上海科技大学
ShanghaiTech University

CS289: **Mobile Manipulation Fall 2025**

Sören Schwertfeger

ShanghaiTech University

Kinematics: Velocities

Cartesian Space

Tool Frame (E)

(aka End-Effector)

Base Frame (B)

$${}^{B}_{E}V = \begin{Bmatrix} {}^{B}_{E}v \\ {}^{B}_{E}w \end{Bmatrix}$$

v: linear velocity

w: angular velocity

Jacobian

$${}^{B}_{E}V = J(q)\dot{q}$$

Inverse Jacobian

$$\dot{q} = J^{-1}(q)\ {}^{B}_{E}V$$

Joint Space

Joint 1 = $\dot{q}_1$

Joint 2 = $\dot{q}_2$

Joint 3 = $\dot{q}_3$

...

Joint n = $\dot{q}_n$

Rigid body transformation

Between coordinate frames

Linear algebra

Written in Matrix Form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_6}{\partial q_1} & \cdots & \frac{\partial f_6}{\partial q_6} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}$$

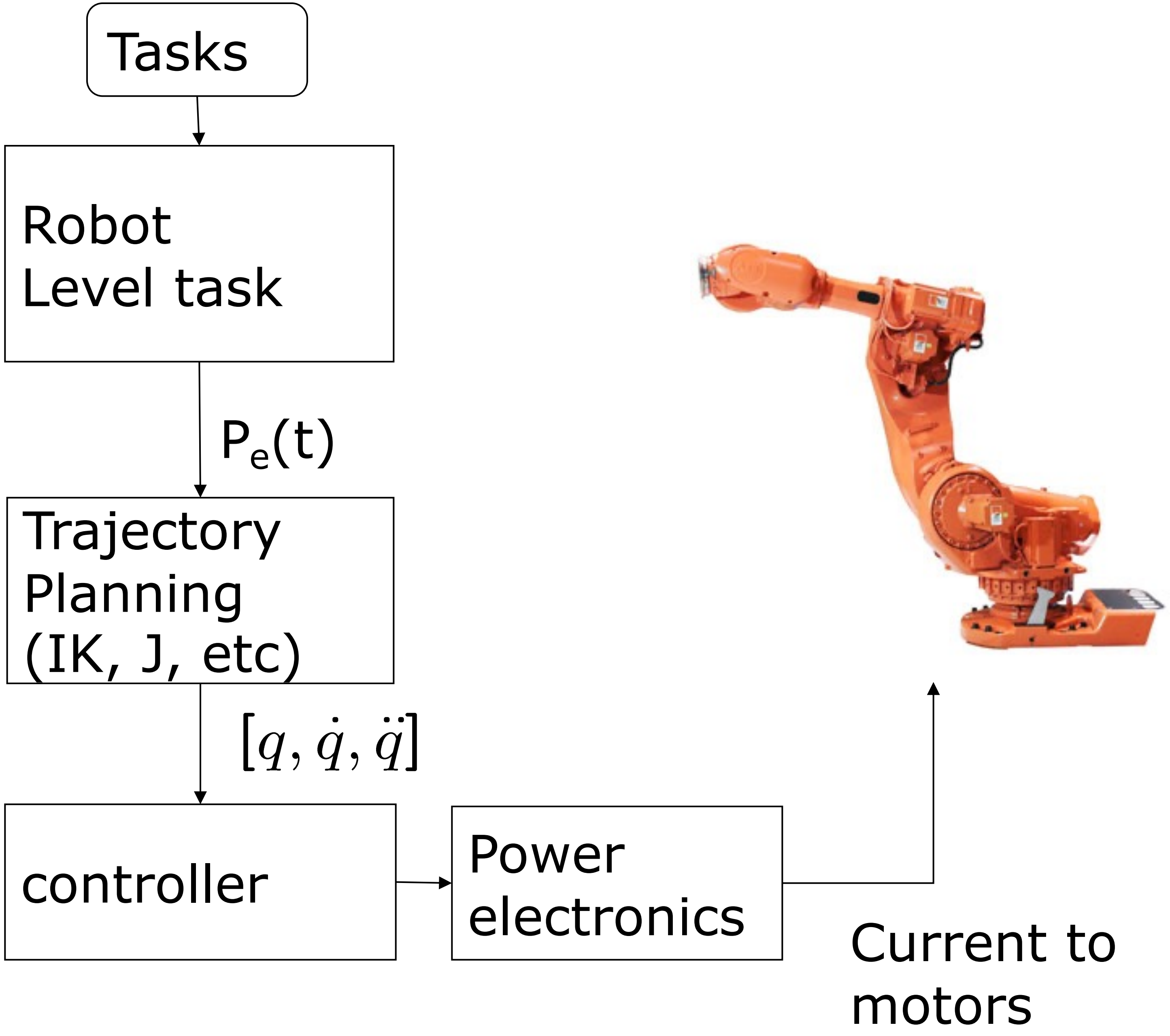
$$\dot{X} = J \dot{q}$$

Rate of change of joint position

Rate of change of Cartesian pose

CONTROL

Control


Tasks
Robot Level task
$P_e(t)$
Trajectory Planning (IK, J, etc)
$[q, \dot{q}, \ddot{q}]$
controller
Power electronics
Current to motors

Actuator Model

- Need to model relationships:
  - between actuator torque and motor angle (q)
    - Second order ode

disturbance

$$J\ddot{q}(t) + B\dot{q}(t) = u(t) - d$$

Rotational inertia of joint, kg m^2

control input

Effective damping (friction, back emf), Nm/amp

Independent Joint Control

- Control each joint independently without “communication” between actuators
- Basic Steps:
  - ✓Model actuator
  - ✓Use kinematics to obtain setpoints for each joint (IK)
  - Develop a controller for each joint
    - Error for joint i:

$$e_i = (q_i^* - q_m)$$

$$q_i^* = \text{desired joint position}$$

$$q_m = \text{measured joint position}$$

Proportional control for each joint

- Input proportional to position error:

$$u(t) = K_{PE} e_i(t) = K_{PE}(q_i^*(t) - q_m(t))$$

- Neglect disturbance, set reference position to zero

$$u(t) = K_{PE}(0 - q_m(t))$$

$$J\ddot{q}(t) + B\dot{q}(t) = -K_{PE} q(t)$$

- or

$$J\ddot{q}(t) + B\dot{q}(t) + K_{PE} q(t) = 0$$

Proportional control for each joint

- Second order linear differential equation:

$$J\ddot{q}(t) + B\dot{q}(t) + K_{PE}q(t) = 0$$

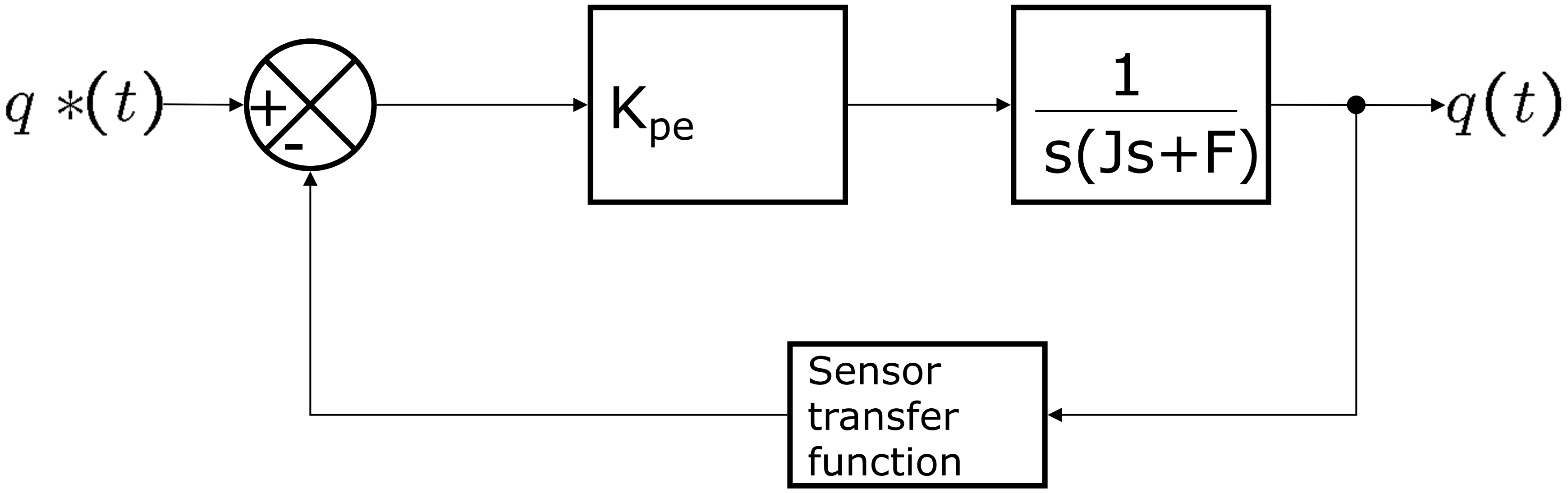
- has general form solution:

$$q(t) = \exp\left(\frac{-Bt}{2J}\right)\left[C_1 \exp(\frac{\omega t}{2}) + C_2 \exp(\frac{-\omega t}{2})\right]$$

- where

$$\omega = \sqrt{\left(\frac{B^2}{J^2}\right) - \left(\frac{4K_{PE}}{J}\right)}$$

Block Diagram of PE controller



Three solutions

$$q(t) = \exp\left(\frac{-Bt}{2J}\right)\left[C_1 \exp(\frac{\omega t}{2}) + C_2 \exp(\frac{-\omega t}{2})\right]$$

$$\omega = \sqrt{\left(\frac{B^2}{J^2}\right) - \left(\frac{4K_{PE}}{J}\right)}$$

Three solutions

$$q(t) = \exp\left(\frac{-Bt}{2J}\right)\left[C_1 \exp(\frac{\omega t}{2}) + C_2 \exp(\frac{-\omega t}{2})\right]$$

- Over-damped ($\omega^2 > 0$)

$$\frac{B^2}{4K_{PE}} > J$$

- Critically damped ($\omega^2 = 0$)

$$\exp(\frac{\omega t}{2}) = \exp(\frac{-\omega t}{2}) = 1$$

$$q(t) = C_{12} \exp\left(\frac{-Bt}{2J}\right)$$

$$\omega^2 = \left(\frac{B^2}{J^2}\right) - \left(\frac{4K_{PE}}{J}\right)$$

Three solutions

$$q(t) = \exp\left(\frac{-Bt}{2J}\right)\left[C_1 \exp(\frac{\omega t}{2}) + C_2 \exp(\frac{-\omega t}{2})\right]$$

$$\frac{B^2}{4K_{PE}} < J$$

- Under-damped ($\omega^2 < 0$)
  - ω has complex roots

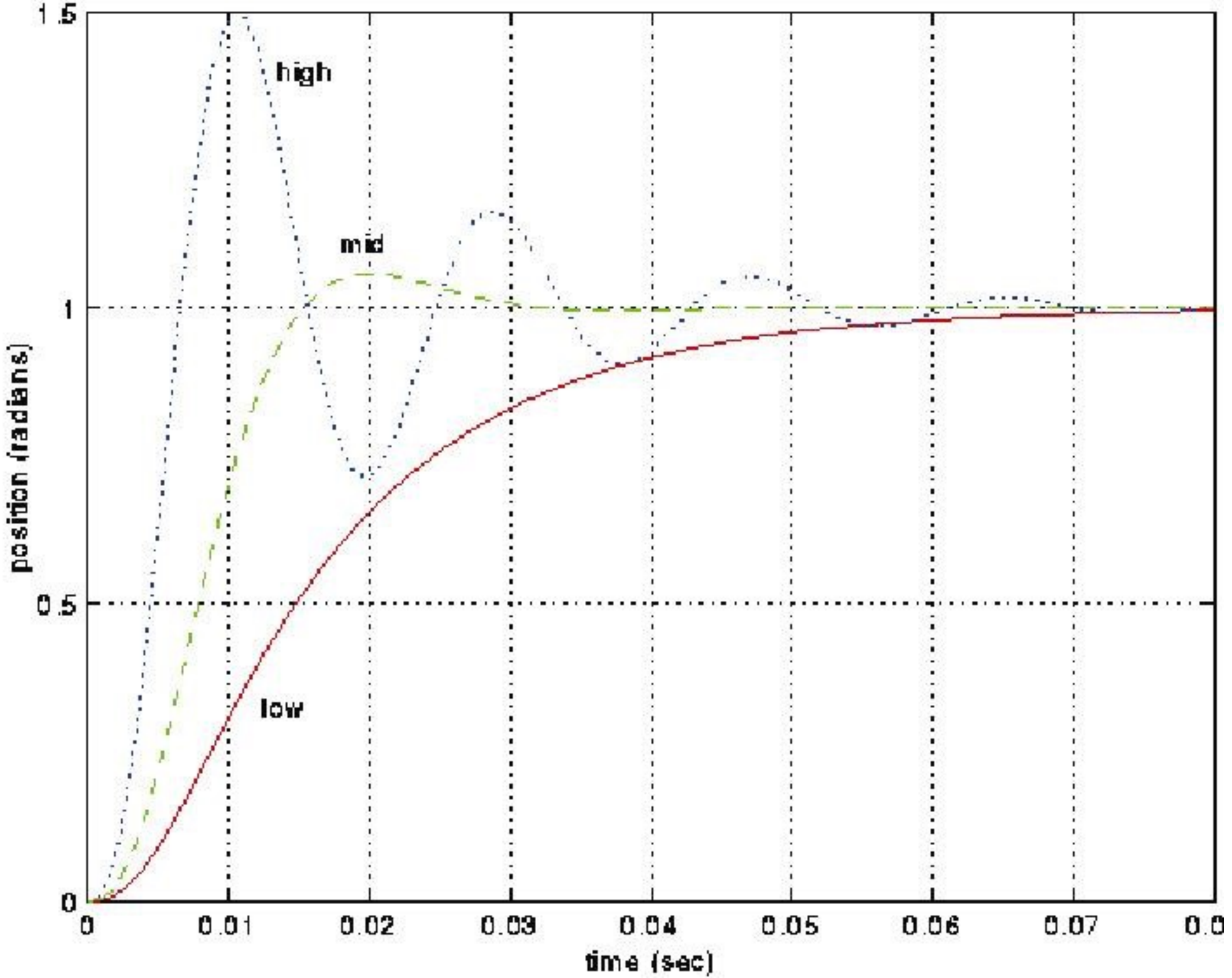
$$q(t) = e^{\frac{-Bt}{2J}}\left[(C_1 + C_2)\cos(\frac{\omega t}{2}) + j(C_1 - C_2)\sin(\frac{\omega t}{2})\right]$$

  - Oscillates with frequency

$$f = \frac{2\pi}{\omega}\ Hz$$

If B is small and K_{PE} is large: unstable!

Example Step Responses (goal: 1 radian)



PI, PID controllers

- PE controllers can lead to
  - Steady state error
  - Unstable behavior
- Add Integral Term:

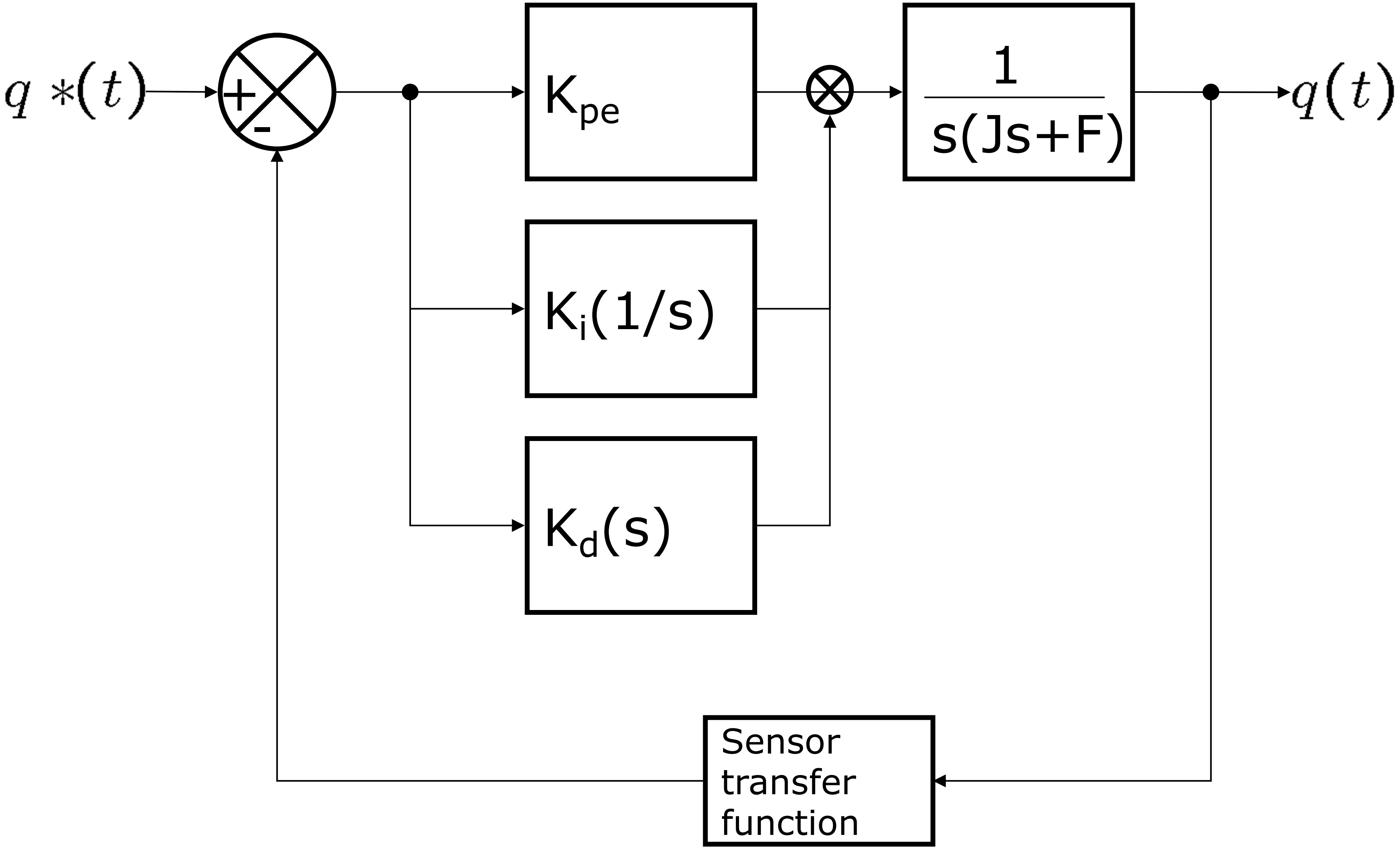
$$\tau_c = K_{pe}(q_r - q_m) + K_I \int_0^t q_r - q_m(u)du$$

….but now we can have overshoot

- Add derivative term (PID Controller)

$$\tau_c = K_{pe}(q_r - q_m) + K_I \int_0^t q_r - q_m(u)du - K_d\dot{q}(t)$$

Block Diagram of PE controller



Set Gains for PID Controller

- wlog set $q^* = 0$ (we al ready have $\dot{q}^* = 0$)

$$J\ddot{q}(t) + B\dot{q}(t) = -K_{pe}q(t) - K_I \int_0^t q(u)du - K_d\dot{q}(t)$$

- Convert to third order equation

$$J \dddot{q} + (B + K_d)\ddot{q} + K_{pe}\dot{q} + K_I q = 0$$

- Solution will be of the form

$$q(t) = f(J, B, K_{pe}, K_I, K_D, \omega, t)$$

- where

$$\omega = \sqrt{g(J, B, K_{pe}, K_I, K_D)}$$

Set Gains for PID Controller

- Critically damped when $\omega = 0$ or

$$g(J, B, K_{pe}, K_I, K_D) = 0$$

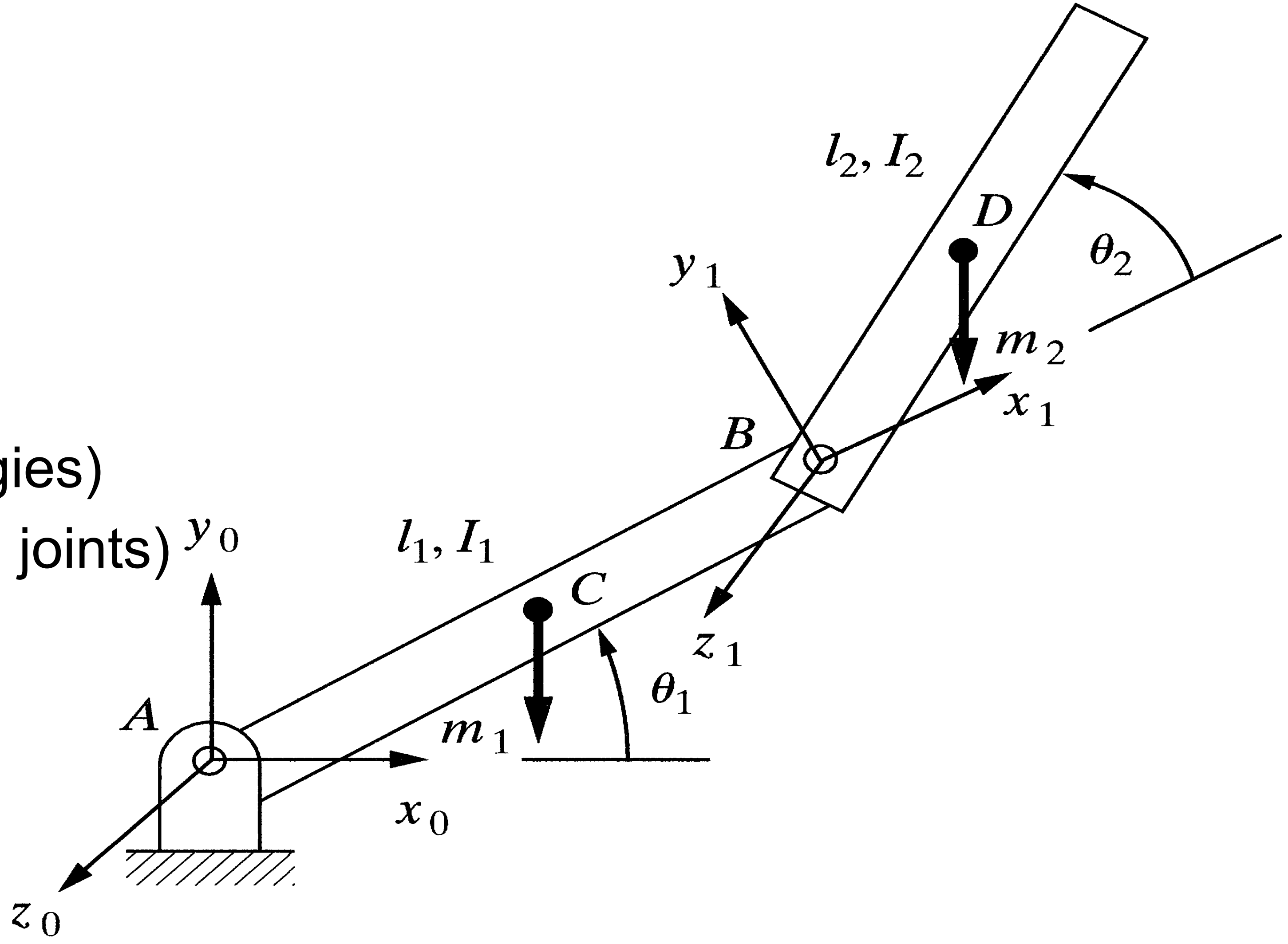
  - An equation in 3 unknowns
  - Need two more constraints:
    - Minimum energy
    - Minimum error
    - Minimum jerk
  - And we need the solution to double minimization
    - Beyond the scope of this class – topic of optimal control class

Problems with Independent Joint Control

- Synchronization ?
  - If one joint does not follow the trajectory, where is the end-effector???
- Ignores dynamic effects
  - Links are connected
  - Motion of links affects other links
  - Could be in-efficient use of energy

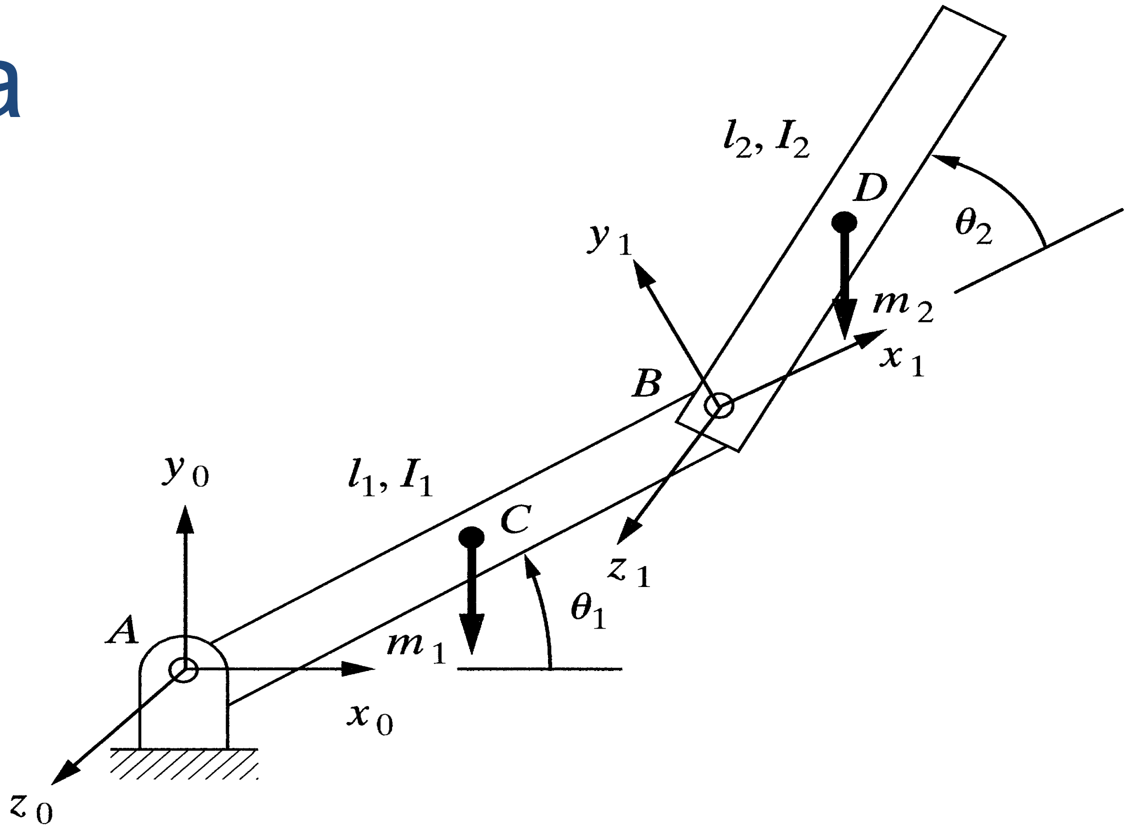
Dynamics

- For all links, consider:
  - Linear Momentum
  - Angular Momentum
  - Gravity Force (potential energies)
  - Force/ Torque of Actuators (in joints)



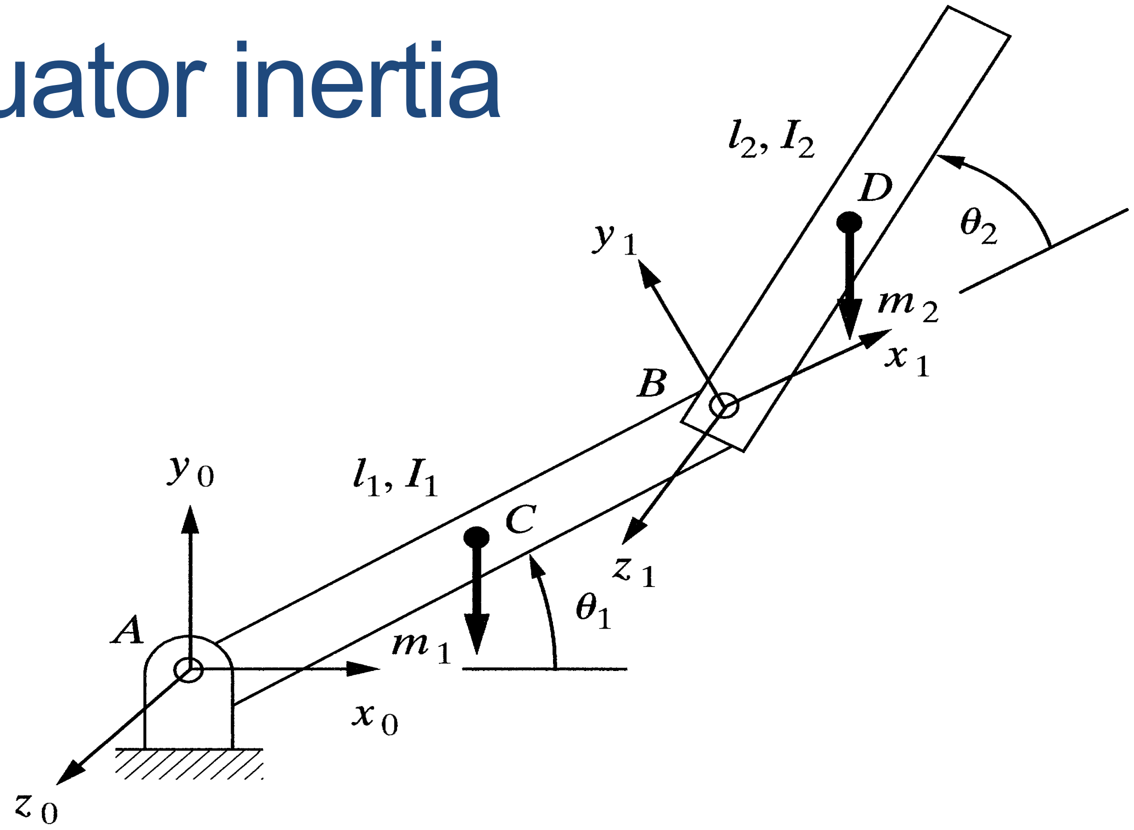
Effective Moments of Inertia

- To Simplify the equation of motion, equations can be rewritten in symbolic form.



$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} D_{ii} & D_{ij} \\ D_{ji} & D_{jj} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_i \\ \ddot{\theta}_j \end{bmatrix} + \begin{bmatrix} D_{iii} & D_{ijj} \\ D_{jii} & D_{jjj} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix} = \begin{bmatrix} D_{iii} & D_{ijj} \\ D_{jii} & D_{jjj} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \dot{\theta}_2 \\ \dot{\theta}_2 \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} D_i \\ D_j \end{bmatrix}$$

Final equations without actuator inertia



$$T_1 = \left(\frac{1}{3}m_1l^2 + \frac{4}{3}m_2l^2 + m_2l^2C_2\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2C_2\right)\ddot{\theta}_2$$

$$+\left(\frac{1}{2}m_2l^2S_2\right)\dot{\theta}_2^2 + \left(m_2l^2S_2\right)\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}m_1glC_1 + \frac{1}{2}m_2glC_{12} + m_2glC_1 + I_{1(act)}\ddot{\theta}_1$$

$$T_2 = \left(\frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2C_2\right)\ddot{\theta}_1 + \left(\frac{1}{3}m_2l^2\right)\ddot{\theta}_2 + \left(\frac{1}{2}m_2l^2S_2\right) + \frac{1}{2}m_2glC_{12} + I_{2(act)}\ddot{\theta}_1$$

Force Control

- Force Control:
  - The robot encounters with unknown surfaces and manages to handle the task by adjusting the uniform depth while getting the reactive force.

- Examples:
  - Tapping a Hole - move the joints and rotate them at particular rates to create the desired forces and moments at the hand frame.
  - Peg Insertion – avoid the jamming while guiding the peg into the hole and inserting it to the desired depth.

OPTIMAL CONTROL

Model Predictive Control

Linear Quadratic Regulator (LQR): For the continuous-time linear system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\delta \qquad \text{with} \qquad \mathbf{x}(0) = \mathbf{x}_{\text{init}}$$

and quadratic cost functional defined as ($\mathbf{Q}$ is a diagonal weight matrix)

$$J = \frac{1}{2}\int_0^\infty \Delta\mathbf{x}^T(t)\mathbf{Q}\Delta\mathbf{x}(t) + q\delta(t)^2\, dt$$

with $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_{\text{target}}$. The feedback control $\delta(t)$ that minimizes J is given by

$$\delta(t) = -\mathbf{k}^T(t)\Delta\mathbf{x}(t)$$

with $\mathbf{k}(t) = \frac{1}{q}\mathbf{b}^T\mathbf{P}(t)$ and $\mathbf{P}(t)$ the solution to a Ricatti equation (no details here).

Model Predictive Control

Generalizes LQR to:

- **Non-linear** cost function and dynamics (consider straight road leading into turn)
- **Flexible:** allows for receding window & incorporation of constraints
- **Expensive:** non-linear optimization required at every iteration (for global coordinates)

Formally:

$$\underset{\delta_1,\ldots,\delta_T}{\operatorname{argmin}} \sum_{t=1}^{T} C_t(\mathbf{x}_t, \delta_t) \quad \text{(sum of costs)}$$

$$s.t. \quad \mathbf{x}_1 = \mathbf{x}_{\text{init}} \quad \text{(initialization)}$$

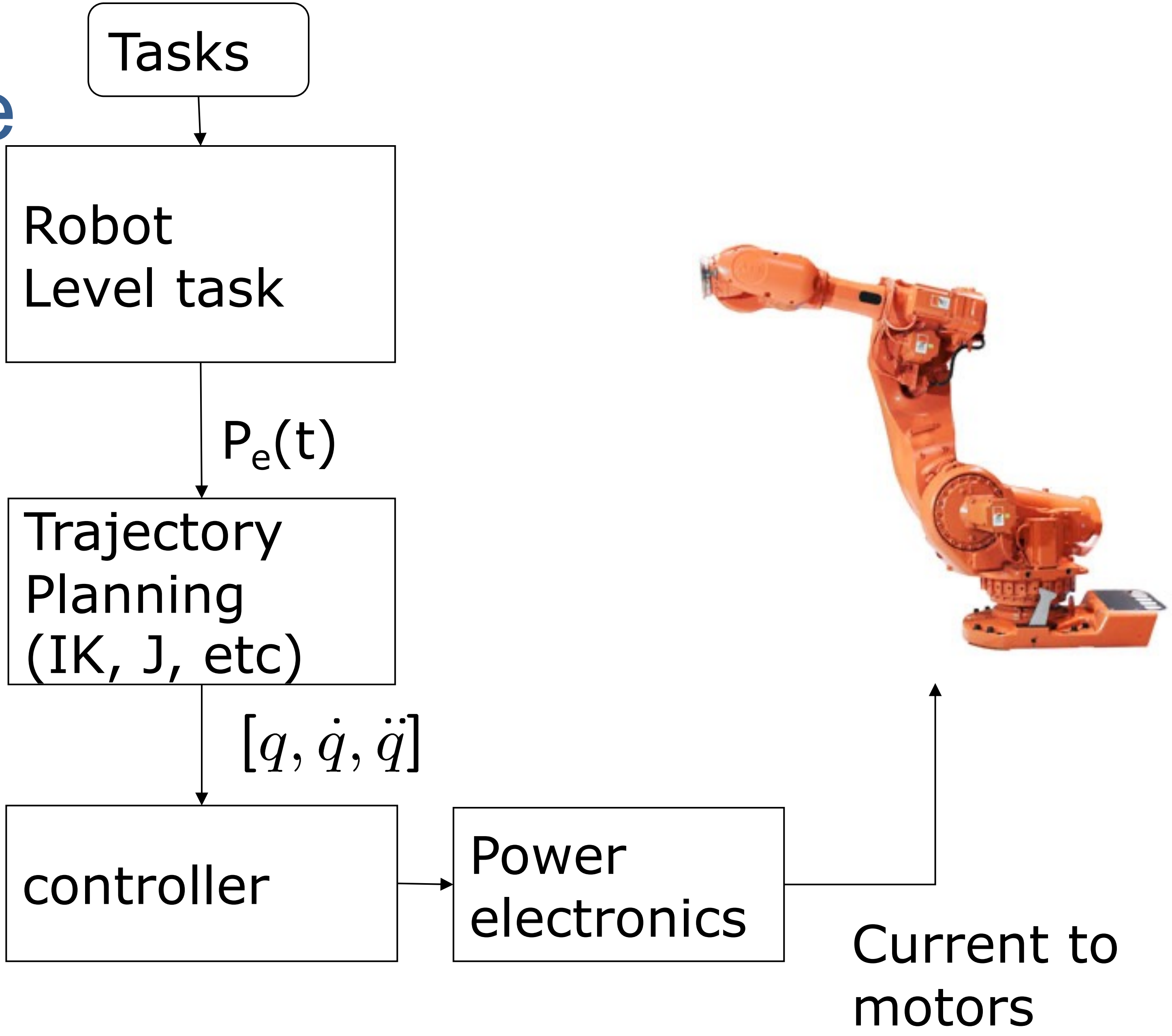
$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \delta_t) \quad \text{(dynamics model)}$$

$$\underline{\delta} \leq \delta_t \leq \bar{\delta} \quad \text{(constraints)}$$

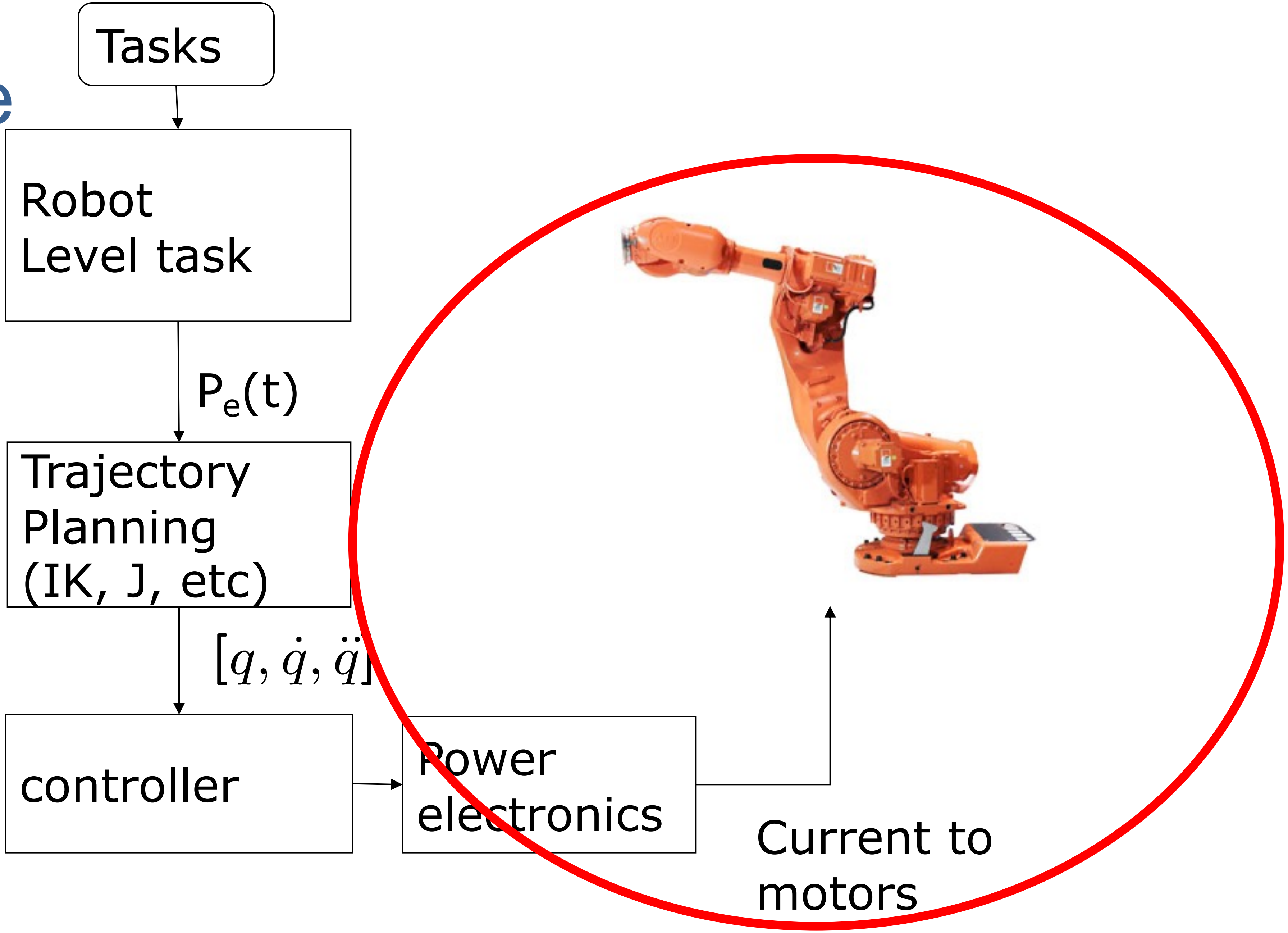
- Unroll dynamic model T times $\Rightarrow$ apply non-linear optimization to find $\delta_1, \ldots \delta_T$

HARDWARE

Hardware


Tasks
Robot Level task
$P_e(t)$
Trajectory Planning (IK, J, etc)
$[q, \dot{q}, \ddot{q}]$
controller
Power electronics
Current to motors

Hardware


Tasks
Robot Level task
$P_e(t)$
Trajectory Planning (IK, J, etc)
$[q, \dot{q}, \ddot{q}]$
controller
Power electronics
Current to motors

Control Hierarchy

- Assume we have a goal trajectory
- Calculate needed joint speeds using Kinematics =>
- Desired joint speeds
  - Typically not just one joint =>
  - Many motor controllers, motors, encoders
- Motor control loop
- Pose control loop

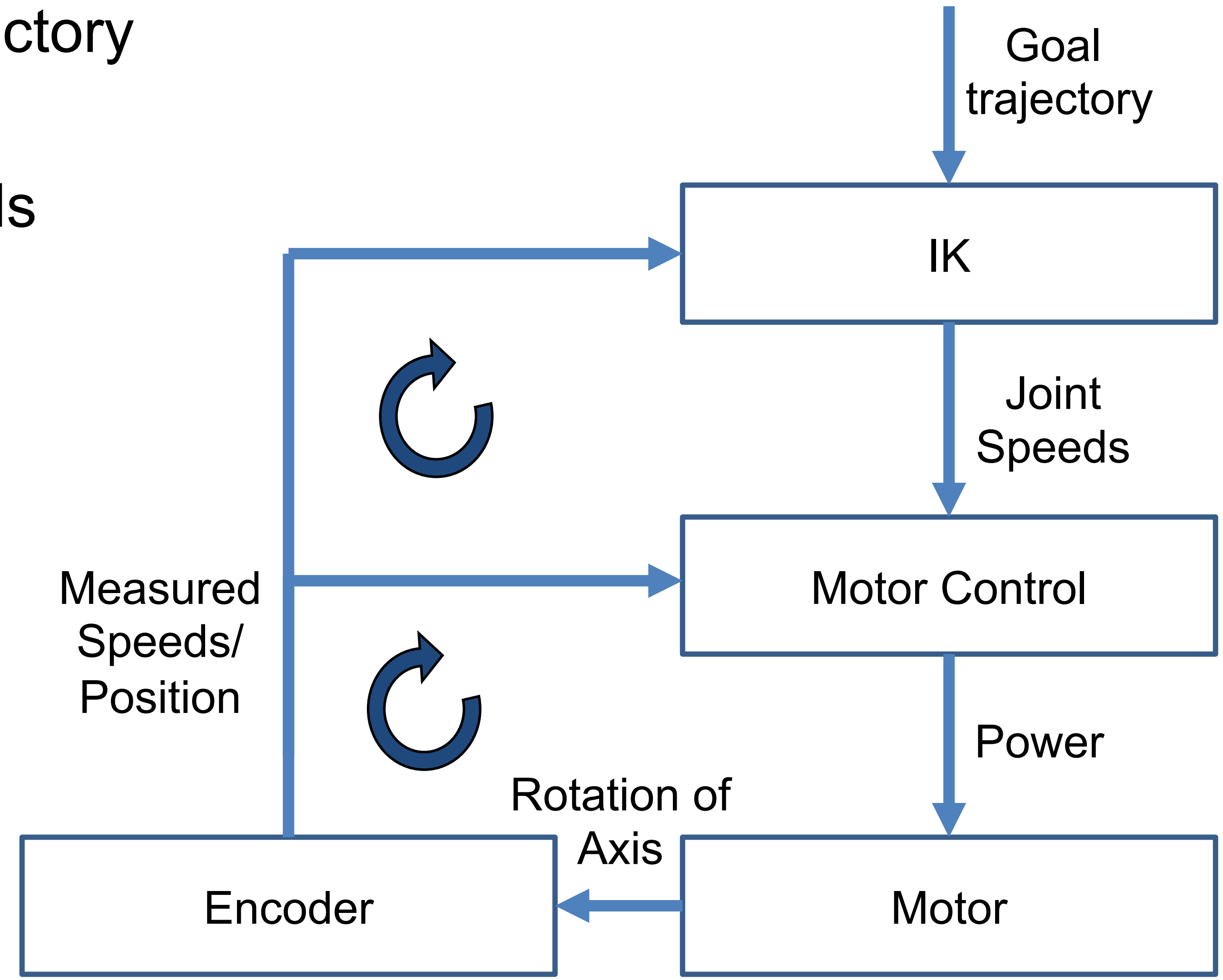
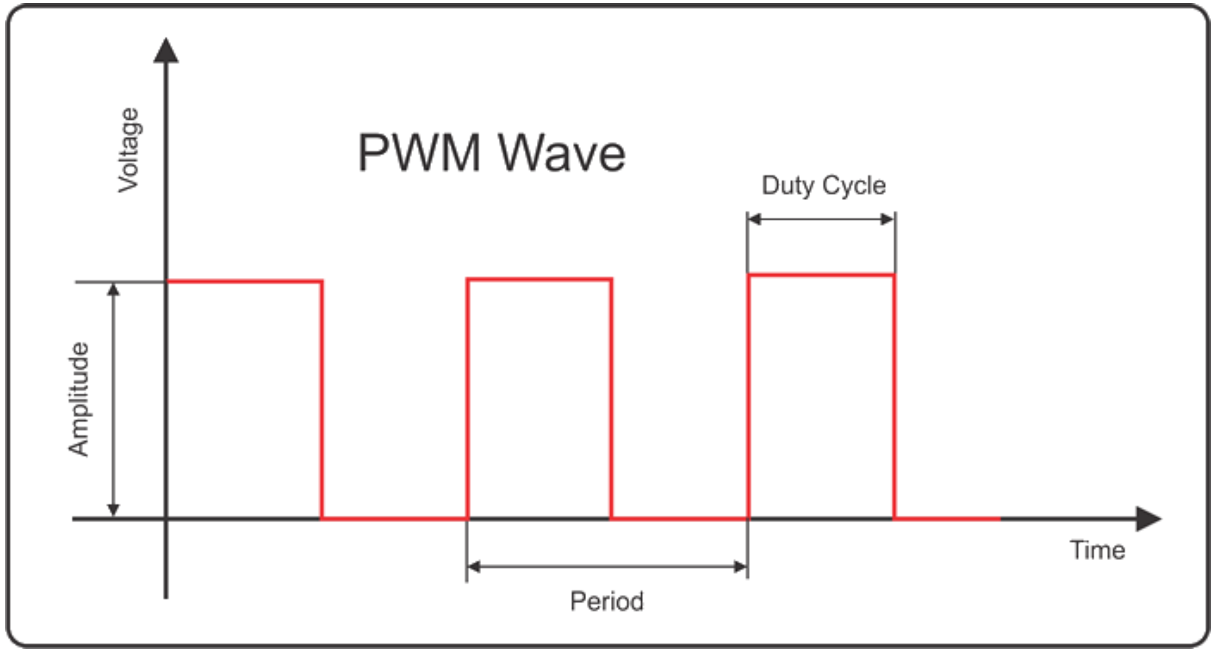
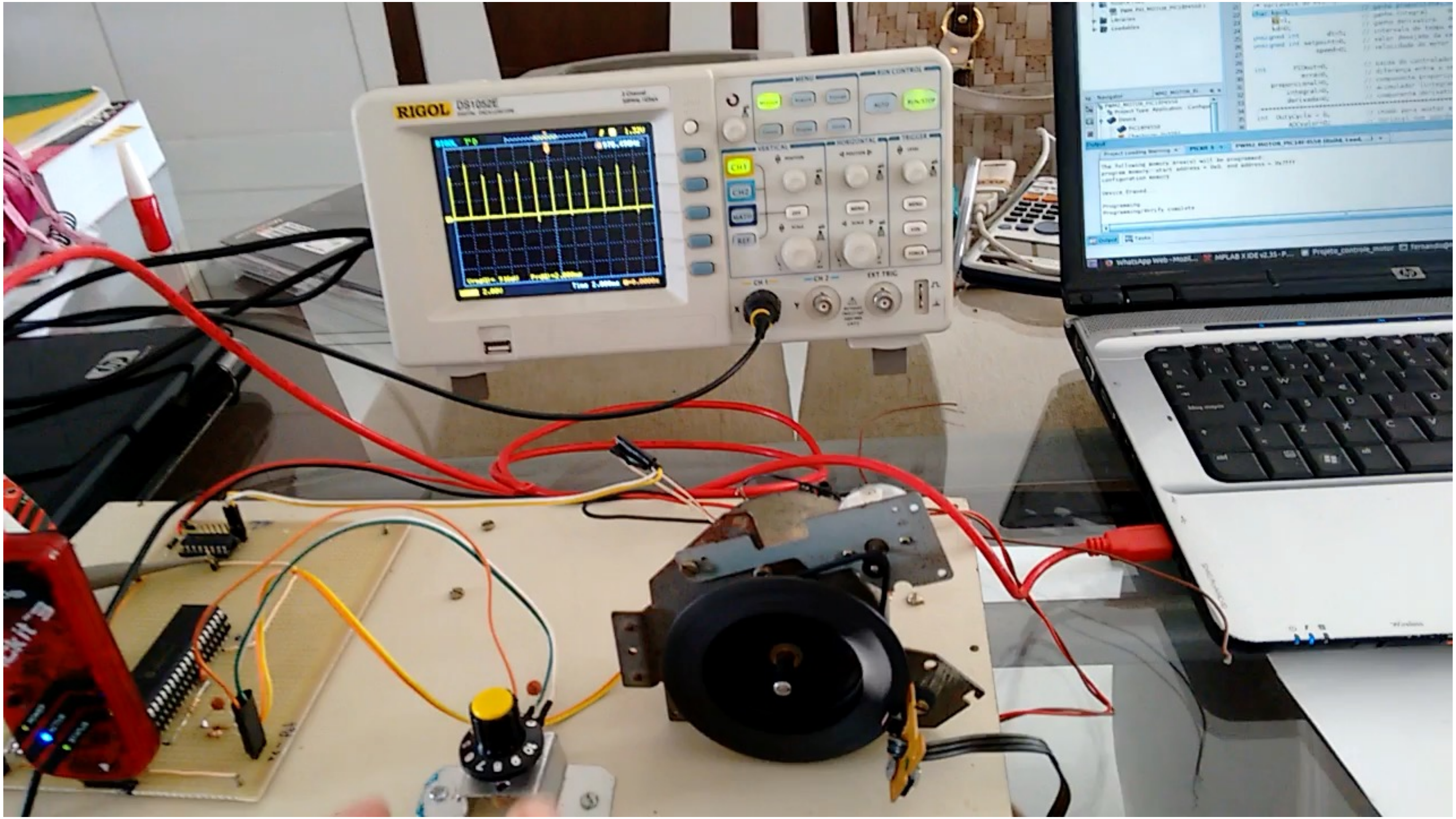


Image: zembedded.com

Pulse Width Modulation



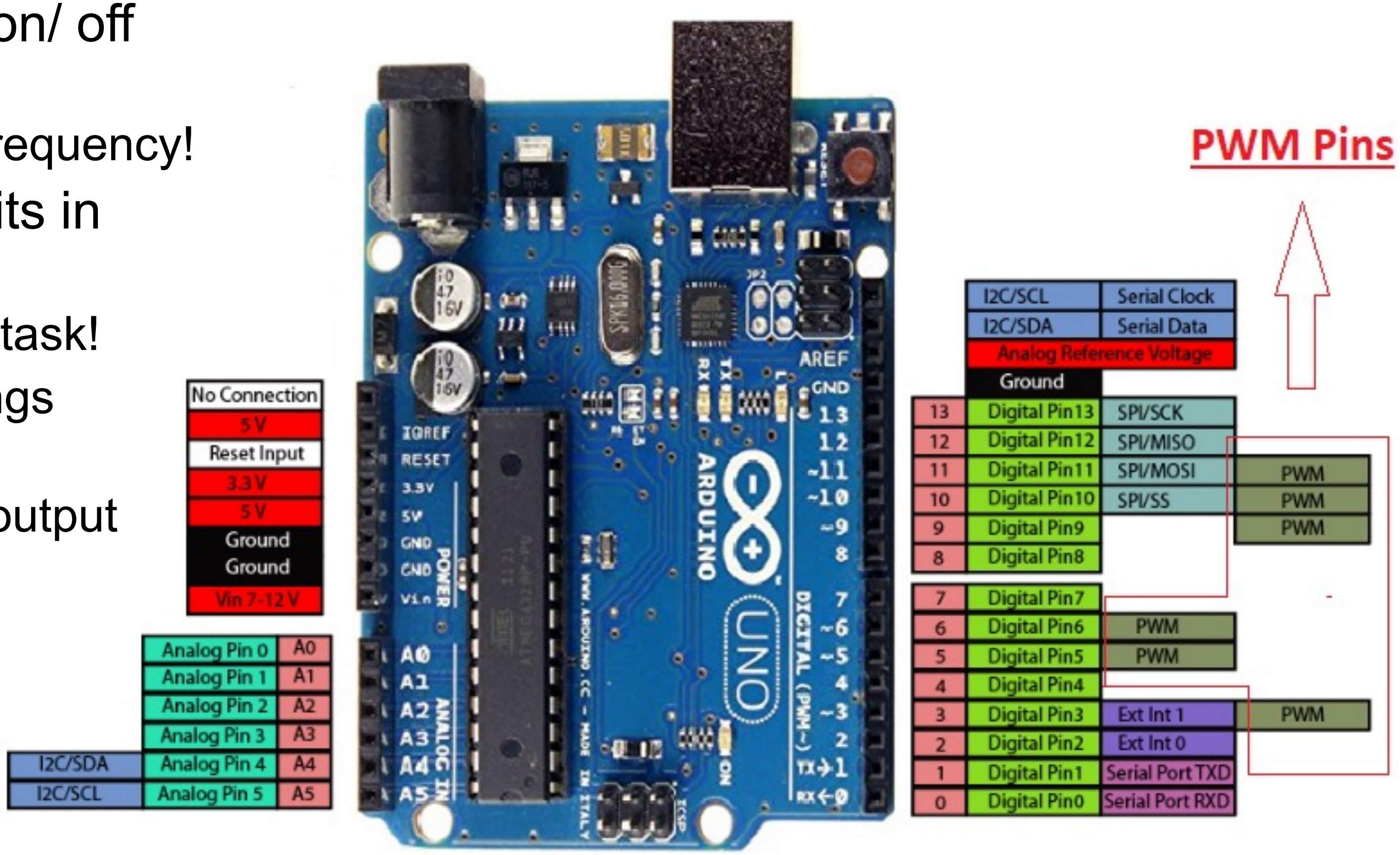
- How can Controller control power?
  - Cannot just tell the motor “use more power”
  - Output of (PID) controller is a signal
  - Typical: Analogue signal
- Pulse Width Modulation (PWM)
  - Signal is either ON or OFF
  - Ratio of time ON vs. time OFF in a given interval: amount of power
  - Frequency in kHz (= period less than 1ms)
  - Very low power loss
- Signal (typica 5V or 3.3V) to Motor Driver
- Used in all kinds of applications:
  - electric stove; audio amplifiers, computer power supply (hundreds of kHz!)



https://www.youtube.com/watch?v=4QzyG5g1blg

PWM Generation

- Motor Control:
  - Frequency in kHz:
  - Smooth motion of motor wanted
  - Use inertia of the motor to smooth the on/ off cycle
    - Still: Sound of motor often from control frequency!
  - High frequency => use dedicated circuits in microcontroller to generate PWM!
    - CPU is not burdened with this mundane task!
    - CPU would suffer from inconsistent timings
      - Interrupts; preemptive computing
    - E.g. Arduino (ATmega48P) has 6 PWM output channels
    - Timer running independently of CPU
    - Comparing to a set register value – if it is up, the output signal is switched

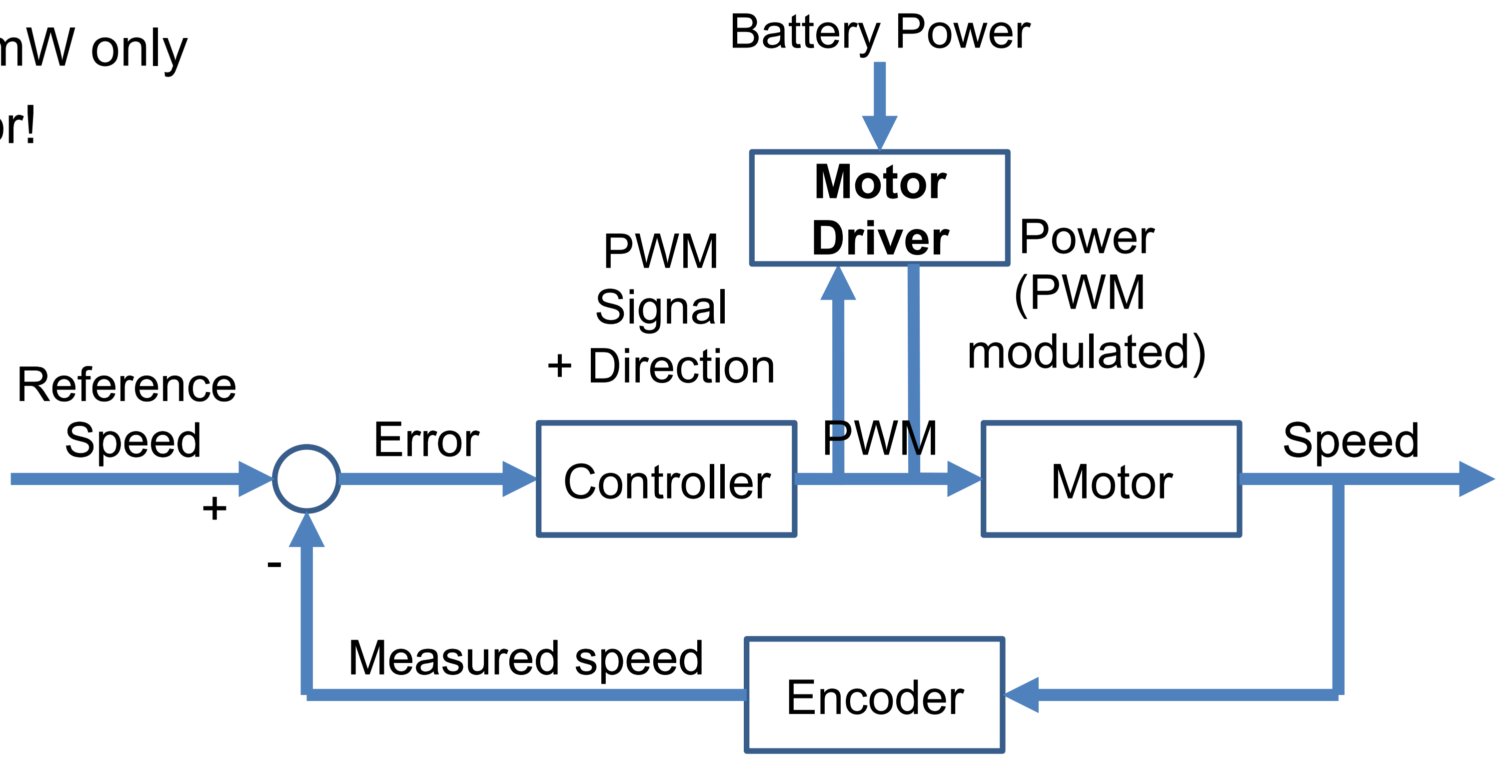


MOTOR DRIVER

Power to the Motor

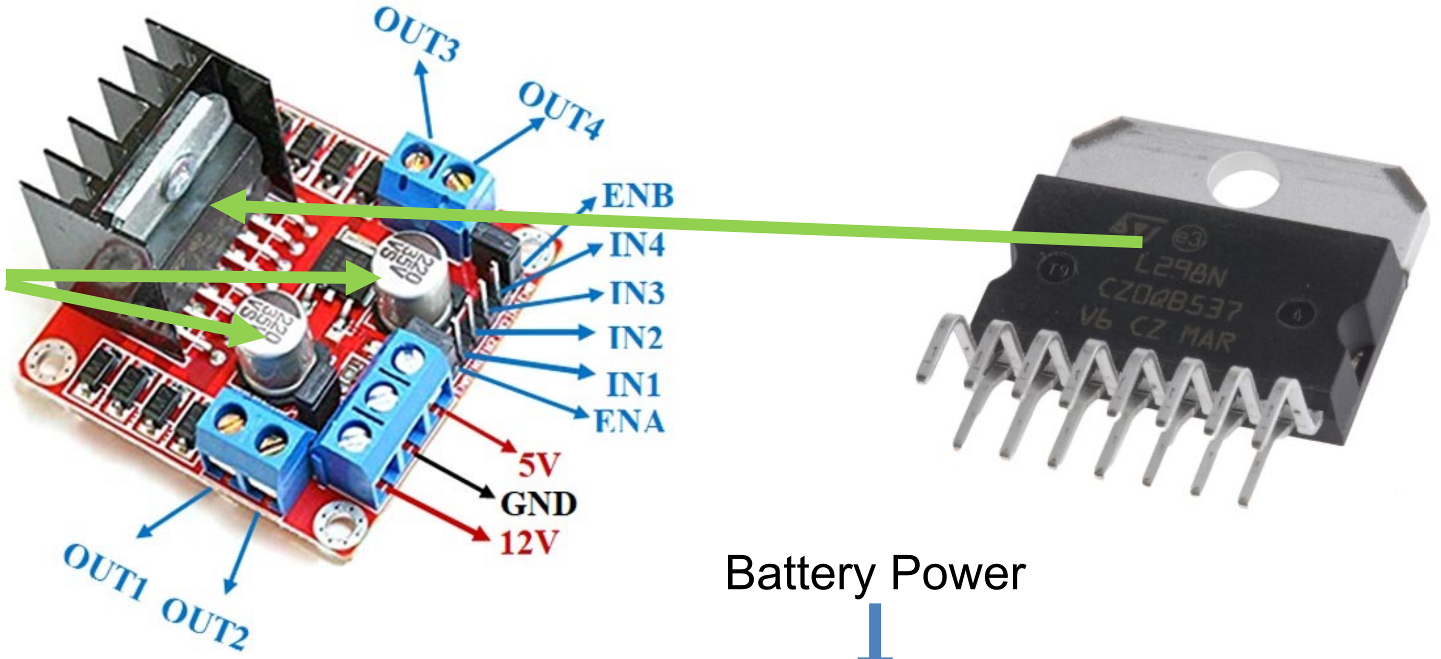
- Direct Current Motor (DC Motor):
  - Two wires for power input
  - Directly connect DC motor to PWM signal?
  - Limited current!
  - E.g.: Arduino: max 30mA => 150mW only
  - Clearpath Jackal: 250W per motor!

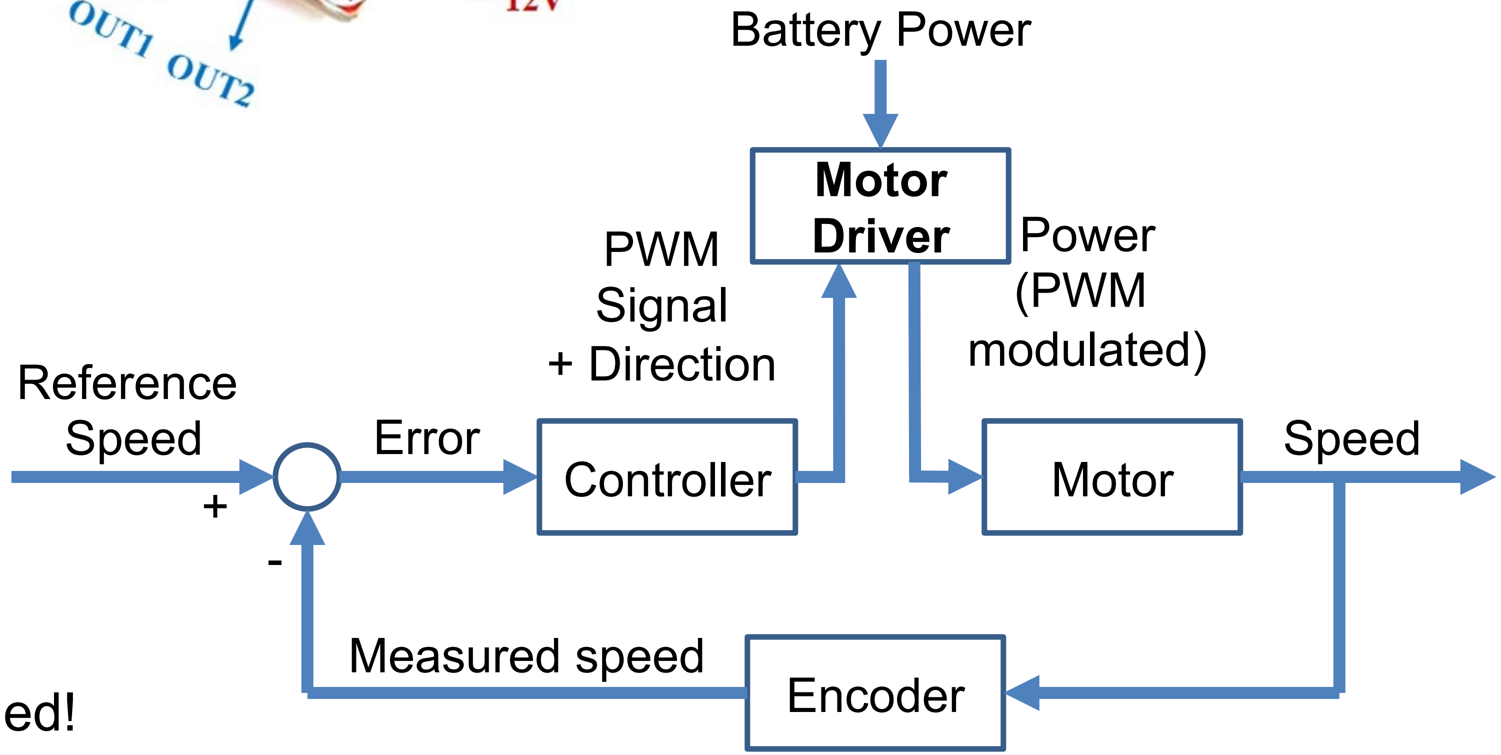
- Need a device to power the motor
- Mobile robots: battery power!



Motor Driver

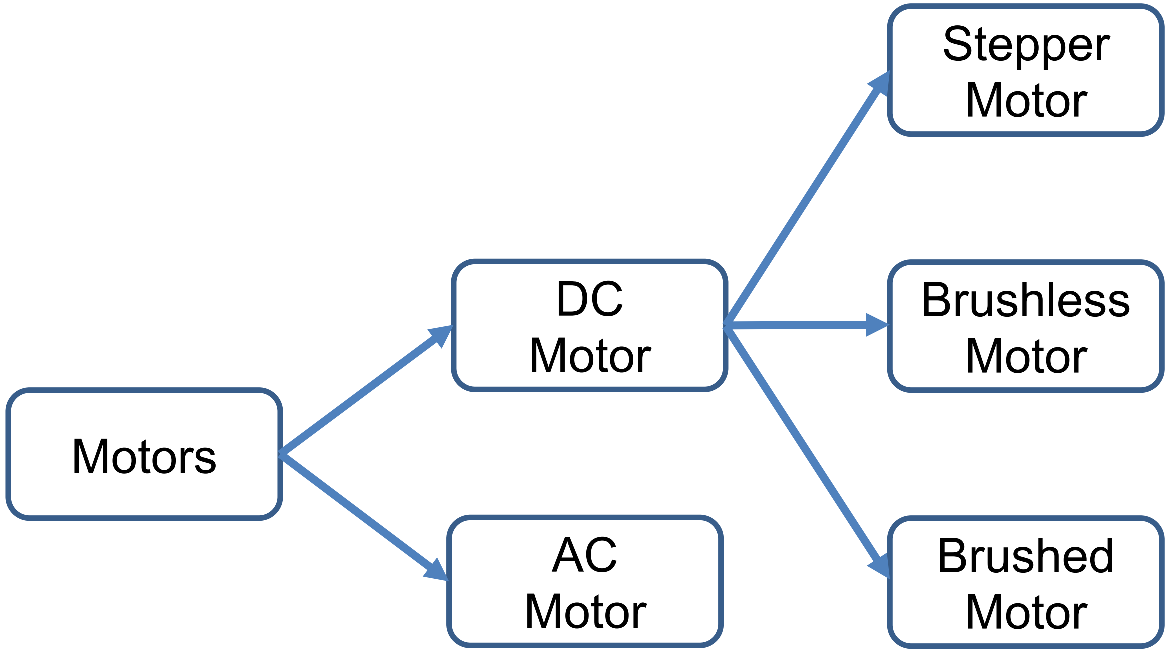
- Motor Driver
  - Input:
    - PWM signal
    - Direction of rotation
    - Battery + & -
    - Optional: Enable =>
      - Emergency Stop
  - Output:
    - Two lines to the DC motor
  - Popular: L298N dual motor driver
    - Up to 48V & 4A
  - High Efficiency (maybe 95%) – but still get‘s hot – cooling needed!





MOTORS

Electrical Motor Types



- DC Motor: Direct Current Motor
- AC Motor: Alternating Current Motor
- Stepper motor:
  - Switching power steps one tooth/ coils forward
  - Open loop control: no encoder needed
  - Low resolution; open loop; torque must be well known
- Brushed motor:
  - Use brushes to power rotating coils => low efficiency and high wear
- Brushless (BL) motor:
  - Electronically control which coil to power => high efficiency low wear
  - Need dedicated controller

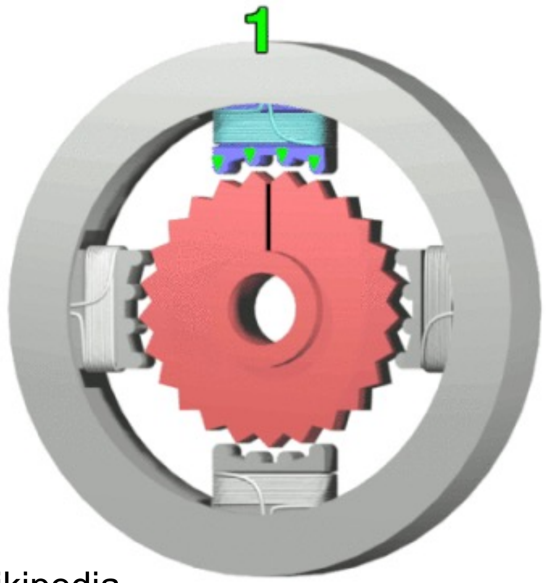
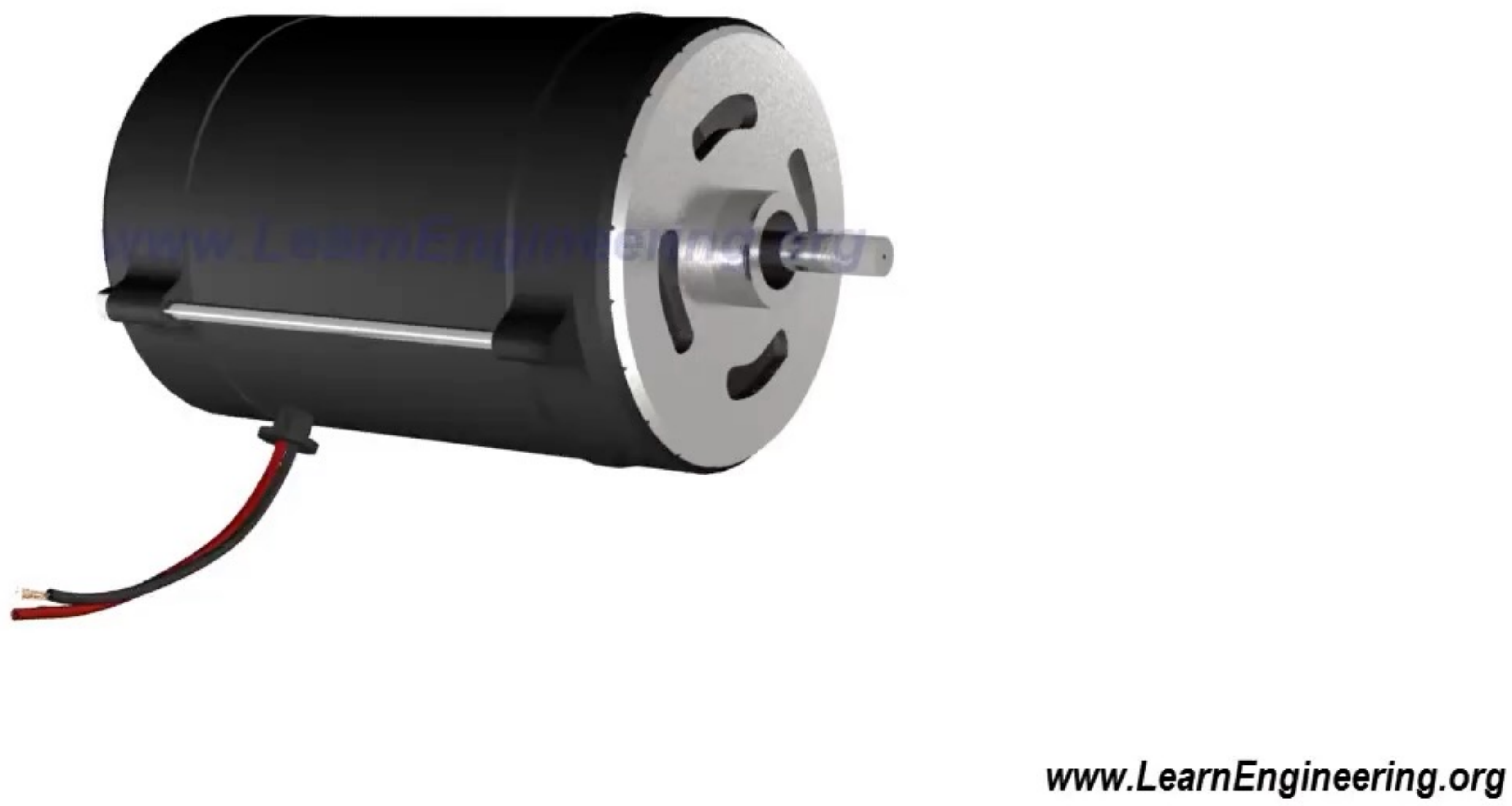


Image: Wikipedia



https://www.youtube.com/watch?v=CWulQ1ZSE3c

https://www.youtube.com/watch?v=bCEiOnuODac

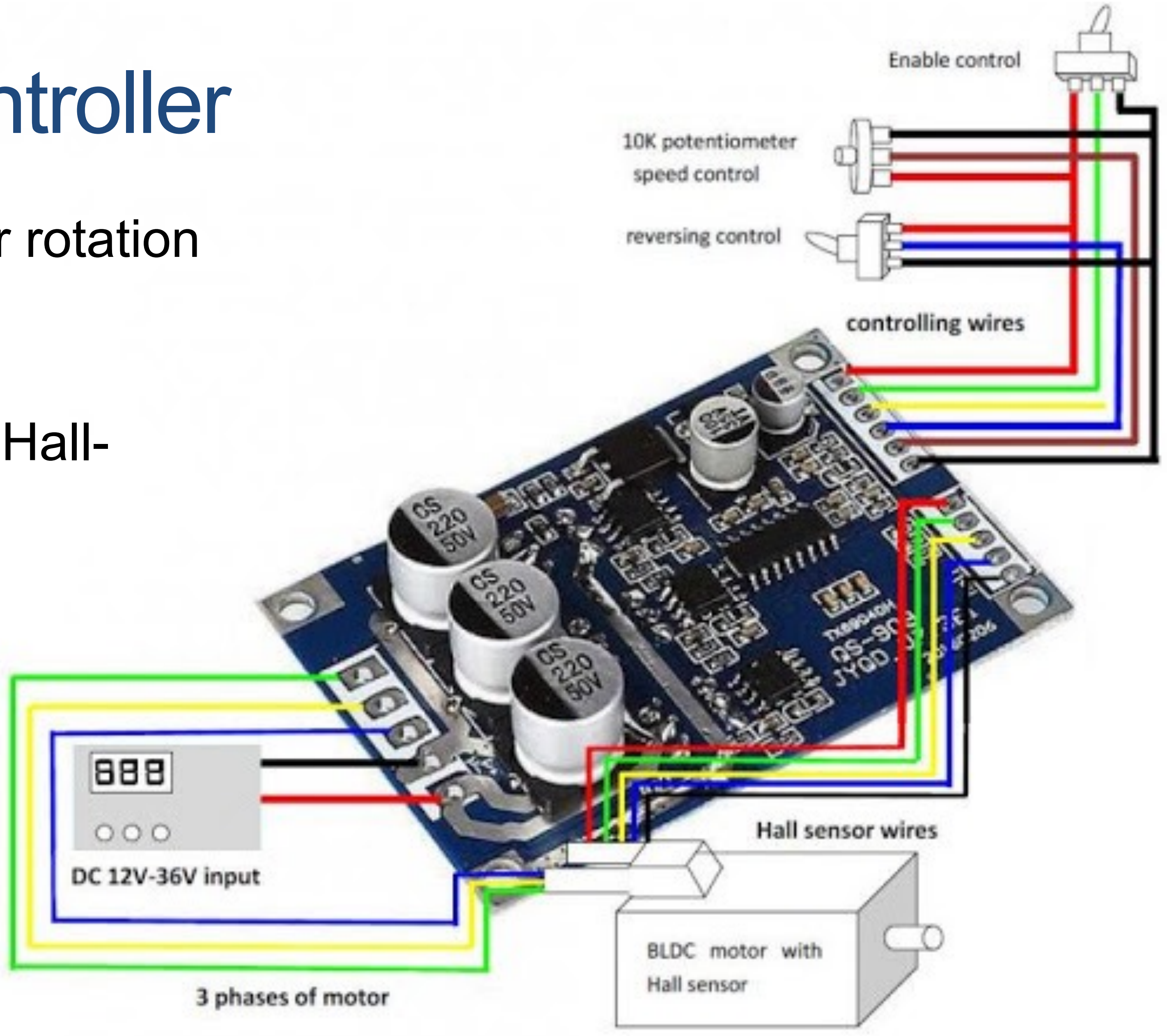
Brushless Motor Controller

- Needs BLDC Controller
  - Does also the job of Motor Driver

- Sensorless BLDC motor:
  - Just apply power to coils in correct order
  - Motor might briefly turn backwards in the beginning
  - Works well for fast spinning motors (e.g. quadcopter)
  - May use the back-EMF (electromotive force) to estimate position

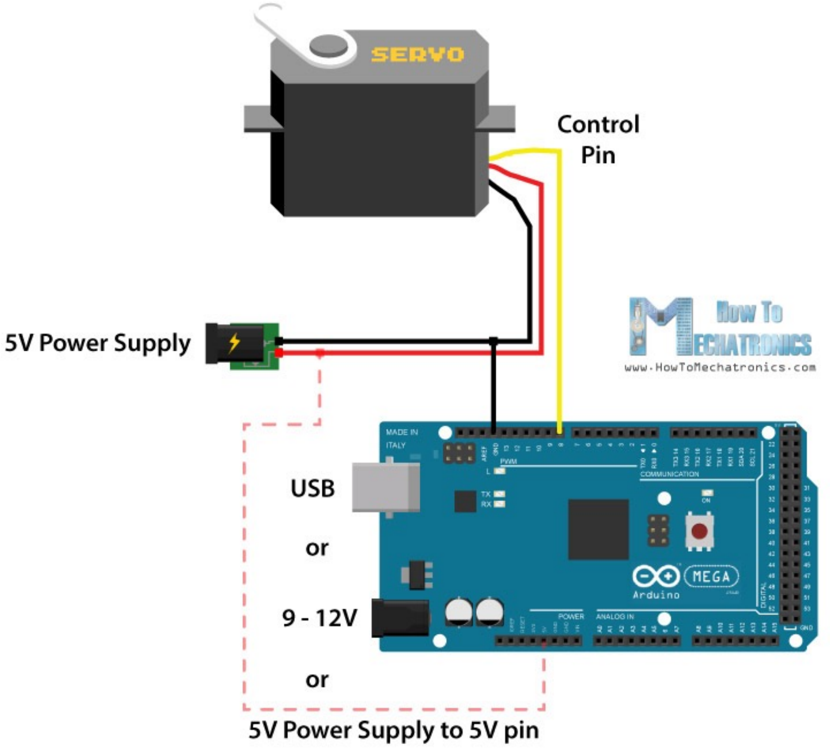
Brushless Motor Controller

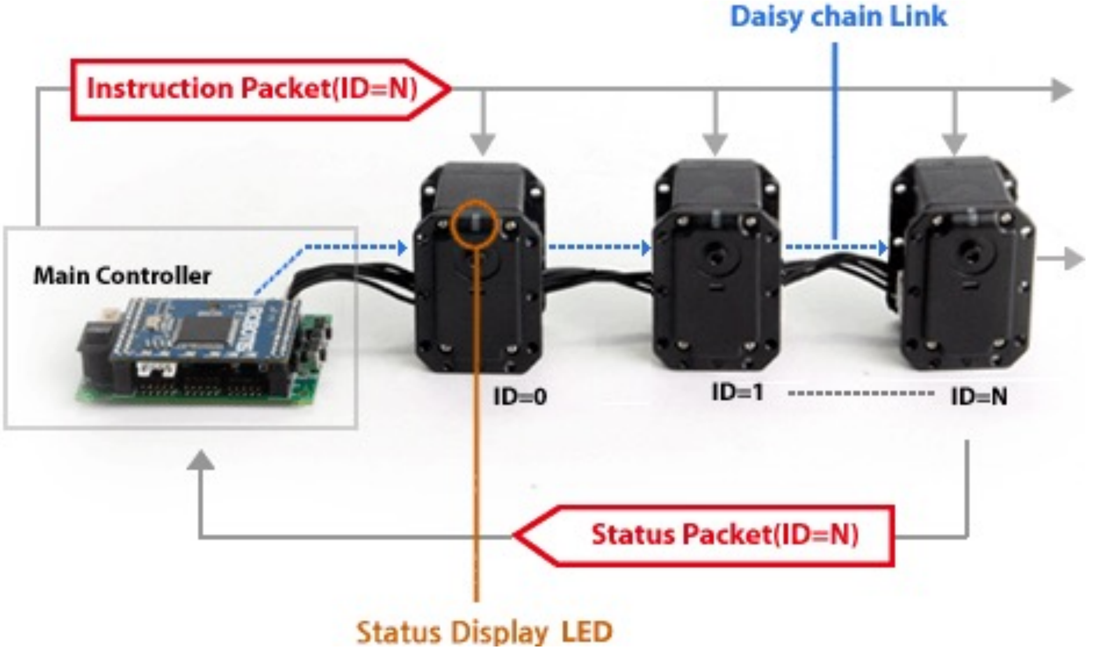
- Hall sensor only 3 positions per rotation
  - Quadrature encoder: up to 4096
- For high torque; low speeds: 3 Hall-effect sensors needed!
- External PID speed control may still be needed!
- Brushless: 20%-30% better efficiency



Servo Motor

- Combines Controller & Motor Driver in the motor
- Input may be analogue (e.g. PWM signal) or digital (e.g. Dynamixel)
- Input specifies a certain (angular) pose for the servo!
  - Servo moves and stays there.
- Continuous Rotation Servos: open loop, speed controlled motors

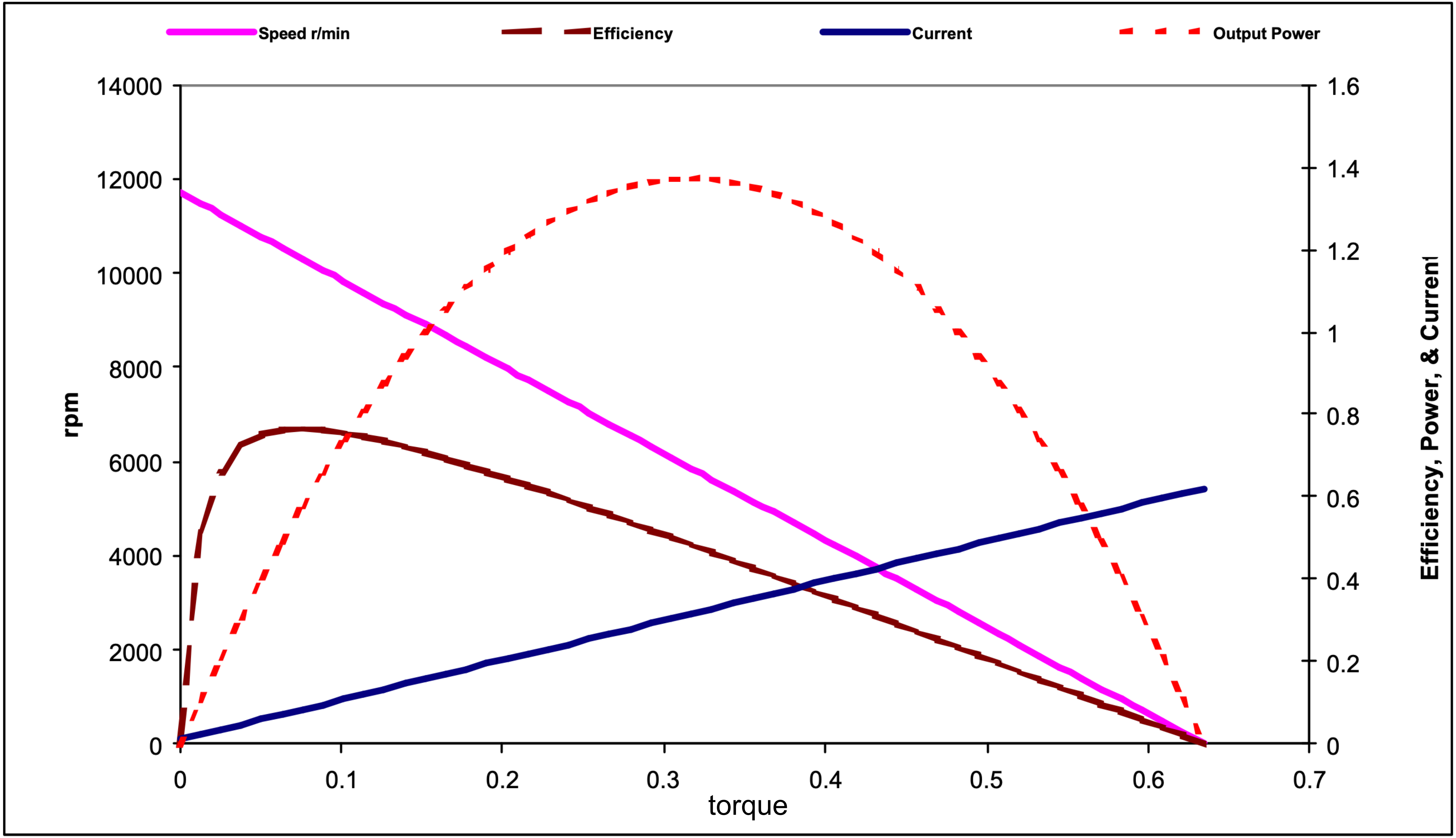




DC Motor Characteristics

- Torque: rotational equivalent to force (aka moment)
  - Measured in Nm (Newton meter)
  - Torque determines the rate of change of angular momentum
- Stall torque:
  - Maximum torque in a DC motor => maximum current => may melt coils
- Maximum energy efficiency:
  - At certain speed/ certain torque
- No-load-speed:
  - Maximum speed; little power consumption

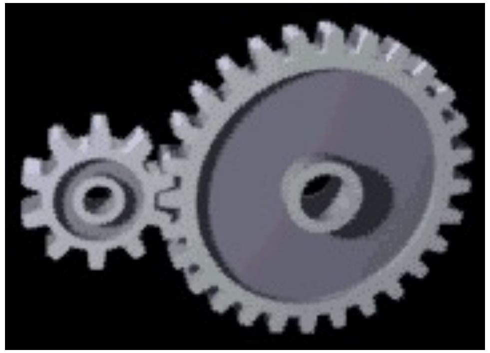
- High-power motors (e.g. humanoid robots) get very hot/ need cooling!


Speed r/min
Efficiency
Current
Output Power
14000
12000
10000
8000
6000
4000
2000
0
rpm
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Efficiency, Power, & Current
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
torque

GEARS

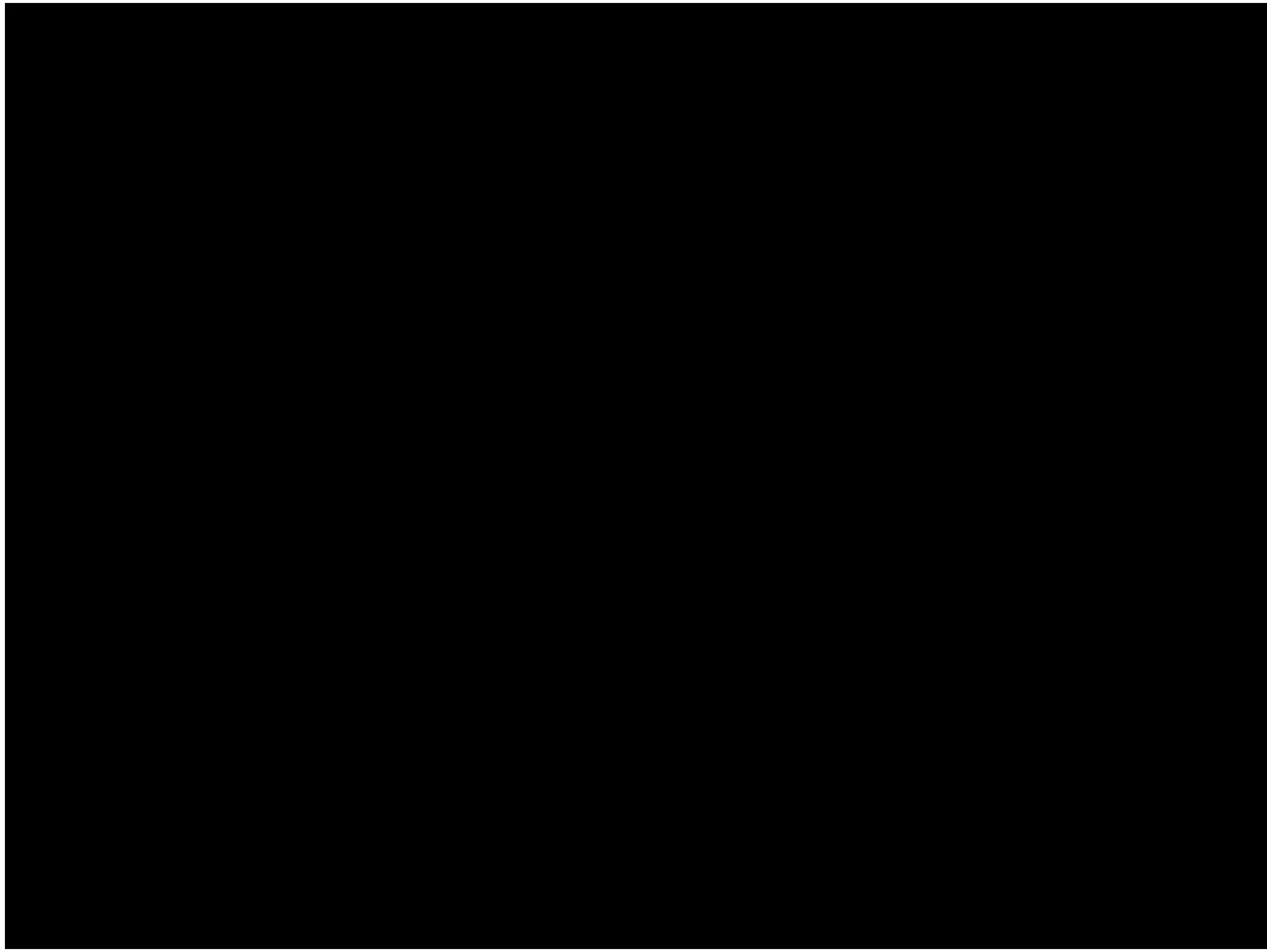
Gears

- Trade speed for torque
- See previous characteristic of DC motor: efficiency highest at high speeds
- Robotics: needs HIGH torque:
  - Inertia of mobile robot (high mass!)
  - Driving uphill
  - Robot arm: lift mass (object and robot arm) at long distances (lever!) – gravity!
- Most important property: Number of teeth => Gear Ratio $= \frac{DrivenGearTeeth}{DriveGearTeeth}$
- Torque = Motor Torque * Gear Ratio
- Speed = Motor Speed / Gear Ratio
- Teeth have same size =>
  gear diameter proportional to Number of teeth…



Gears

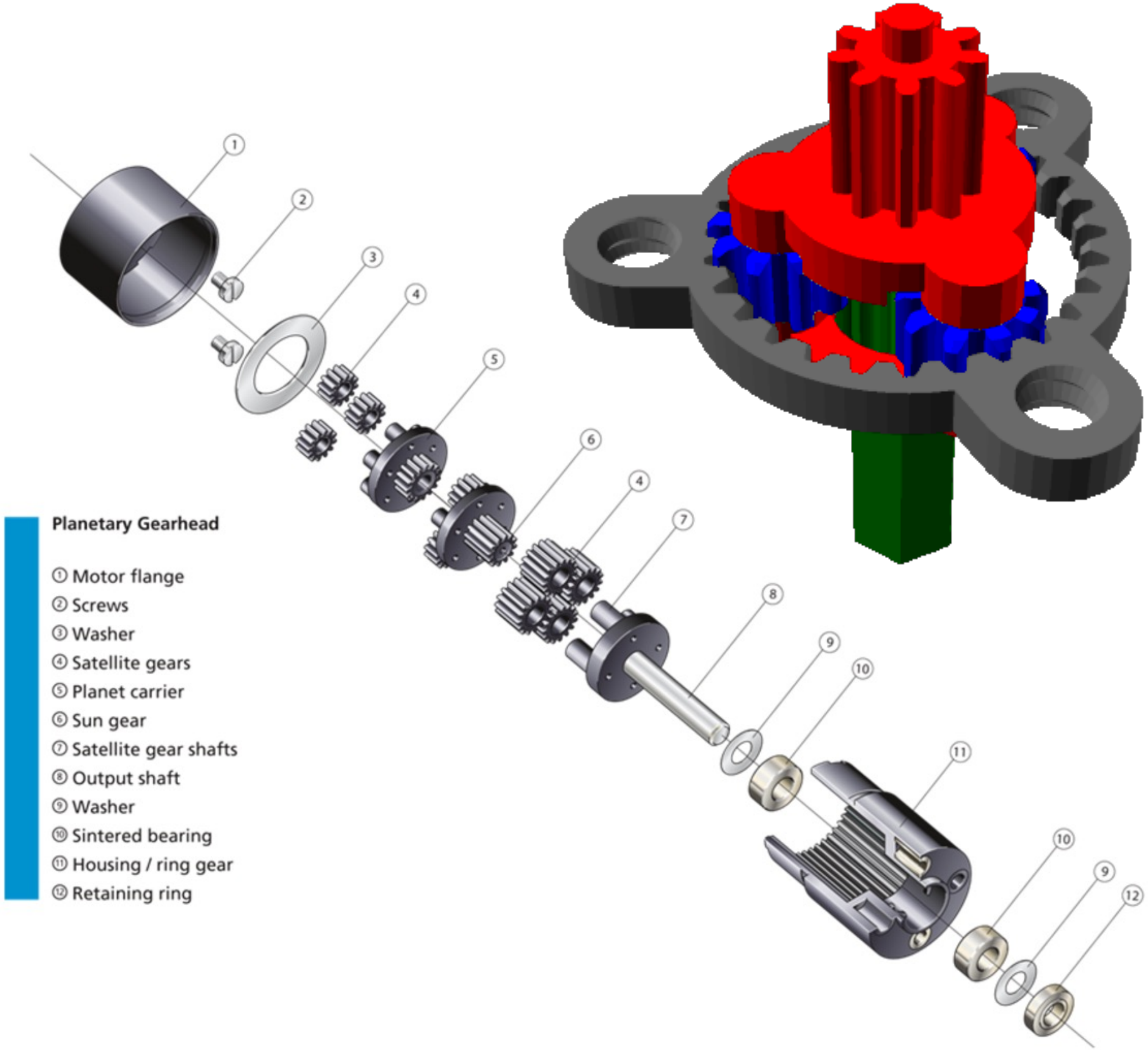
- Must be well designed to provide constant force transmission
  - Low wear/ low noise
- Back drivable: Can the wheel move the motor?
- Spur Gear reverses rotation direction!
- Backlash: when reversing direction: short moment of no force transmission => error in position estimate of wheel!



https://www.youtube.com/watch?v=8s4zm_ajxAA

Planetary Gear

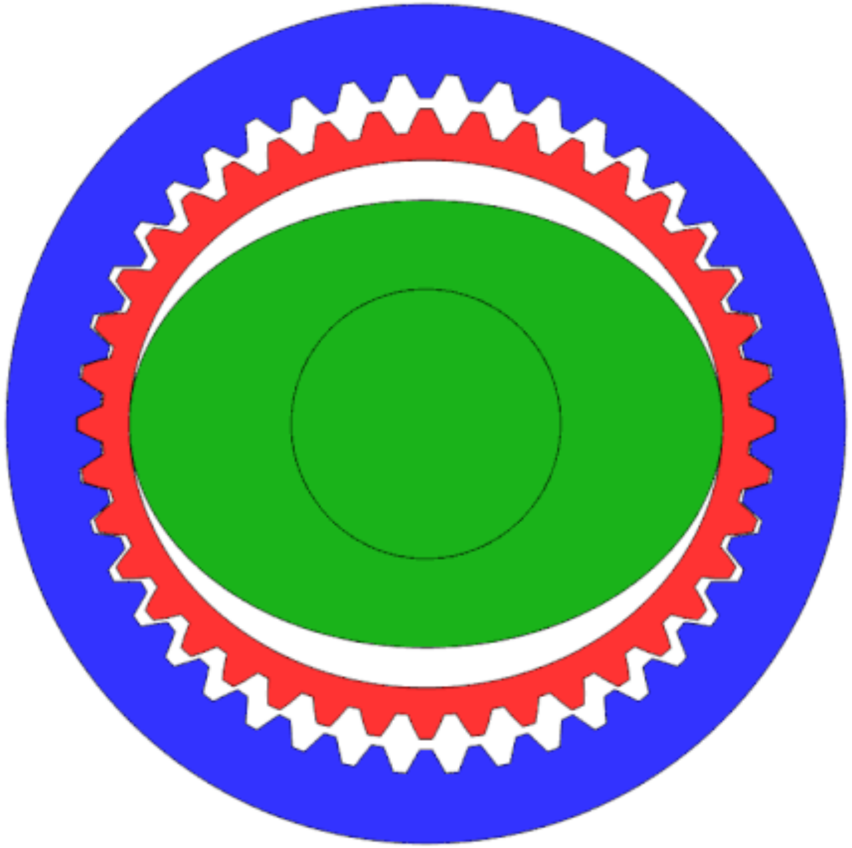
- Aka epicyclic gear train
- Quite common!
- Ratios: 3:1 … 1526:1
- Typical setup:
  - Sun (green) to motor
  - Carrier (red) output
  - Planets (blue): support
  - Ring (black): constraints the planets
  - => Ratio = $1:(1 + N_{Ring}/N_{Sun})$



Harmonic Drive

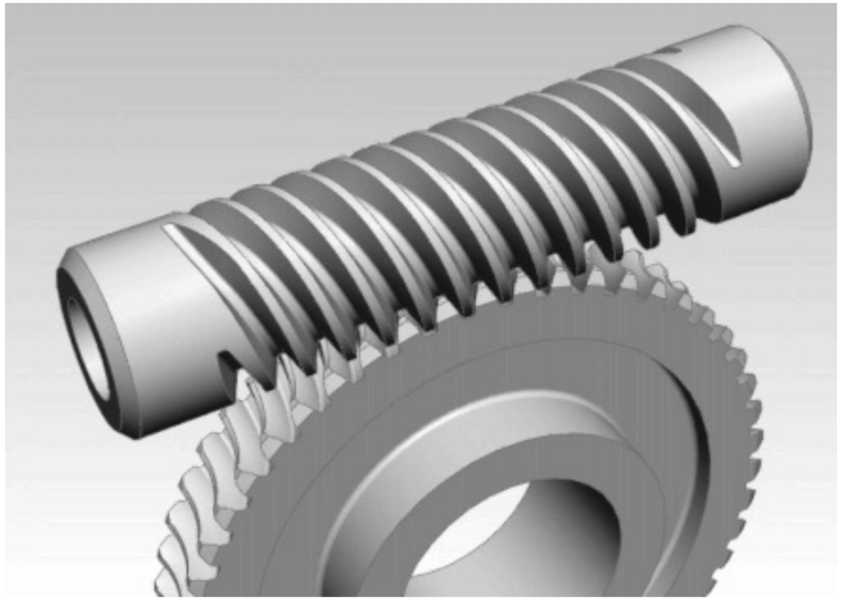
- High reduction in small volume (30:1 to 320:1)
- No backlash
- Light weight

- Used in robotics, e.g. robotic arms (e.g. our Schunk arm!)



$$\text{reduction ratio} = \frac{\text{flex spline teeth} - \text{circular spline teeth}}{\text{flex spline teeth}}$$

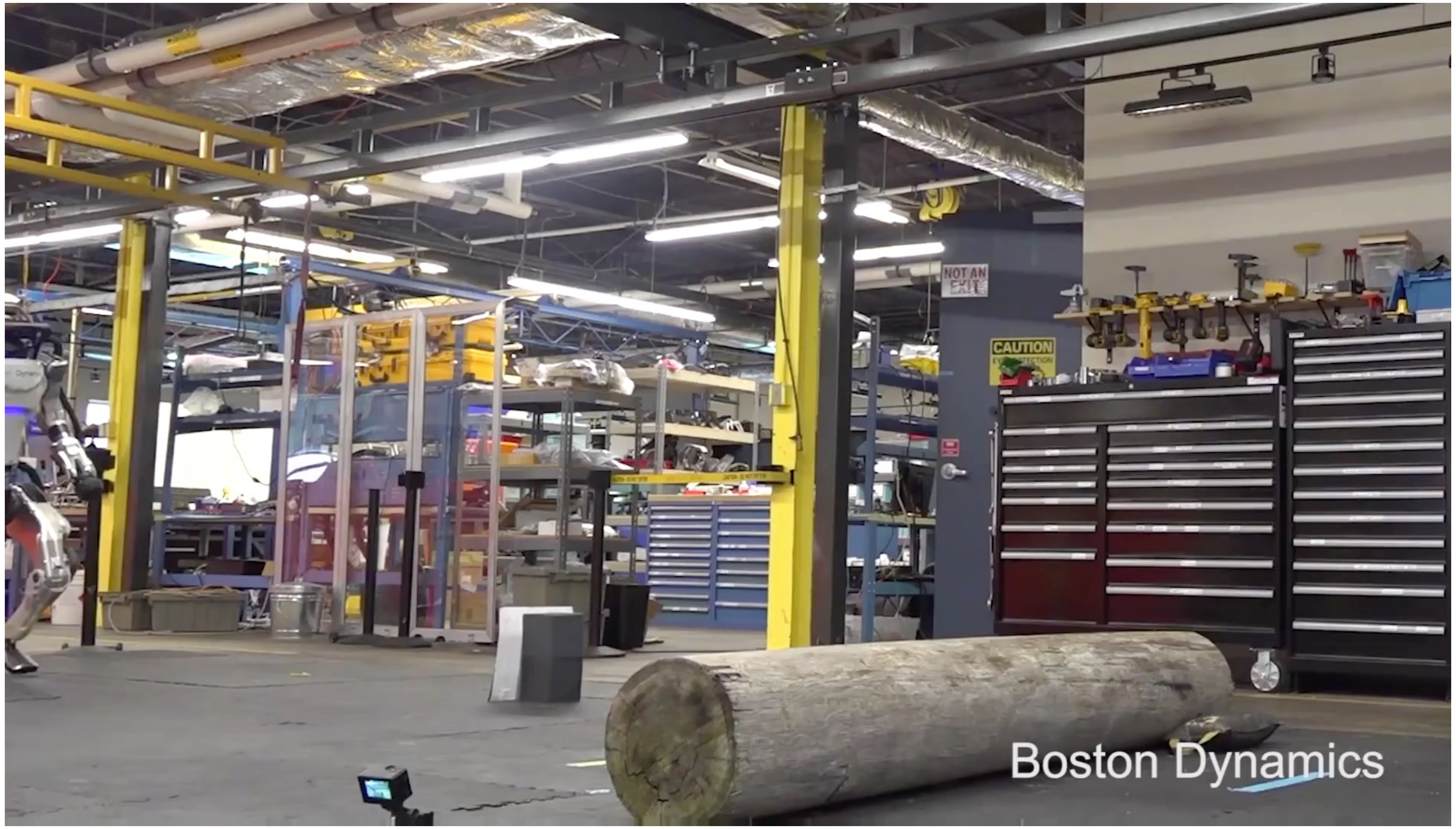
More Gears



- Rack and pinion
  - linear drive
- Worm drive
  - Very high torque
  - Ratio: N_{Wheel} : 1
  - Locking (not back-drivable) gear)
- Bevel gear
  - Mainly to change direction

ALTERNATIVES

Hydraulics



- 28 Hydraulic actuated joints
- Why?
  - Compact actuators with high torque – do not get hot!
  - Low mass
  - One central, highly efficient motor to pressurize the hydraulic fluid

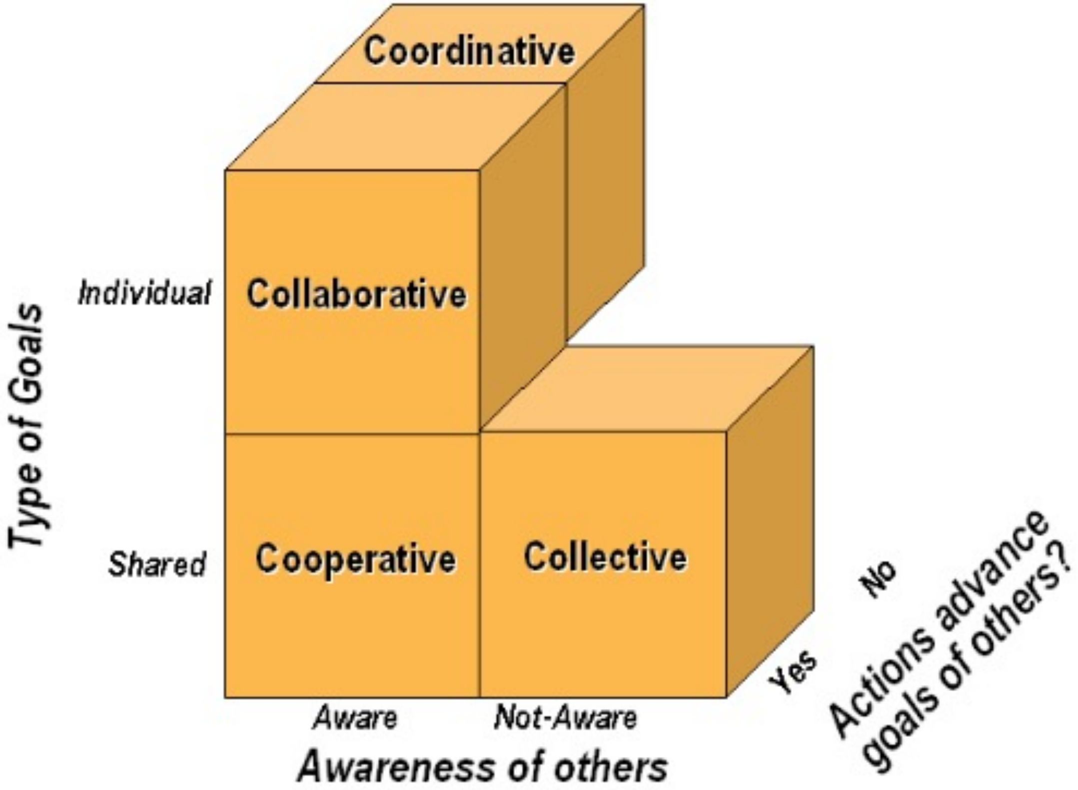
- Actuation controlled via controlling valves

Synthetic Muscles

- Electroactive polymer: Apply voltage => change shape by 30% OR: …

MULTIPLE MANIPULATORS

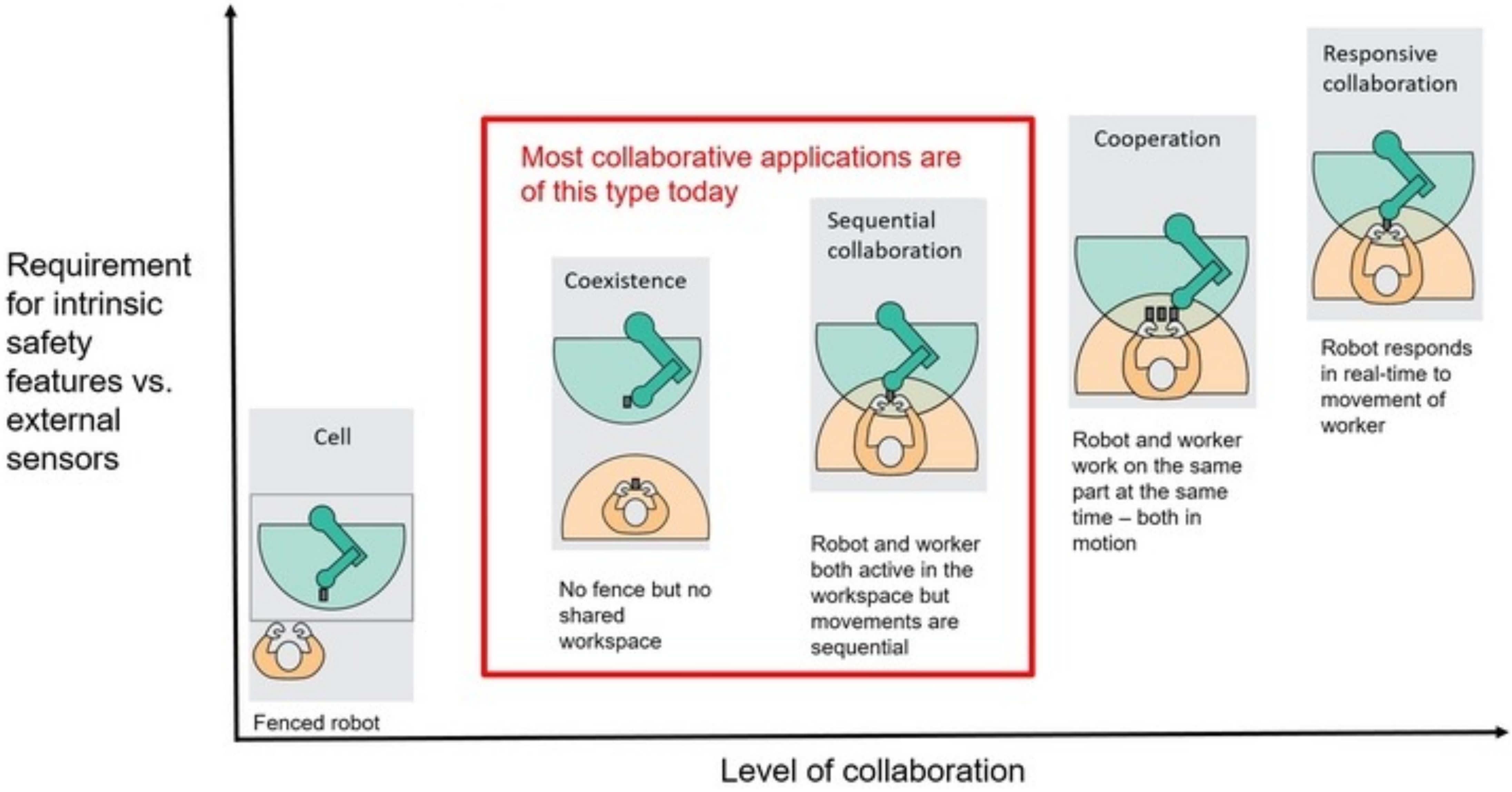
Multi-Robot & Human-Robot Co*****



- Often in terms of task and mission planning
  - E.g.: tidy up the room together, cook together, build a house together, search together, …
- Sometimes: Perception and/ or Control problem:
  - Typically when manipulating the same object (at the same time)
  - E.g.: two agents carrying a heavy object together, shaking hands, throwing & catching ball, …

Parker, L. E. (2007, November). Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems. In *AAAI fall symposium: regarding the intelligence in distributed intelligent systems* (pp. 1-6).

Types of collaboration with industrial robots



Green area: robot's workspace; yellow area: worker's workspace
Source: IFR (classification), adapted and modified from Bauer et al. (2016).

Industrial vs. Collaborative Robot Arms

Industrial Arms

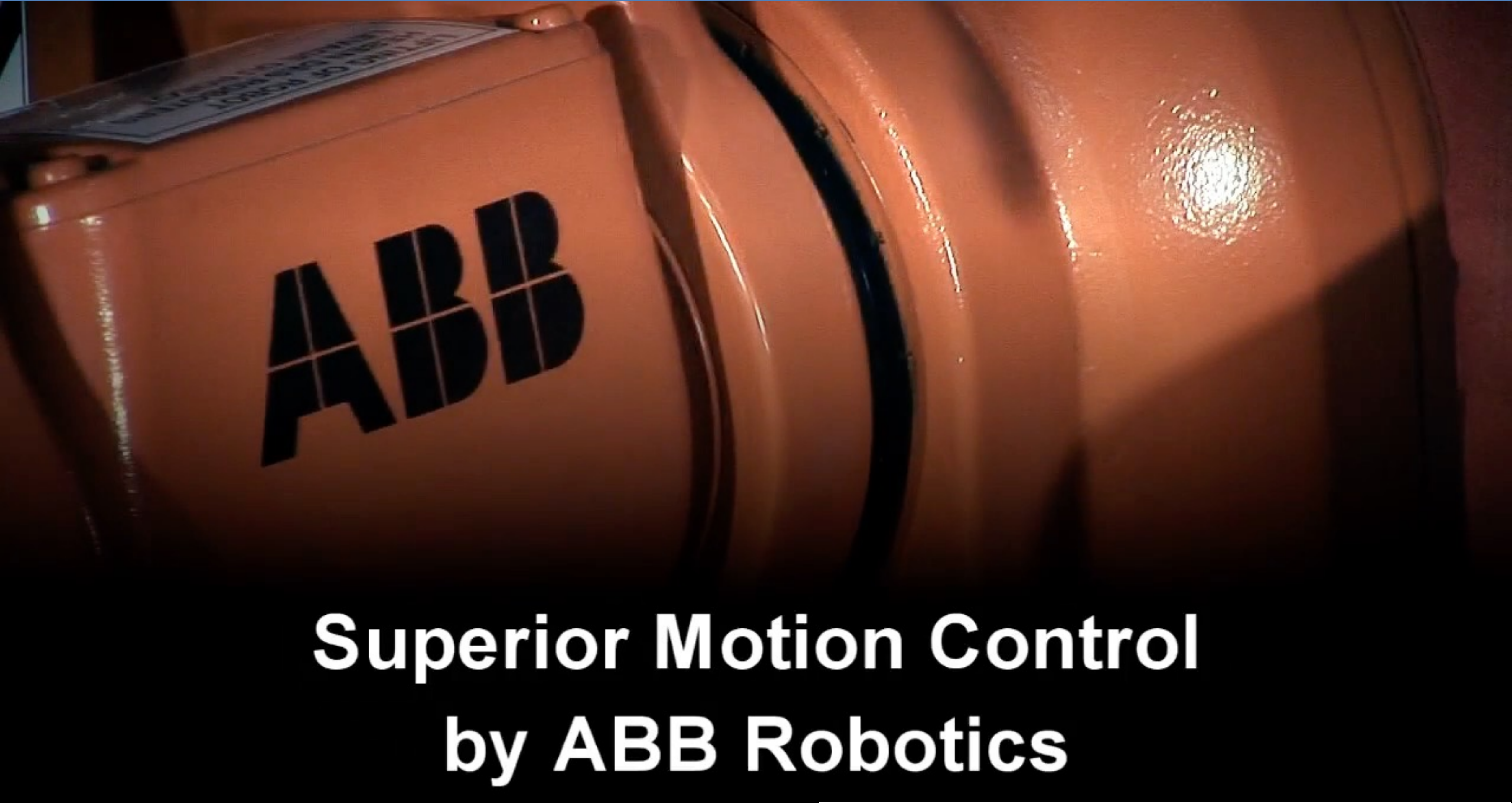
- Can be very precise (up to sub-mm)
- Can be very fast
- Can have very high payload
- May smack you over if you get in the way…

Collaborative Robots

- Often related to soft robotics (to a certain degree) because:
  - Inherent safety due to softness
- Often made compliant (you can move against them) – steer them
  - Also for teaching them easily
- Often less precise, slower, less payload

https://www.youtube.com/watch?v=ArBxq3mOt2s

MULTI-ROBOT KINEMATIC CONTROL



https://www.youtube.com/watch?v=SOESSCXGhFo