring velocity
- Ripple: typ. 3%

As already seen…





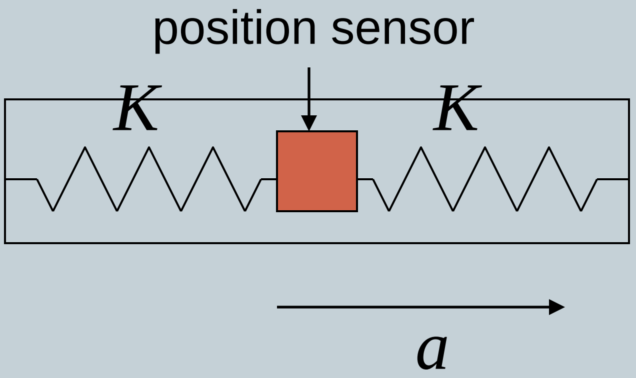
Measuring speed with digital encoders

- Frequency to voltage converters
  - Costly (additional electronics)
- Much better: in software
  - Take the derivative (for free!)

$$v(kT) = \frac{p(kT) - p((k-1)T)}{T}$$

Inertial sensors

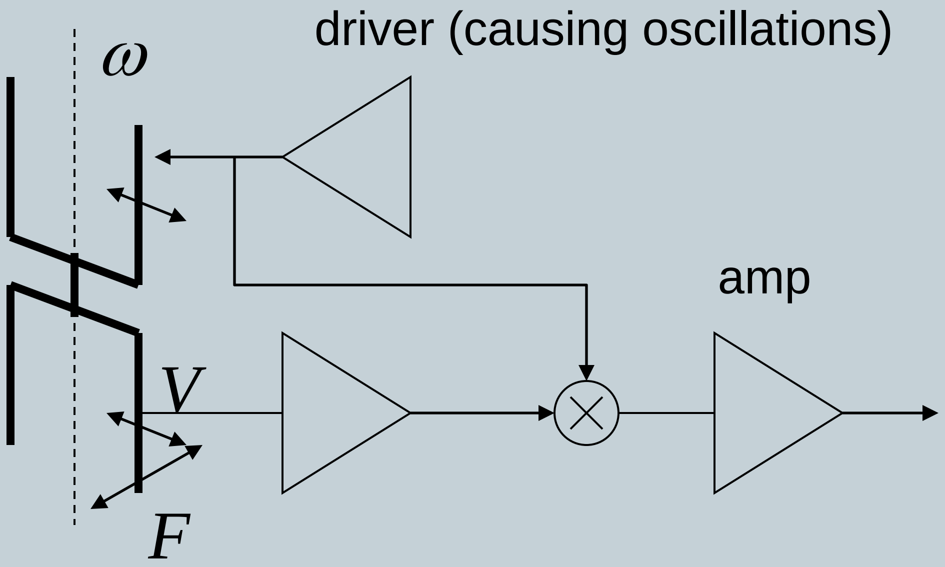
- Accelerometers:



$$Ma = 2Kx \Rightarrow a = \frac{2Kx}{M}$$

Gyroscopes

- Quartz forks



$$F = 2m\omega \times V$$

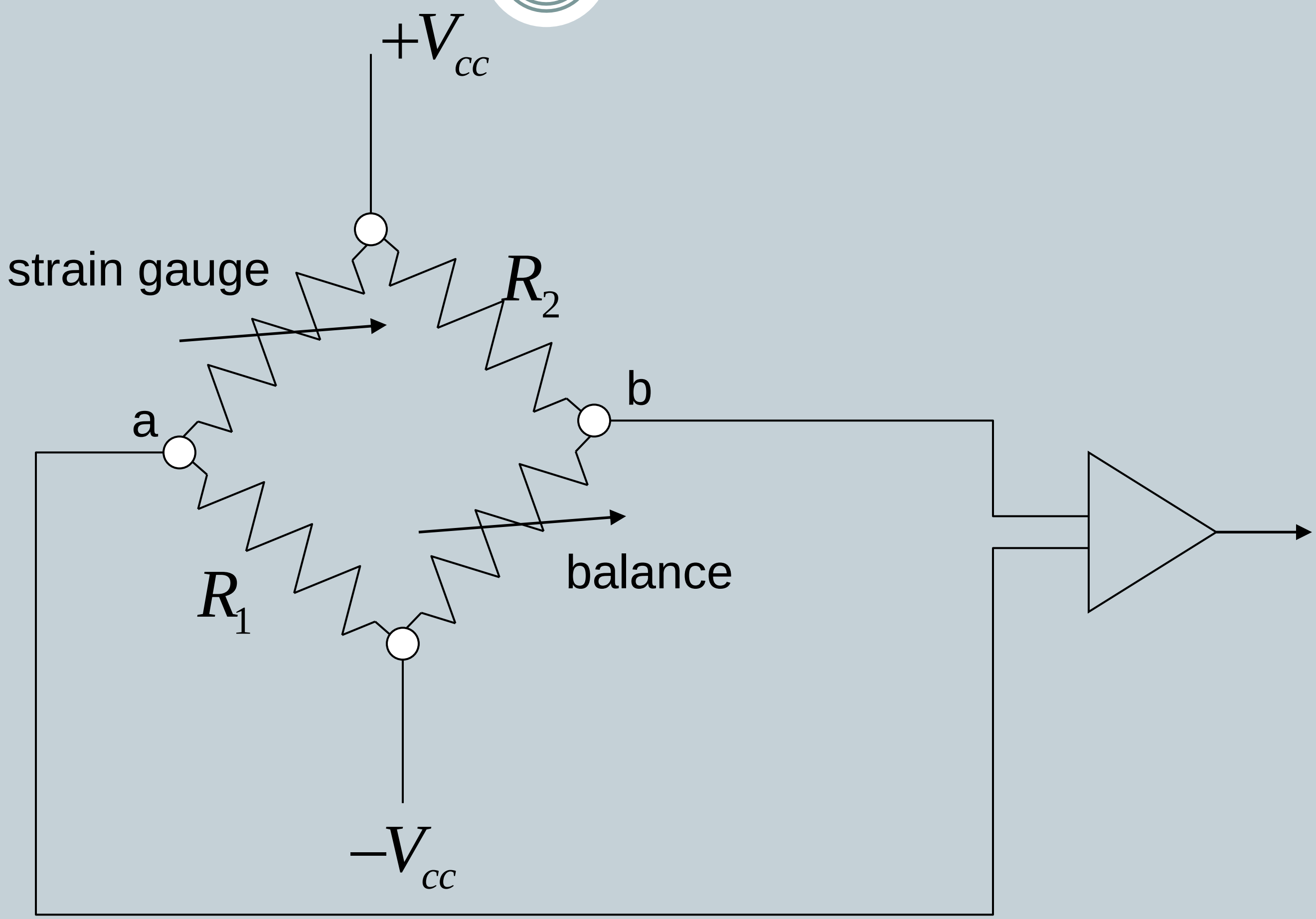
Strain gauges

- Principle: deformation →ΔR (resistance)
  - Example: conductive paint (Al, Cu)
  - The paint covers a deformable non-conducting substrate

$$R = \frac{L}{\sigma A} \Rightarrow \Delta L, A = const \Rightarrow \Delta R$$

conductivity (σ)

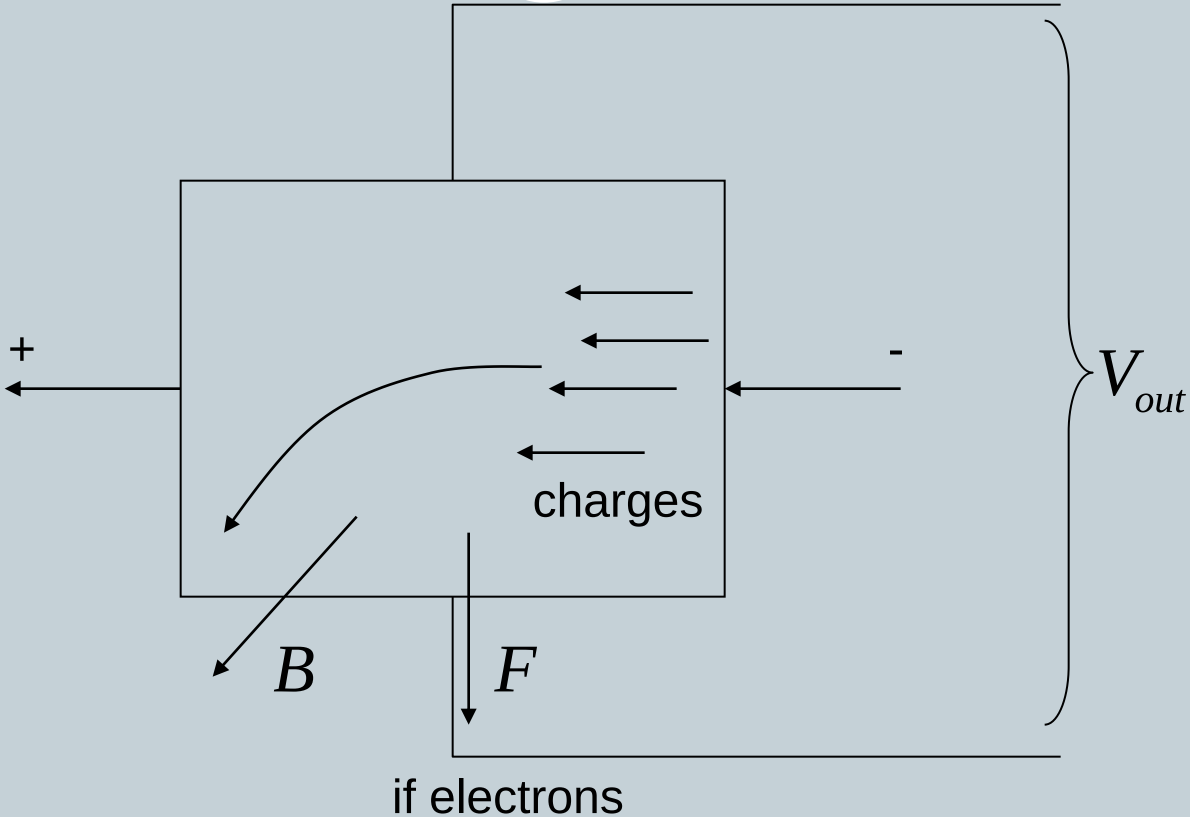
Reading from a strain gauge



$$R_1R_2 = R_gR_b \Rightarrow V_{ab} = 0$$

$$\Delta V_{ab} = f(\Delta R_g)$$

Hall-effect sensors



$$F_{lorentz} = q\vec{v} \times \vec{B}$$

Example

- Measuring angles (magnetic encoders)

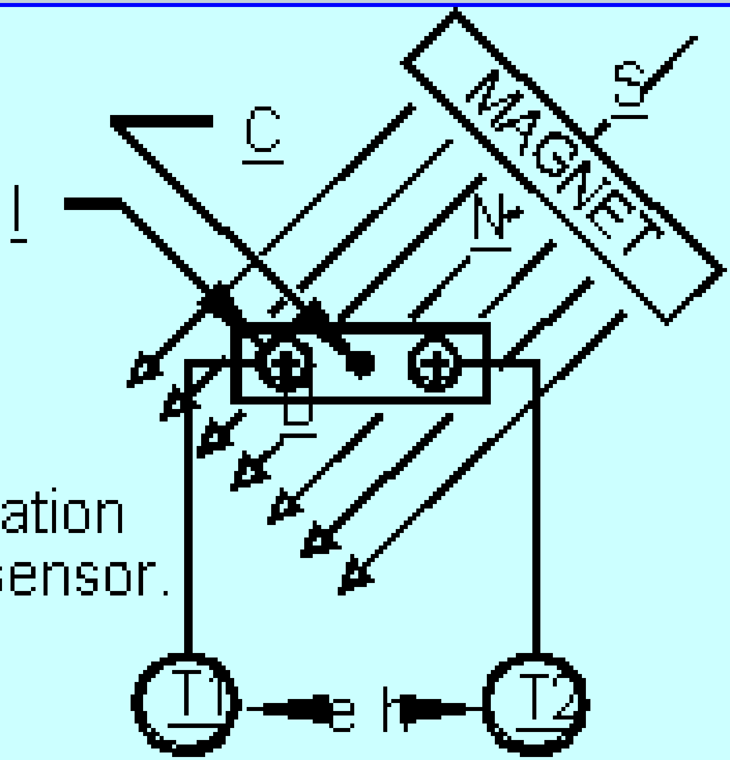
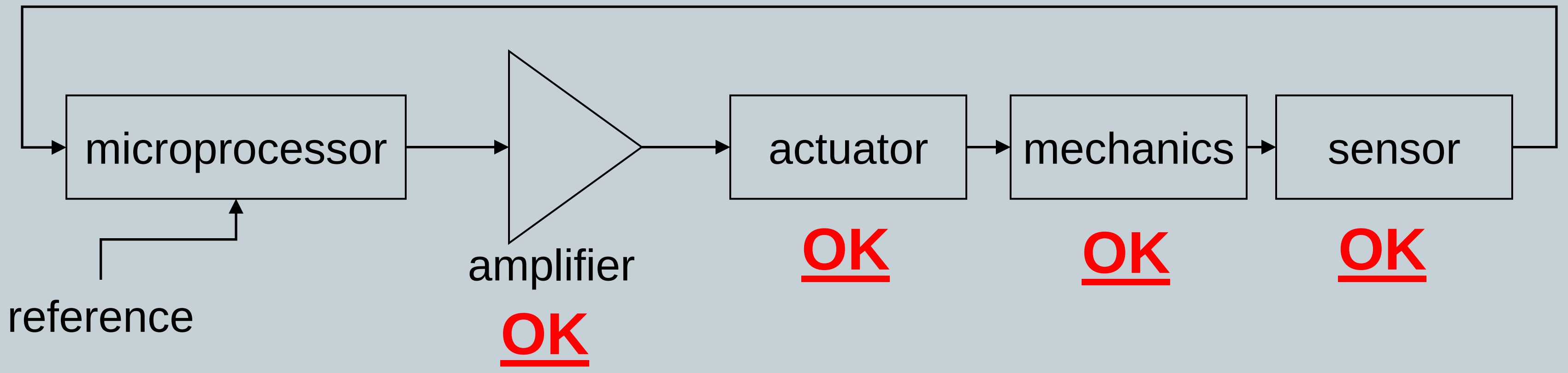


FIGURE 1--Basic Configuration of magnet and hall effect sensor. Edge views are shown.

Back to the global view



Microprocessors

- Special DSPs for motion control
  - Some are barely programmable (the control law is fixed)
  - Others are general purpose and they are mixed mode (analog and digital in a single chip)

Example

- DSP 16 bit ALU and instruction set
- PWM generator (simply attach this to either T or H amplifier)
- A/D conversion
- CAN bus, Serial ports, digital I/O
- Encoder counters
- Flash memory and RAM on-board
- Enough of all these to control two motors (either brush- or brushless)

Kinematics

- Kinematics:
  - Given the joint angles, compute the hand position

$$\mathbf{x} = \Lambda(\mathbf{q})$$

- Inverse kinematics:
  - Given the hand position, compute the joint angles to attain that position

$$\mathbf{q} = \Lambda^{-1}(\mathbf{x})$$

- As usual, inverse problems might be troublesome!

Kinematics

- Inverting:
  - Geometrically: closed form solution exists in certain cases
  - By minimization:

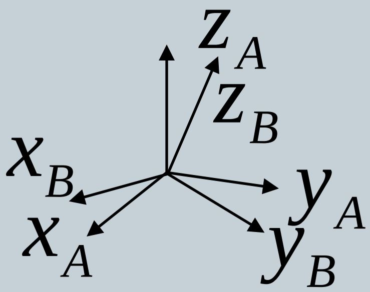
$$J = \frac{1}{2}\left\|\mathbf{x} - \Lambda(\mathbf{q})\right\|^2 \Rightarrow \mathbf{q} = \arg\min_{\mathbf{q}} J$$

- Kinematic redundancy: more joints than constraints
  - E.g. a rigid body (hand) in space is described by 6 numbers (position + orientation). A robot (or human) arm might have 7 or more joints (degrees of freedom)

Representing kinematics

- Representing rotations and translations between coordinate frames of reference

$$^{A}v = [?]\,^{B}v$$



$$^{A}v = [\,^{A}x_B \mid \,^{A}y_B \mid \,^{A}z_B\,]\,^{B}v = \,^{A}R_B\,^{B}v \qquad B \rightarrow A$$

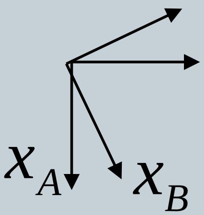
$$^{A}x_B = \,^{A}R_B\,^{B}x_B = \,^{A}R_B[1,0,0]^T$$

Rotation matrix

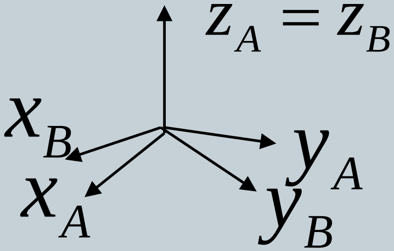
$$^{A}R_{B}(^{A}R_{B})^{T} = I \Leftrightarrow (^{A}R_{B})^{T} = (^{A}R_{B})^{-1} = {}^{B}R_{A}$$

Orthogonal matrix

Example: rotation along the Z axis $\begin{bmatrix} \cos\vartheta & -\sin\vartheta & 0 \\ \sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 1 \end{bmatrix}$



$$^{A}x_{B} = \begin{bmatrix} \cos\vartheta \\ \sin\vartheta \\ 0 \end{bmatrix}$$



More simple rotations

Example: rotation along the Y axis

$$\begin{bmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\vartheta & -\sin\vartheta \\ 0 & \sin\vartheta & \cos\vartheta \end{bmatrix}$$

Example: rotation along the X axis

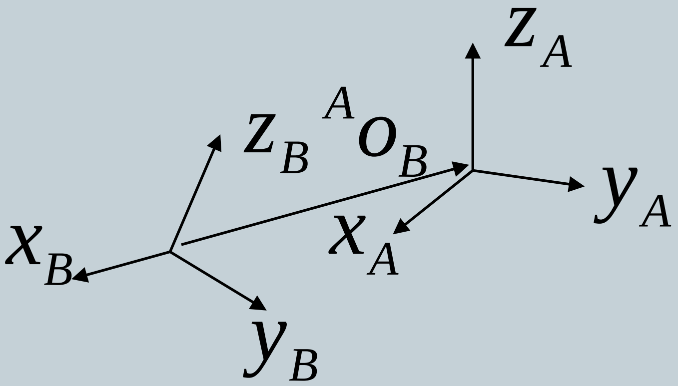
Representing 3D rotations

- Sequences of elementary rotations
  - Euler angles: z, y, z or z, x, z
  - Roll, pitch, yaw angles: z, y, x
  - Vector (axis of rotation) and angle

Roto-translation

- Rotation combined with translation

$$^{A}v = {}^{A}R_{B}\,{}^{B}v + {}^{A}o_{B}$$



Homogeneous representation

- To make things uniform

$$ {}^{A}v = {}^{A}R_{B}\,{}^{B}v + {}^{A}o_{B} $$

$$ \begin{bmatrix} {}^{A}v \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{A}R_{B} & {}^{A}o_{B} \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^{B}v \\ 1 \end{bmatrix} $$

$$ {}^{A}v = {}^{A}T_{B}\,{}^{B}v \qquad \dim(v) = 4 $$

Clearly

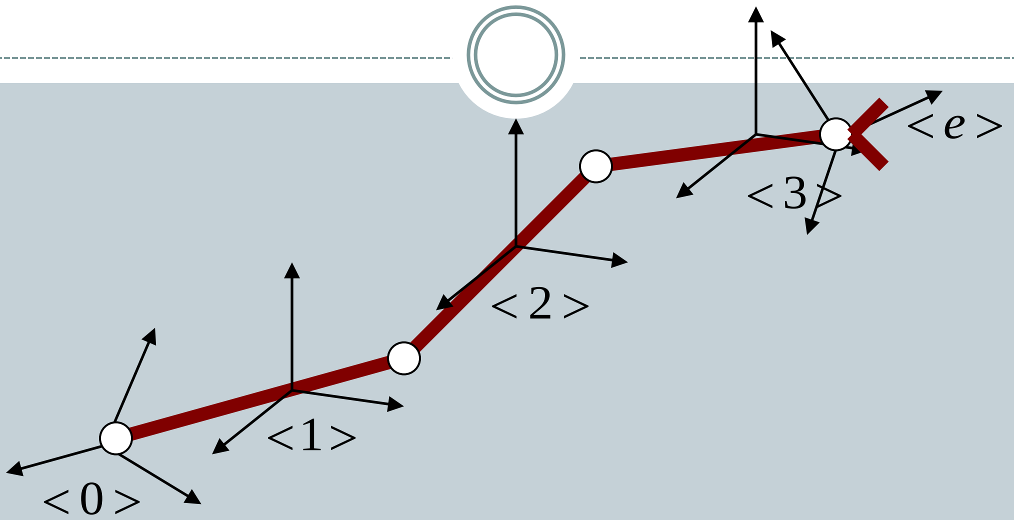
- Composition of transforms
- Inverse of a rototranslation

$$^{A}v = {}^{A}T_{B}\,{}^{B}T_{C}\,{}^{C}v \qquad C \rightarrow A$$

$$\begin{bmatrix} {}^{A}R_{B} & {}^{A}o_{B} \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} {}^{A}R_{B}^{T} & -{}^{A}R_{B}^{T}\,{}^{A}o_{B} \\ 0 & 1 \end{bmatrix}$$

$${}^{A}T_{B}^{\;-1} = {}^{B}T_{A}$$

Direct kinematics



$$ {}^{0}T_1(q_1)\cdots{}^{n-1}T_n(q_n) $$

$$ (x, y, z) = {}^{0}T_e(q_1, q_2, q_3, q_4)\cdot(0,0,0)^T $$

$$ \mathbf{x} = \Lambda(\mathbf{q}) $$

$$ \textbf{orientation} = \tilde{\Lambda}(\mathbf{q}) $$

Conventions

- For placing the reference frames on each link
  - Denavit-Hartenberg
- Many times DH parameters are given for a manipulator (and various useful equations are also given wrt DH convention)

Inverse kinematics

- Direct approach
- Geometric
- Minimization
- Neural network, learning

Inverse kinematics

- Direct approach
  - Try solving:

$$x = NL_x(q_1, q_2, q_3, q_4)$$
$$y = NL_y(q_1, q_2, q_3, q_4)$$
$$z = NL_z(q_1, q_2, q_3, q_4)$$

for q_1, q_2, q_3, q_4

Geometric approach

- For certain manipulator the solution exists in close form
  - Decomposable structures (e.g. translation and rotations can be handled separately)
  - Rotations follow certain rules
- Many industrial manipulators were designed with inverse kinematics in mind

Minimization

- Find the solution to:

$$J = \frac{1}{2}\left\|\mathbf{x} - \Lambda(\mathbf{q})\right\|^2 \Rightarrow \mathbf{q} = \arg\min_{\mathbf{q}} J$$

- Neural network/learning:

$$(\mathbf{q}, \mathbf{x}) \rightarrow \Lambda^{-1}$$

- Approximate the inverse out of a family of functions (NN approach) starting from examples

What about velocity?

- Jacobian matrix

$$\mathbf{x} = \Lambda(\mathbf{q}) \Rightarrow \frac{d\mathbf{x}}{dt} = \begin{bmatrix} \frac{dx_1}{dq_1} & \cdots & \frac{dx_1}{dq_m} \\ \vdots & \ddots & \vdots \\ \frac{dx_n}{dq_1} & \cdots & \frac{dx_n}{dq_m} \end{bmatrix} \cdot \frac{d\mathbf{q}}{dt}$$

$$\frac{d\mathbf{x}}{dt} = J(\mathbf{q}) \cdot \frac{d\mathbf{q}}{dt}$$

Note on representing velocities

- If **x** is:

$$\mathbf{x} = (x, y, z, \vartheta, \varphi, \psi)$$

  - Position + Euler angles

$$\mathbf{v} = (v_x, v_y, v_z, \dot{\vartheta}, \dot{\varphi}, \dot{\psi})$$

  - Euler angles derivatives do not have any clear physical meaning

$$\mathbf{v} = (v_x, v_y, v_z, \boldsymbol{\omega})$$

  - Angular velocity (rate of rotation along the axis

Anyway…

- Just make sure the representation and the equations are consistent

$$\mathbf{v} = (v_x, v_y, v_z, \dot{\vartheta}, \dot{\varphi}, \dot{\psi}) \Rightarrow J_r$$

$$\mathbf{v} = (v_x, v_y, v_z, \boldsymbol{\omega}) \Rightarrow J_v$$

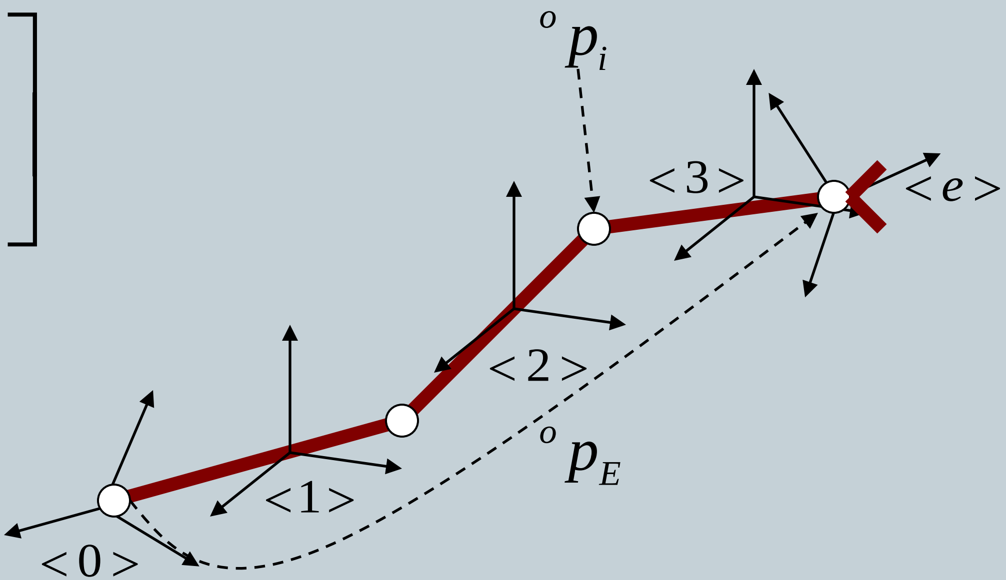
Jacobian

- Formula
  - Given the DH representation of transformations
  - Considering only rotational joints

$$J_v = [J_1 \mid J_2 \cdots J_n] \quad \text{for } n \text{ joints}$$

$$J_i = \begin{bmatrix} {}^o z_i \times {}^o p_{E,i} \\ {}^o z_i \end{bmatrix}$$

$${}^o p_{E,i} = {}^o p_E - {}^o p_i$$



Having written

$$ {}^{0}T_i = \begin{bmatrix} {}^{0}x_i & {}^{0}y_i & {}^{0}z_i & {}^{0}p_i \\ 0 \quad 0 \quad 0 & & & 1 \end{bmatrix} $$

$$ {}^{0}T_i = {}^{0}T_1 \, {}^{1}T_2 \cdots {}^{i-1}T_i $$

When J is invertible

- Can compute the joint velocities to obtain a certain hand velocity

$$\dot{\mathbf{q}} = J^{-1}\dot{\mathbf{x}}$$

- If n>6, redundancy:

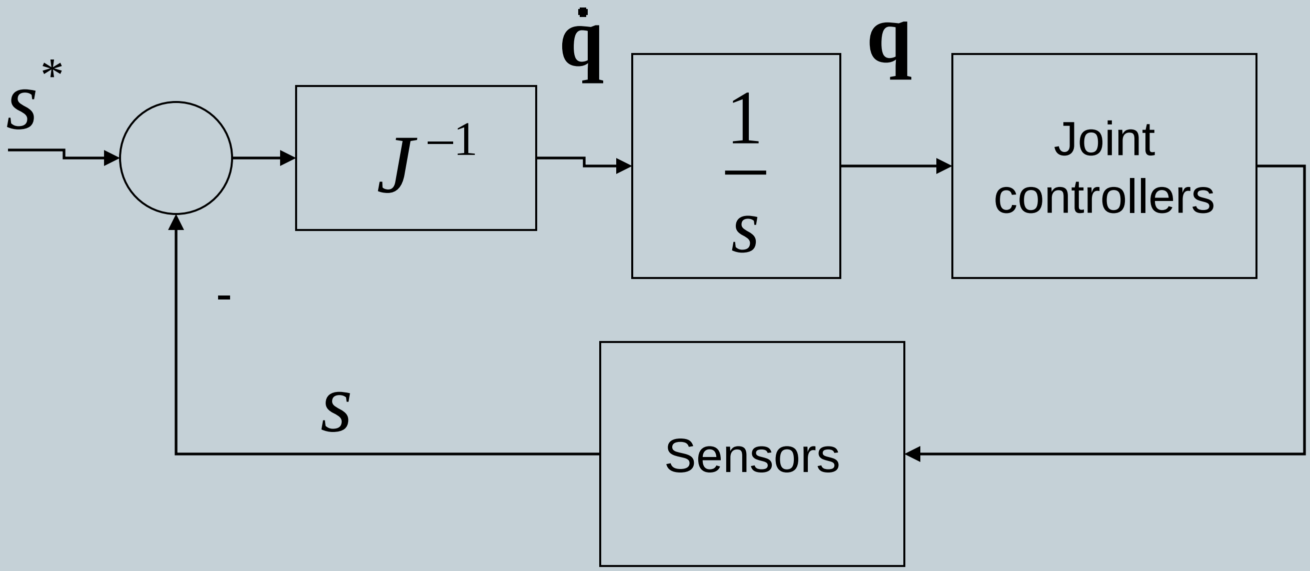
$$\dot{\mathbf{q}} = J^{+}\dot{\mathbf{x}} + (I - J^{+}J)\mathbf{k}$$

- **k** is a constant vector

Troubles

- Even if n≤6 there are many situations where J cannot be inverted (singularities)
  - Movement singularities (chain of rotations)
  - J not invertible because certain elements go to zero

Resolved rate controller



Static

- Relationship between forces and torques

$$d\mathbf{x} = J d\mathbf{q}$$

$$d\mathbf{q}^T \boldsymbol{\tau} = d\mathbf{x}^T \mathbf{F}$$

$$d\mathbf{q}^T \boldsymbol{\tau} = d\mathbf{q}^T J^T \mathbf{F}$$

$$\Downarrow$$

$$\boldsymbol{\tau} = J^T \mathbf{F}$$

- Imagining the integrals where appropriate

Another idea

$$\boldsymbol{\tau} = J^T \mathbf{F}$$

- Use this equation to design a force controller:
  - Given F compute the torques to drive the joints

Dynamics

- Two methods to derive the equation of motion (differential equations)
  - Newton-Euler
  - Lagrange formalism

Newton-Euler

- Start from:

$$\begin{cases} \mathbf{F} = \dfrac{d}{dt}(m\mathbf{v}) \\ \boldsymbol{\tau} = \dfrac{d}{dt}(I\boldsymbol{\omega}) \end{cases}$$

$$\begin{cases} \mathbf{F} = \dfrac{d}{dt}(m\mathbf{v}) \\ \boldsymbol{\tau} = \dfrac{d}{dt}(I\boldsymbol{\omega}) = \boldsymbol{\omega} \times (I\boldsymbol{\omega}) + I\dot{\boldsymbol{\omega}} \\ \text{kinematics} \end{cases}$$

⇨ Write down every equation (6): find the angular velocity and *I* with respect to a base frame

Lagrange formulation

- Lagrange equations:

$$\begin{cases} L = K - P \\ \sum_{\mu} F_{\mu} \frac{\partial x_{\mu}}{\partial q_i} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \end{cases}$$

$$x_{\mu} = x_{\mu}(q_1 \cdots q_N, t)$$

External forces (no potential)

$$K = \frac{1}{2} m \mathbf{v}^T \mathbf{v} + \frac{1}{2} \boldsymbol{\omega}^T I \boldsymbol{\omega}$$

For a manipulator

- Take the joint angles as variable, write the position *x* of the links, write down K, P and the external forces

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + h(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q})$$

External forces (control)

Inertia (generalized)

Coriolis, centrifugal effects

Gravity

Complexity

- Newton-Euler: $o(n)$
- Lagrange: $o(n^4)$

Estimation

- Kinematics → just measure the params
- Dynamics → estimate from data

Dynamics

- Direct dynamics:

$$\tau(t) \rightarrow q(t)$$

- Simulation (integrate the equations – Runge-Kutta, Euler, etc.)

- Inverse dynamics:

$$q(t) \rightarrow \tau(t)$$

Dynamics and control

- Case 1: parameters are such that feedback gain at each joint is >> gravity, Coriolis, centrifugal, disturbances, etc.

- Case 2: feedback in not enough for high-speed, precision, etc. → compensation is required

Case 1

- Approx behavior:

$$A\ddot{\mathbf{q}} + B\dot{\mathbf{q}} + k[\mathbf{q} - \mathbf{q}^*] = 0$$

- Can design k or a PID controller to make this system behave as desired

Case 2

- Let's imagine we know all the parameters with a certain precision:

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + h(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q})$$

$$\boldsymbol{\tau}_{control} = M(\mathbf{q})\mathbf{u} + h(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q})$$

$$M(\mathbf{q})\ddot{\mathbf{q}} + h(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) = M(\mathbf{q})\mathbf{u} + h(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q})$$

$$M(\mathbf{q})\ddot{\mathbf{q}} = M(\mathbf{q})\mathbf{u}$$

$$\mathbf{u} = \ddot{\mathbf{q}}^* + k_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) + k_p(\mathbf{q}^* - \mathbf{q})$$

Case 2 (continued)

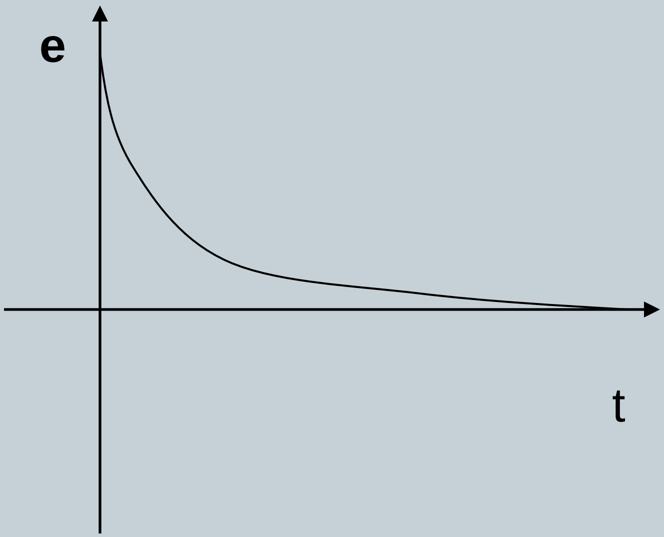
$$\ddot{\mathbf{q}} = \mathbf{u}$$

$$\mathbf{u} = \ddot{\mathbf{q}}^* + k_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) + k_p(\mathbf{q}^* - \mathbf{q})$$

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}^* + k_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) + k_p(\mathbf{q}^* - \mathbf{q})$$

$$\mathbf{e} = \mathbf{q}^* - \mathbf{q}$$

$$0 = \ddot{\mathbf{e}} + k_d\dot{\mathbf{e}} + k_p\mathbf{e}$$



- Appropriate design of the gains can get arbitrary exponential behavior of the error