上海科技大学
ShanghaiTech University

CS283: **Robotics Fall 2020: The Mechatronics of Wheeled Locomotion**

Sören Schwertfeger / 师泽仁

ShanghaiTech University

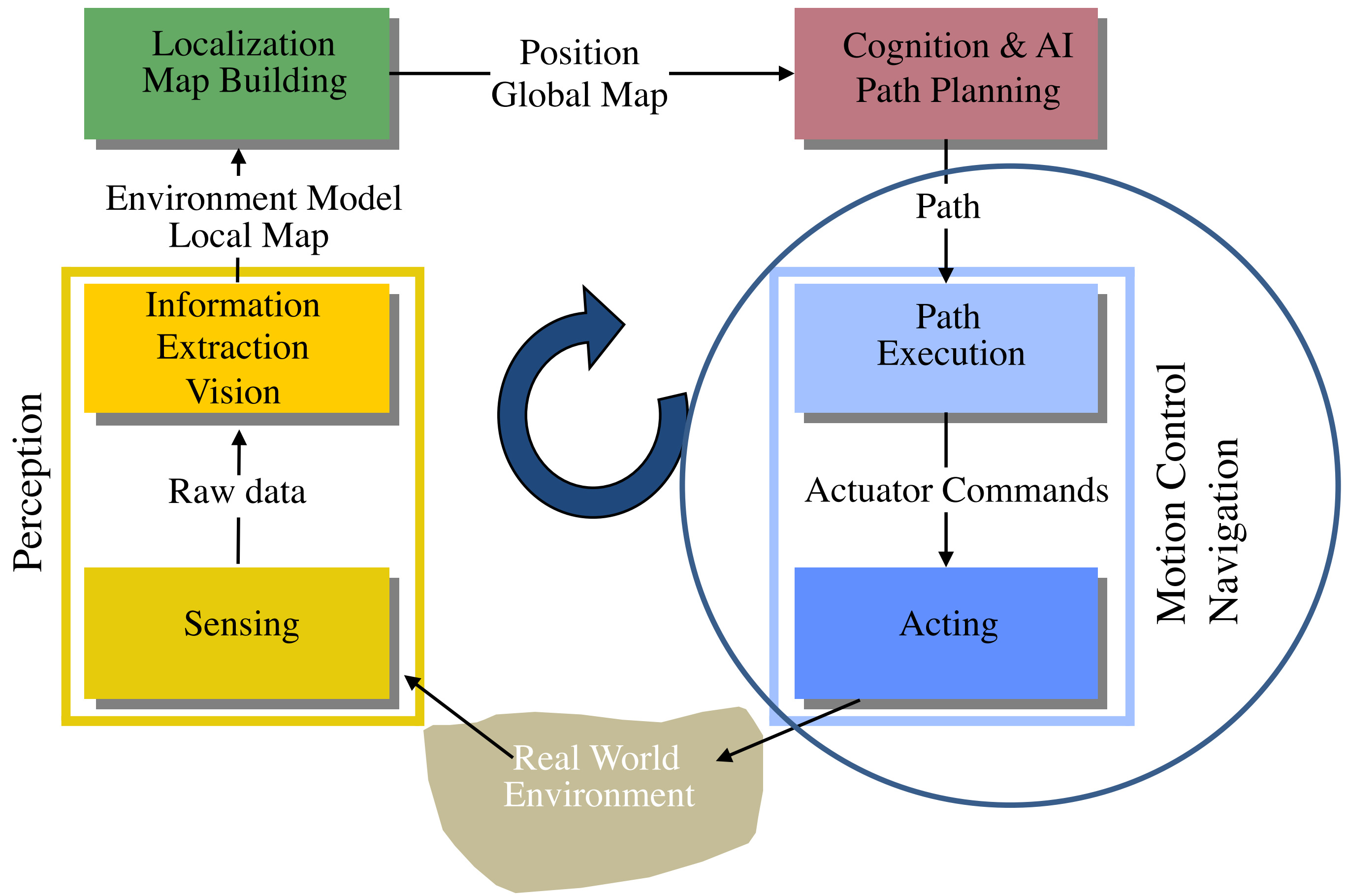
Admin

- Big part of this course is project => some weeks will skip a class in favor of you doing project
- Check course website (will be updated today)

- No class this Thursday

- Next week: Vision I and II on Tuesday and Thursday

General Control Scheme for Mobile Robot Systems



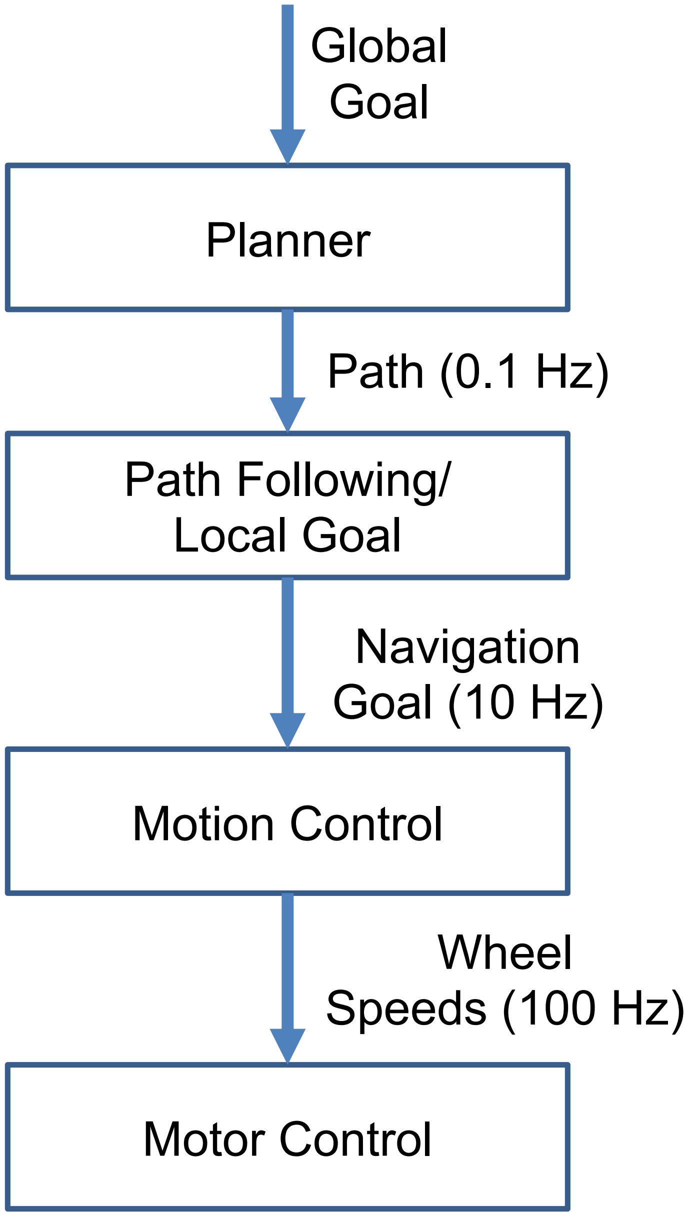
This lecture:
Control

- Autonomous mobile robots move around in the environment. Therefore **ALL** of them:
  - They need to know **where** they **are**.
  - They need to know **where** their **goal** is.
  - They need to know **how** to get there.

- Different levels:
  - Control:
    - How much power to the motors to move in that direction, reach desired speed
  - Navigation:
    - Avoid obstacles
    - Classify the terrain in front of you
    - Predict the behavior (motion) of other agents (humans, robots, animals, machines)
  - Planning:
    - Long distance path planning
    - What is the way, optimize for certain parameters

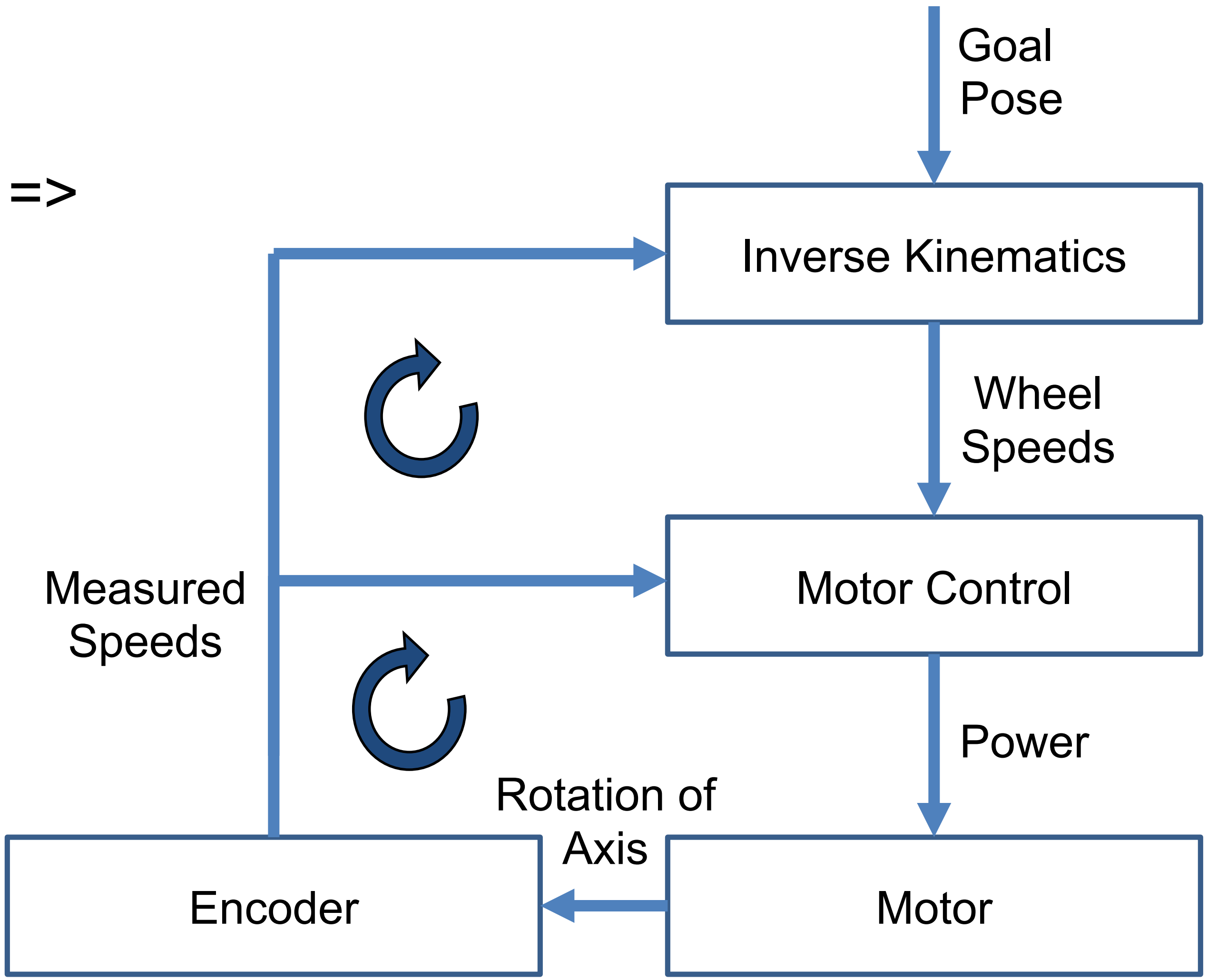
Navigation, Motion & Motor Control

- Navigation/ Motion Control:
  - Where to drive to **next** in order to reach goal
  - Output: motion vector (direction) and speed
  - For example:
    - follow path (Big Model)
    - go to unexplored area (Big Model)
    - drive forward (Small Model)
    - be attracted to goal area (Small Model)
- Motion Control:
  - How use propulsion to achieve motion vector
- Motor Control:
  - How much power to achieve propulsion (wheel speed)

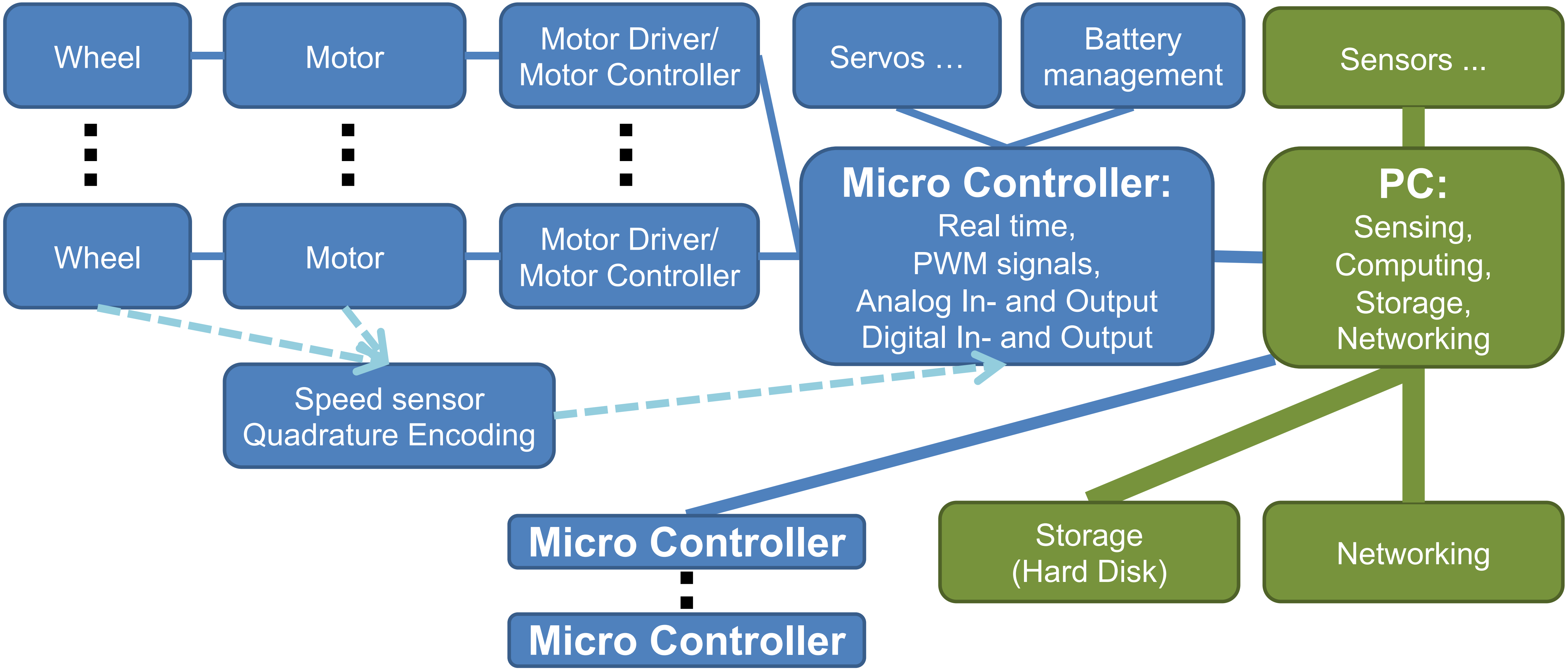


Control Hierarchy

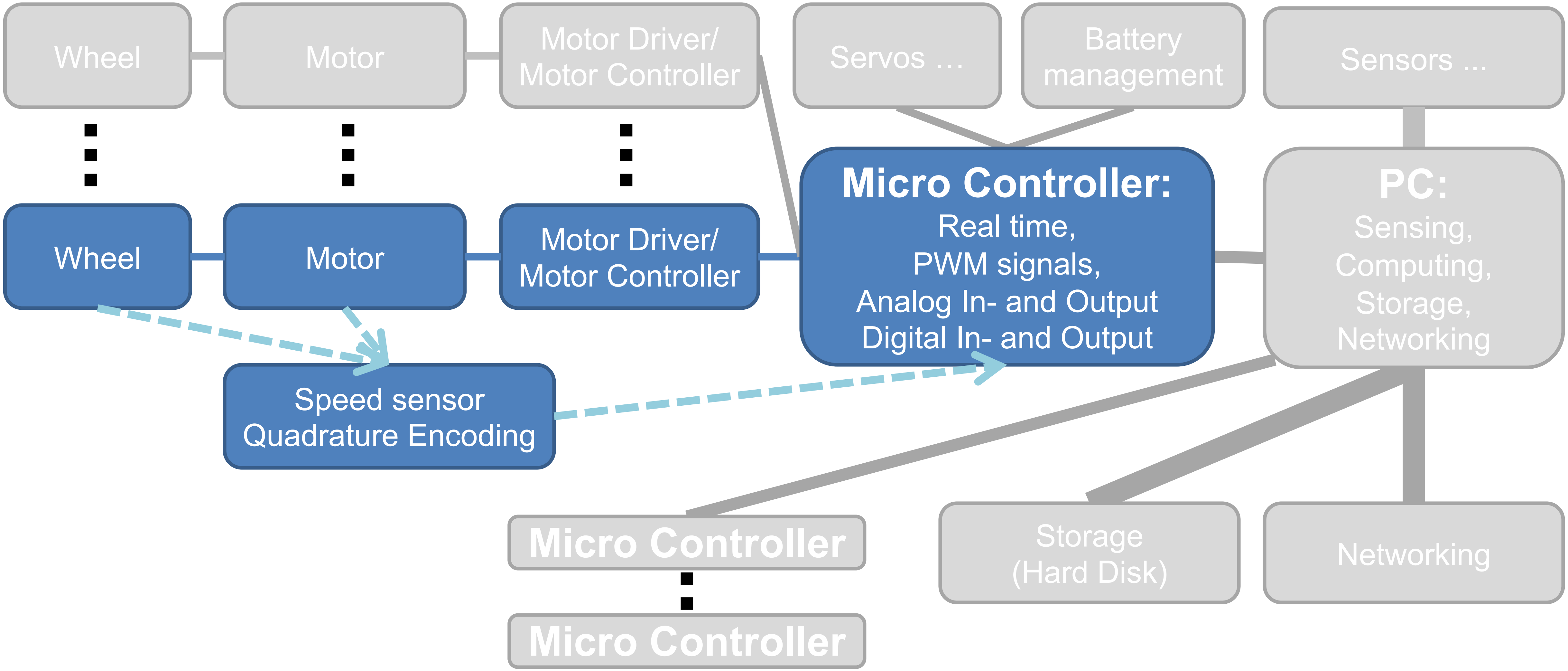
- Assume we have a goal pose (close by)
- Calculate Inverse Kinematics =>
- Desired wheel speeds
    - Typically not just one wheel =>
    - Many motor controllers, motors, encoders
- Motor control loop
- Pose control loop



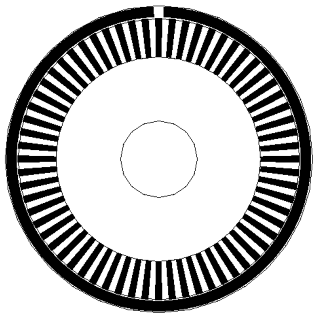
Overview Hardware

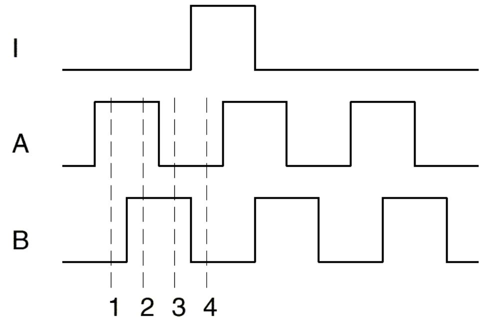


The Mechatronics of Wheeled Locomotion



DC Motor with Gearbox and Quadrature Encoder

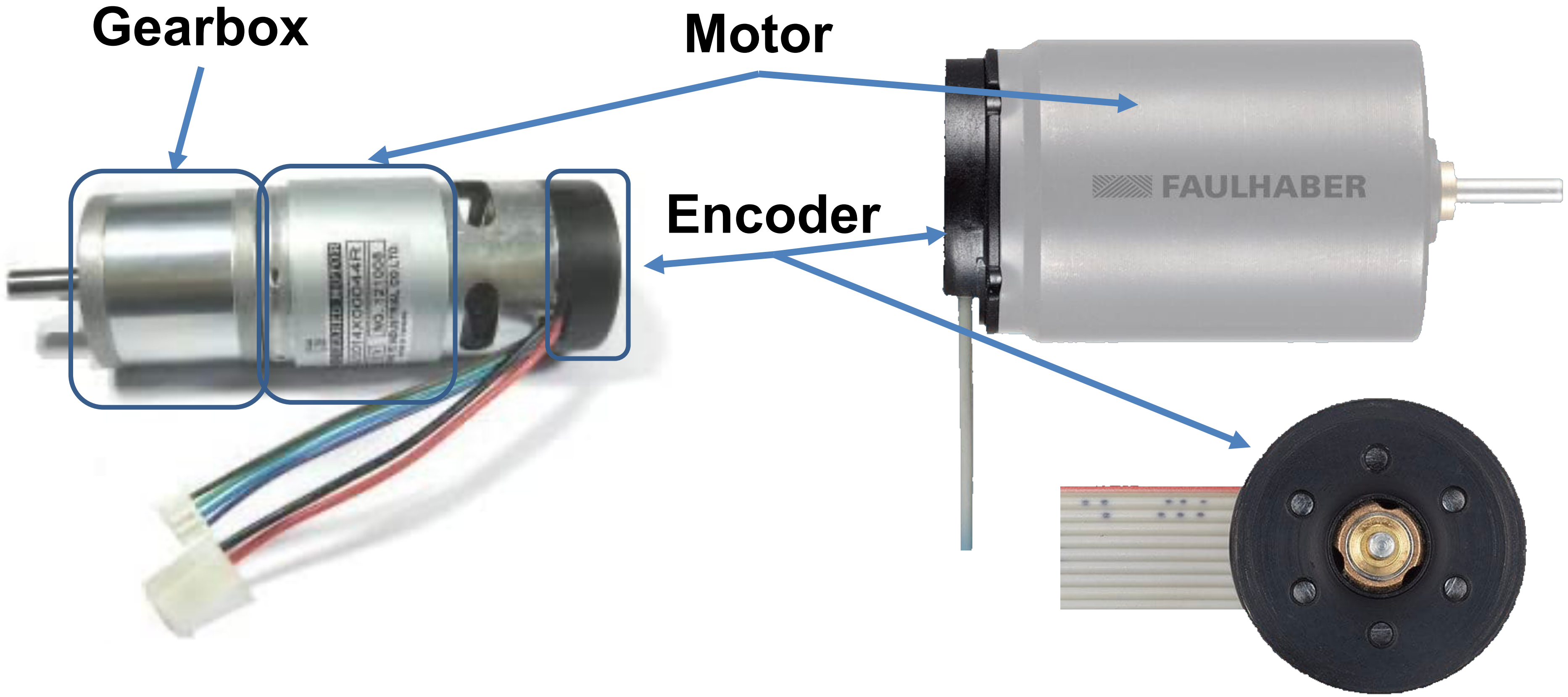




State	Ch A	Ch B
S_1	High	Low
S_2	High	High
S_3	Low	High
S_4	Low	Low

Encoder Wires
Enc GND
Enc Vcc (5V)
Ch A
Ch B
Index (Optional)
Optional: Negative Signals for saver transmission

DC Motor Wires
Motor A
Motor B



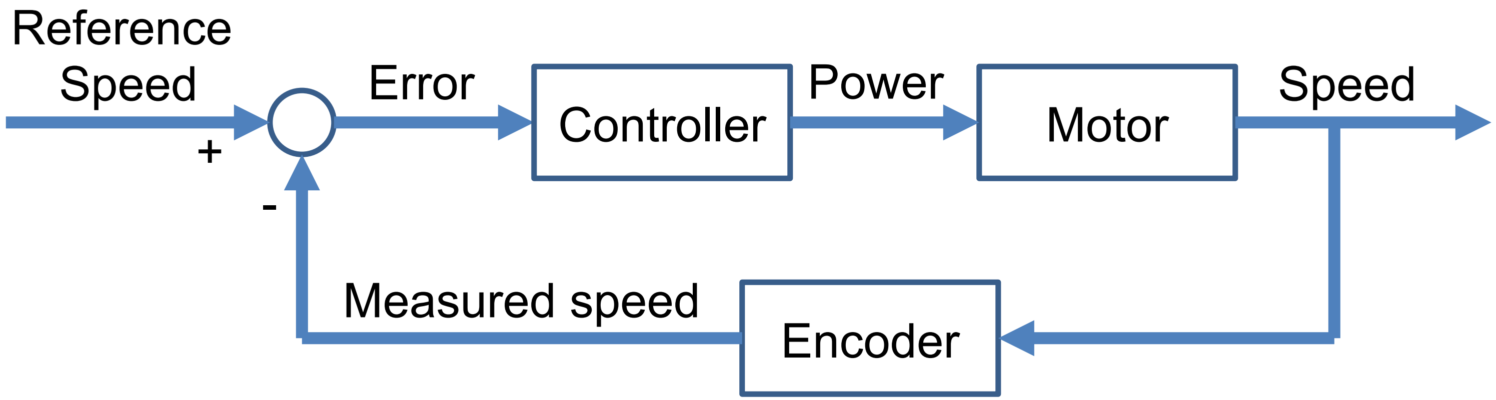
Provide Brief Overview of the following topics:

- PID Control
- PWM Signal
- Motor Driver
- DC brushed and brushless Motors & Servos
- Gears

PID CONTROL

Motor Control

- How much power is needed for desired reference speed?
  - Inertia of the motor + robot
  - Friction
  - Need more power during acceleration of robot vs. constant speed
  - Up hill/ down hill different power needs
  - Motors are even used to break the robot!
- Closed loop control (negative feedback)
- Proportional-Integral-Derivative Controller (PID)
- Motor speed reacts slowly to power changes

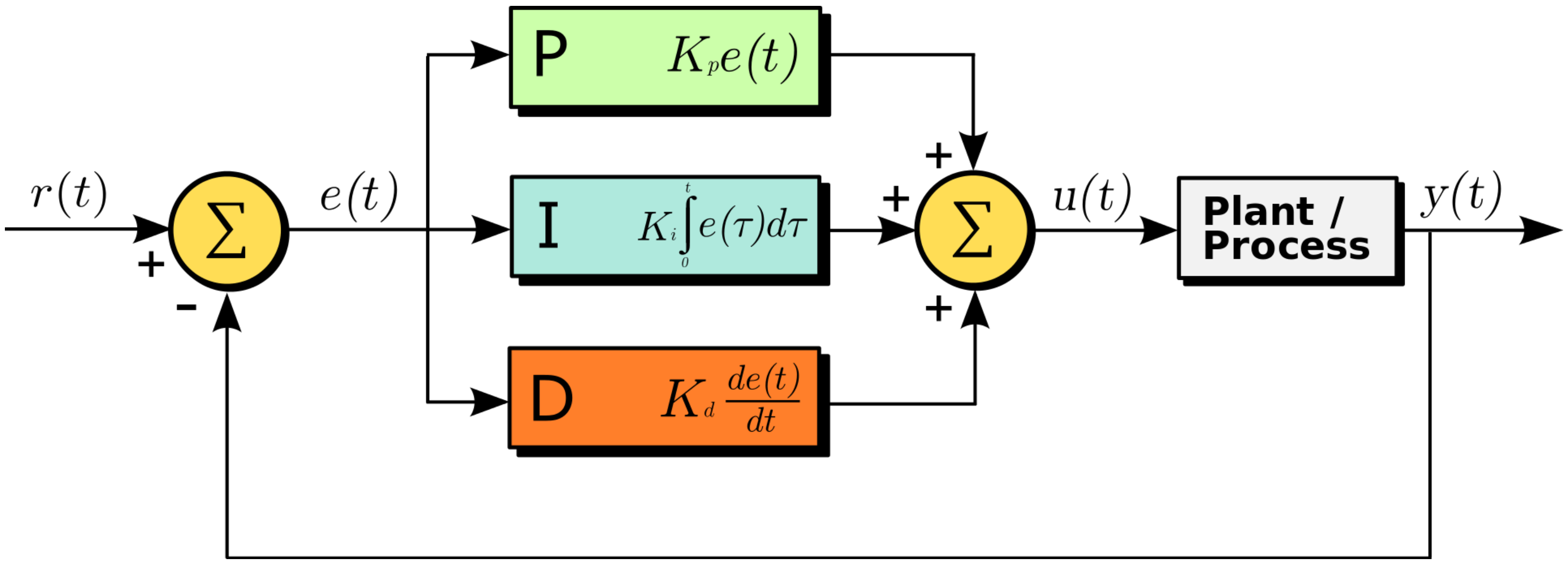


PID: Proportional-Integral-Derivative Controller

- Input: Desired Speed (of wheel/ motor)
  - Actually: Error of the current speed (process variable) to the desired speed (setpoint)
- Output: Amount of power to the motor
- Not needed: Model of the plant process (e.g. motor, robot & terrain parameters)
- Parameters:
  - K_p proportional gain constant
  - K_i integral gain
  - K_d derivative gain
- Discrete Version:

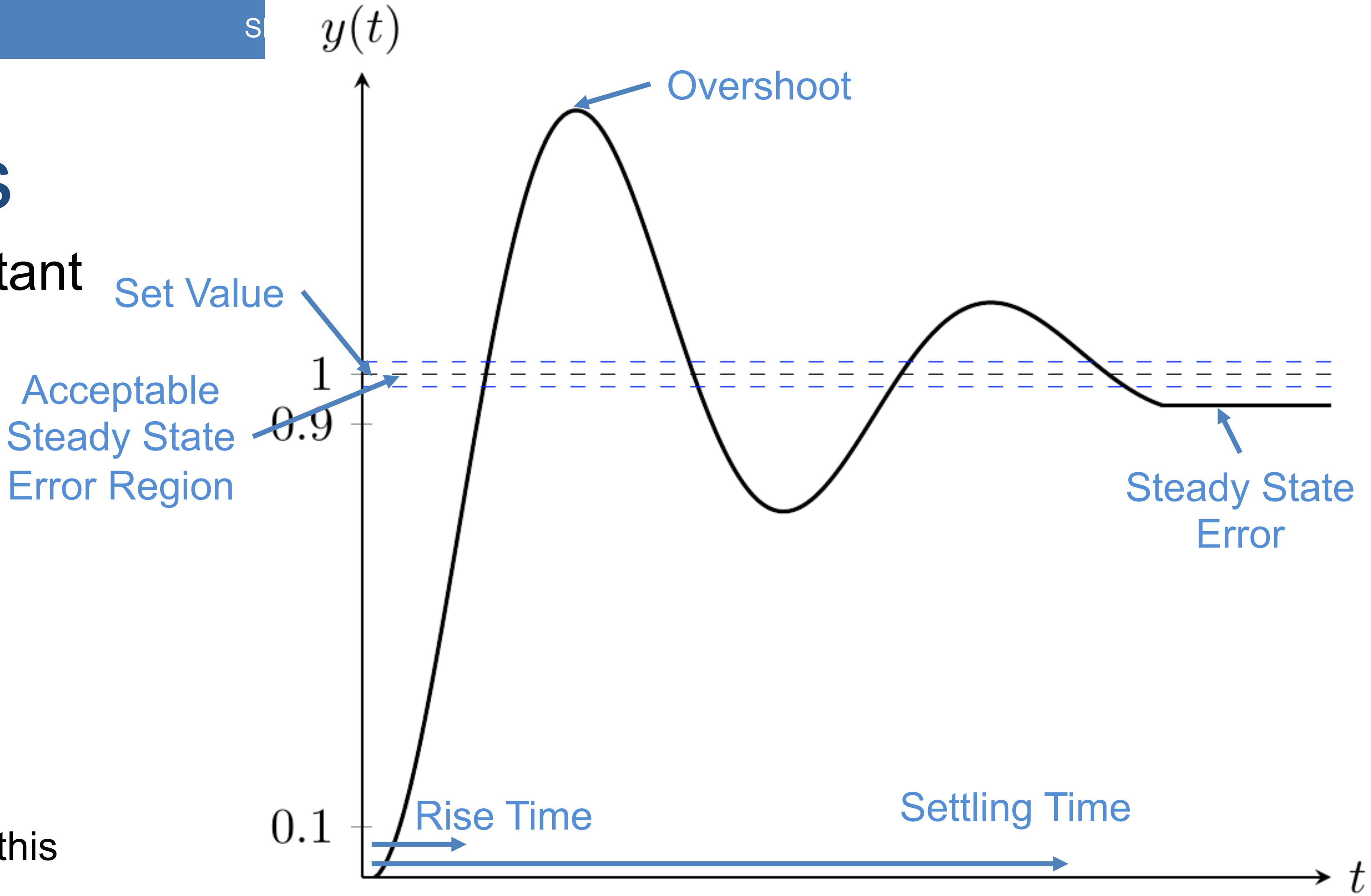
$$\int_0^{t_k} e(\tau)\, d\tau = \sum_{i=1}^{k} e(t_i)\, \Delta t$$

$$\frac{de(t_k)}{dt} = \frac{e(t_k) - e(t_{k-1})}{\Delta t}$$



Tune Parameters

- K_p proportional gain constant
  - Too small: long rise time
  - Too big: big overshoot or even unstable control
  - Should contribute most of the output change
- K_i integral gain
  - Reduces steady state error
  - May cause overshoot
    - Leaky integration may solve this
- K_d derivative gain
  - Predicts error by taking slope into account
  - May reduce settling time and overshoot



Parameter Increase	Rise time	Overshoot	Settling Time	Steady-state error
Kp	↓	↑	Small Change	↓
Ki	↓	↑	↑	Great reduce
Kd	Small Change	↓	↓	Small Change

Table (2) PID controller parameter characteristics on a fan's response

https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2012/fas57_nyp7/Site/pidcontroller.html

Control Theory

- Other controllers used
  - P Controller
  - PD Controller
  - PI Controller

```
previous_error := 0
integral := 0

loop:
    error := setpoint − measured_value
    integral := integral + error × dt
    derivative := (error − previous_error) / dt
    output := Kp × error + Ki × integral + Kd × derivative
    previous_error := error
    wait(dt)
    goto loop
```

Pseudo Code PID Controller

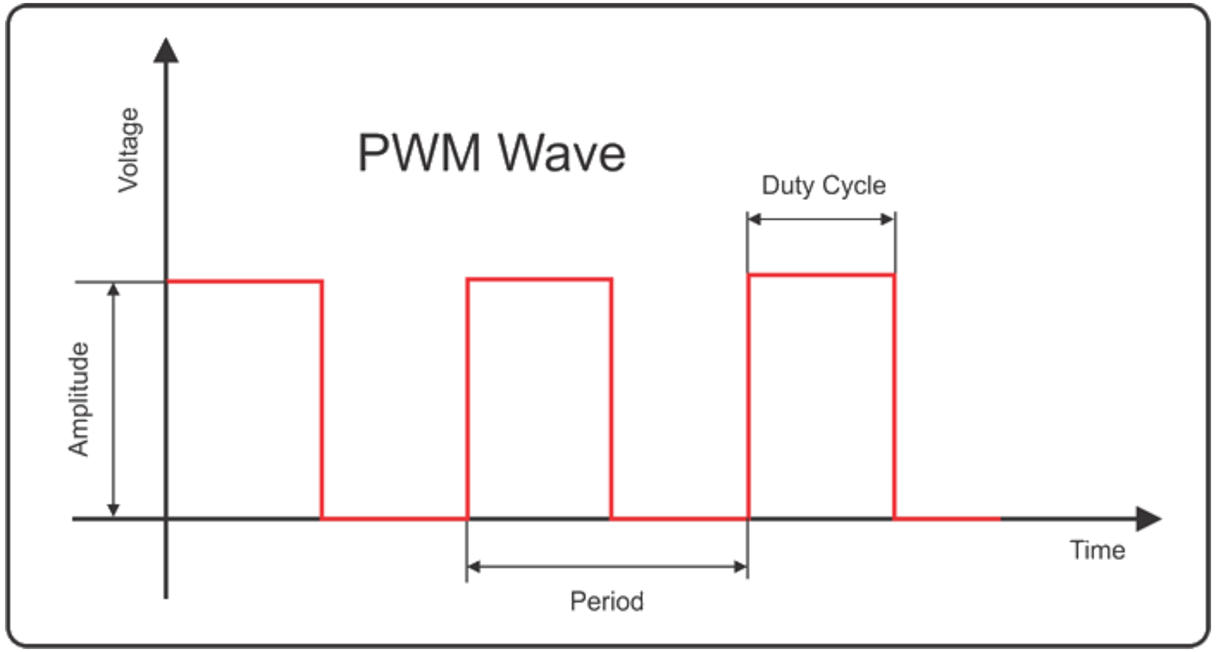
- PID sufficient for most control problems
- PID works well if:
  - Dynamics of system is small
  - System is linear (or close to)
- Control Theory
  - EE 160: Introduction to Control (Prof. Houska)
  - EE264: Adaptive Control (Prof. Yang Wang)

- Popular alternative: Model Predictive Control (MPC)
  - Optimal Control Technique: satisfy a set of constraints
  - Finite time horizon to look into the future (“plan”)
  - Used when PID is not sufficient; e.g.:
    - Very dynamic system
    - Second order system (oscillating system)
    - Multi-variable control
  - Use Cases: Chemical plants; planes; robot arms

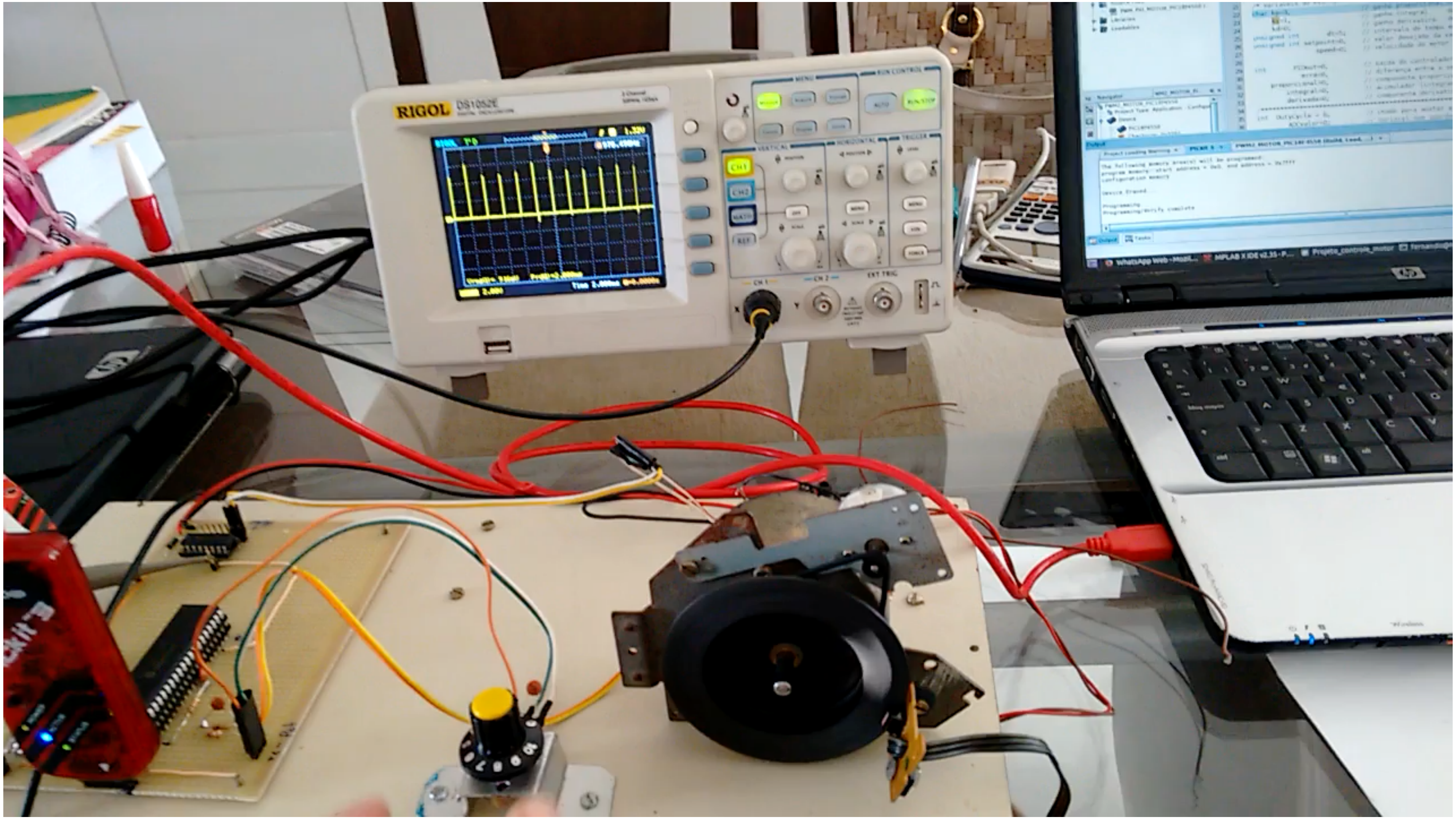
PULSE WIDTH MODULATION

Image: zembedded.com

Pulse Width Modulation



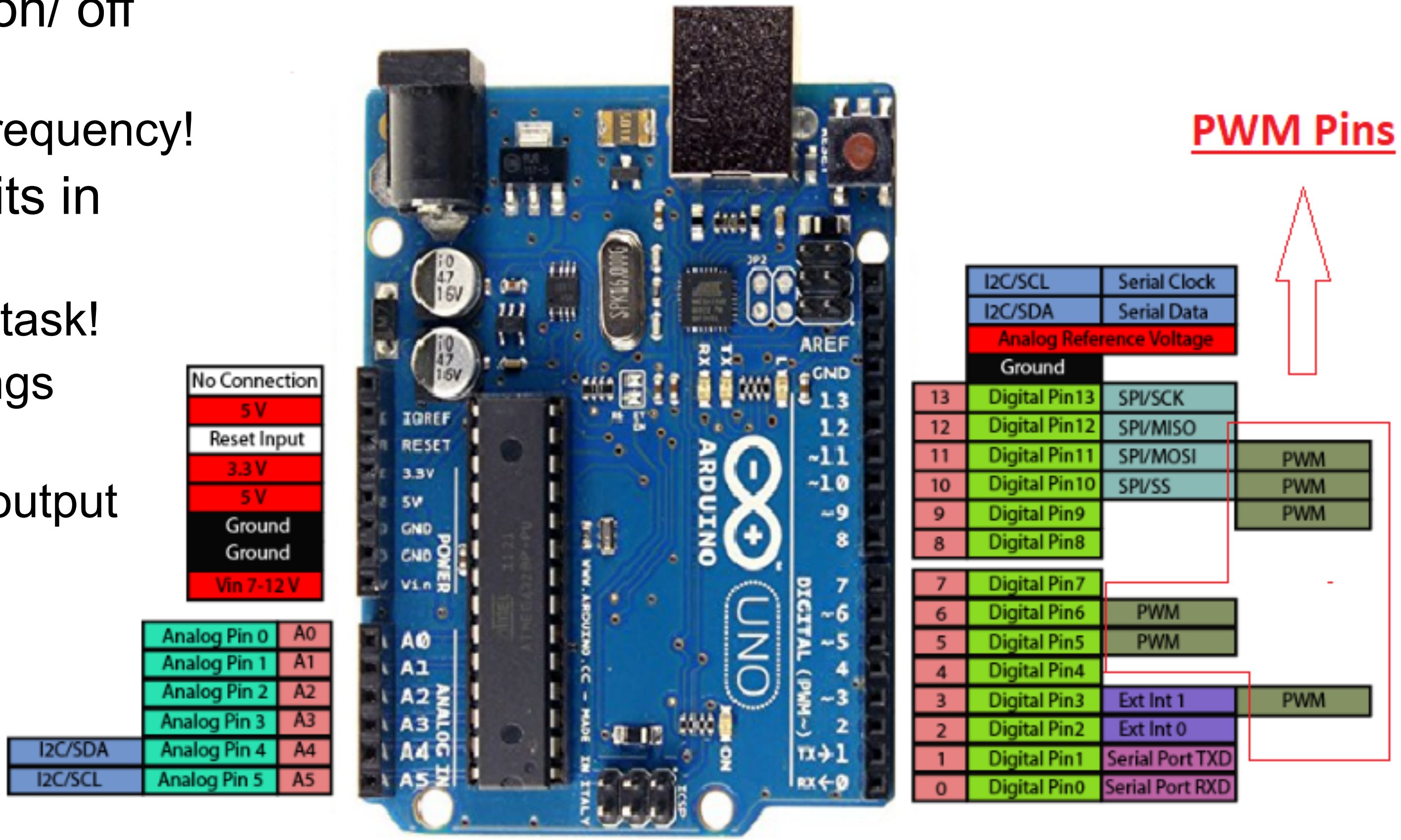
- How can Controller control power?
  - Cannot just tell the motor “use more power”
  - Output of (PID) controller is a signal
  - Typical: Analogue signal
- Pulse Width Modulation (PWM)
  - Signal is either ON or OFF
  - Ratio of time ON vs. time OFF in a given interval: amount of power
  - Frequency 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

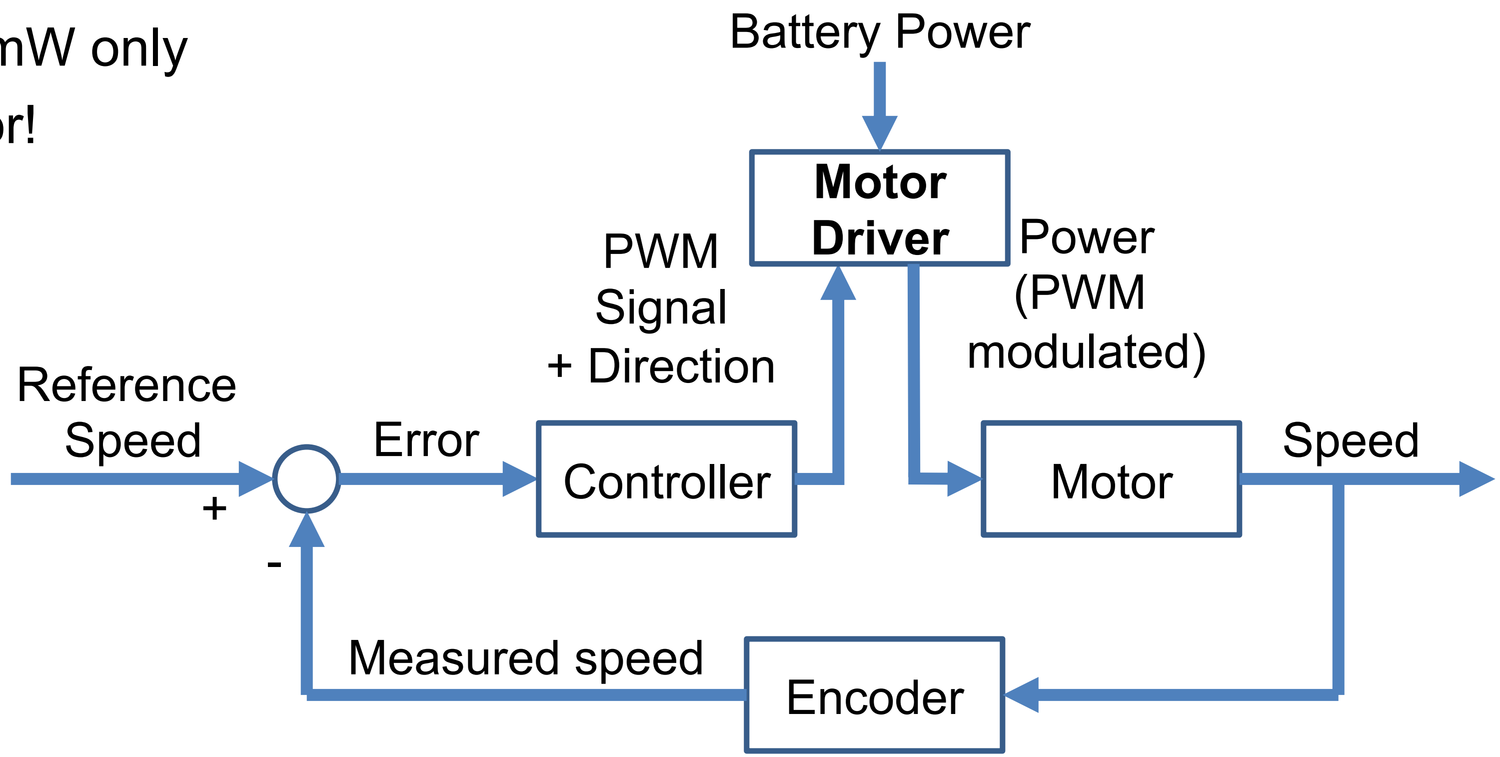


MOTOR DRIVER

Power to the Motor

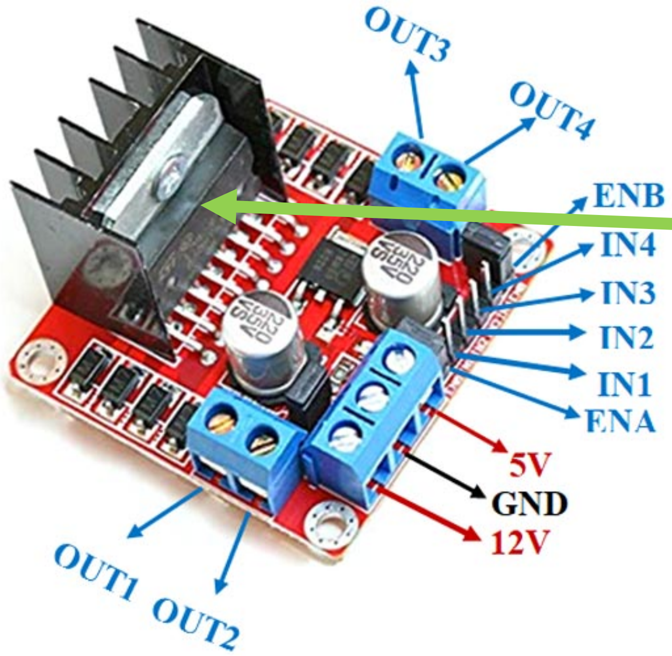
- Direct Current Motor (DC Motor):
  - Two wires for power input
  - Directly connect DC motor to PWM signal?
  - Limited current!
  - E.g.: Arduino: max 30mA => 150mW only
  - Clearpath Jackal: 250W per motor!

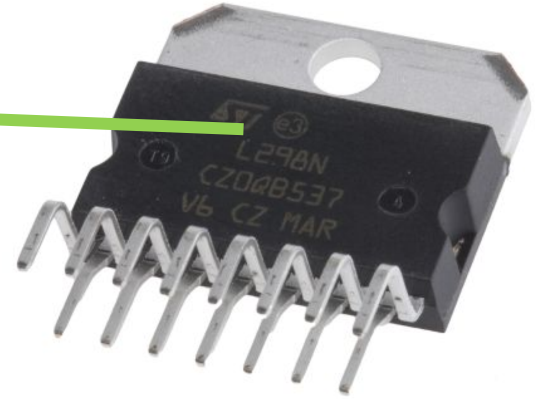
- Need a device to power the motor
- Mobile robots: battery power!

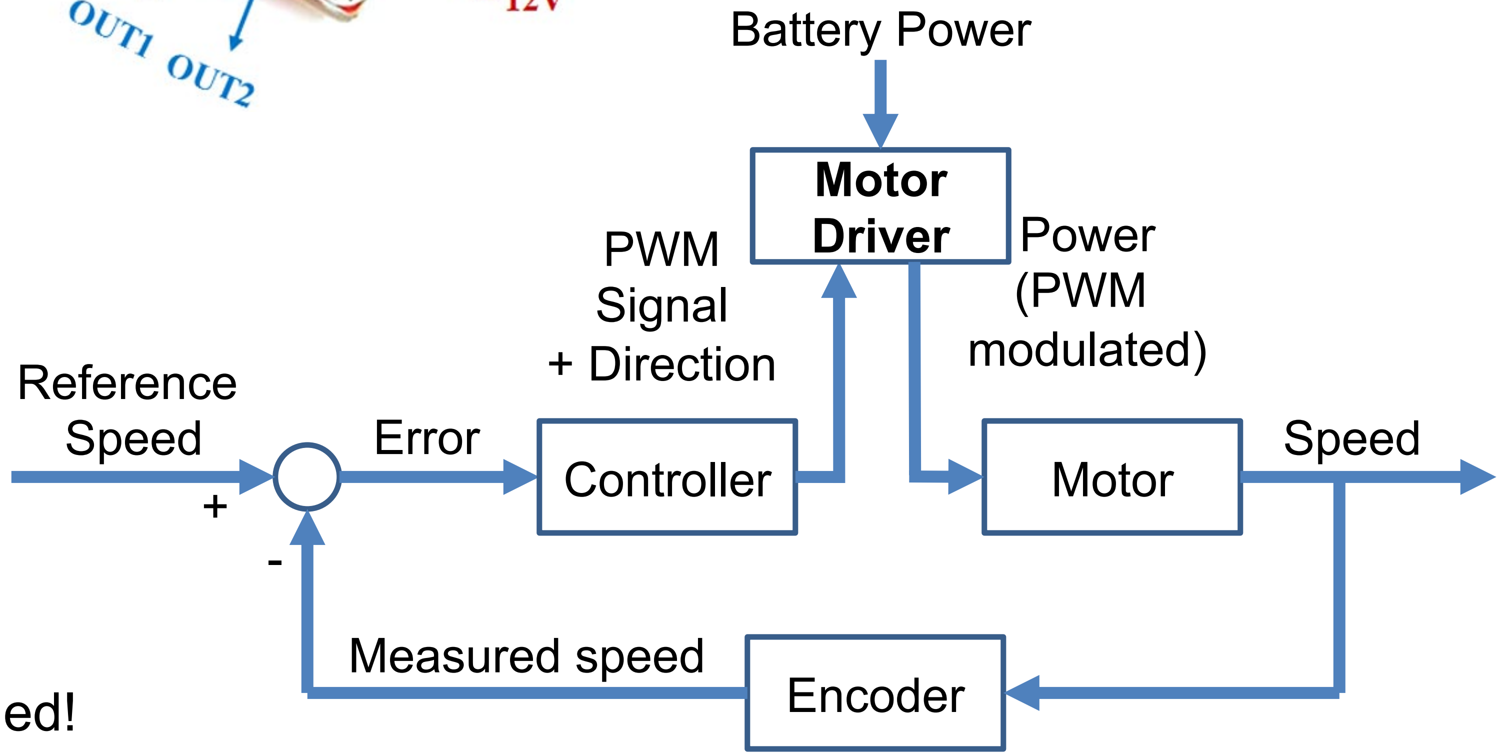


Motor Driver

- 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!

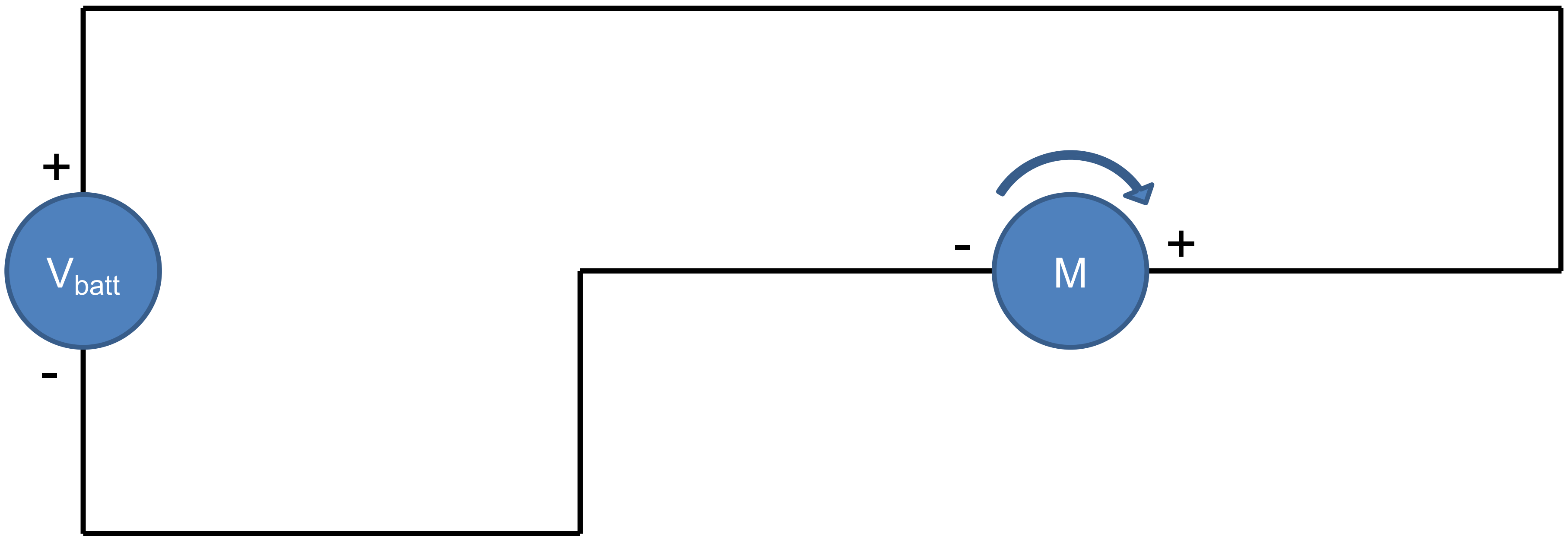






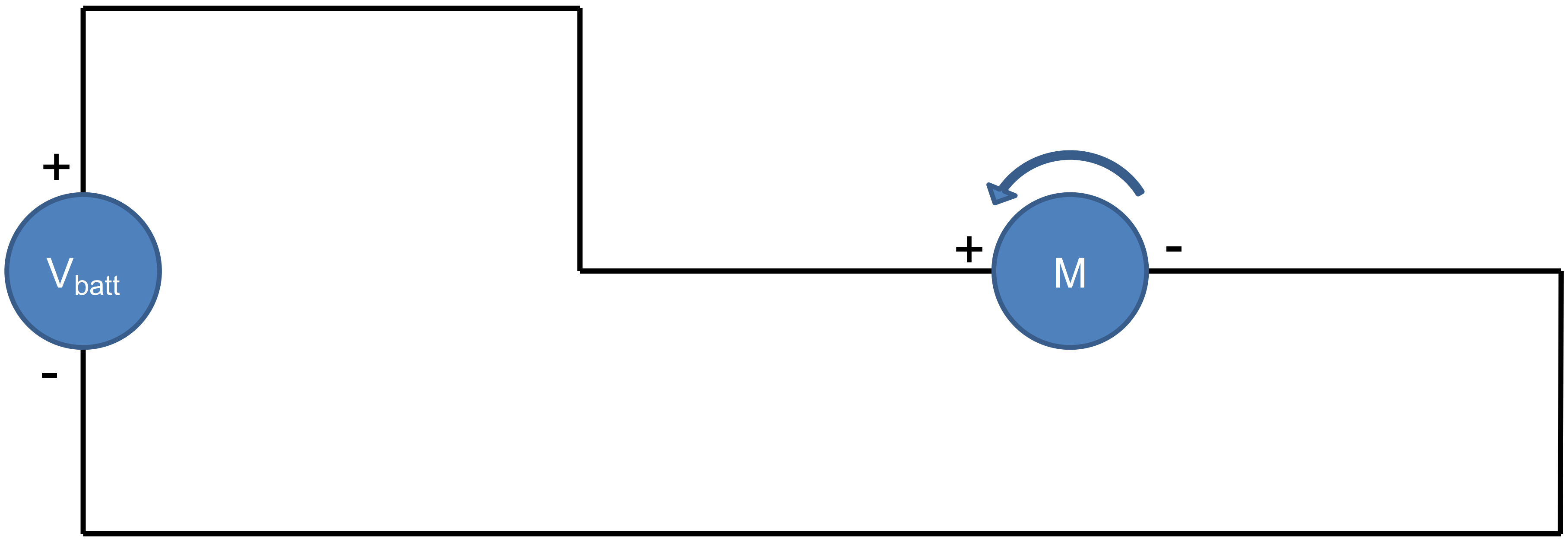
How does a Motor Driver work? H-Bridge

- H-Bridge:
  - Change direction of energy flow -> change direction of motor
  - Also: switch motor on and off



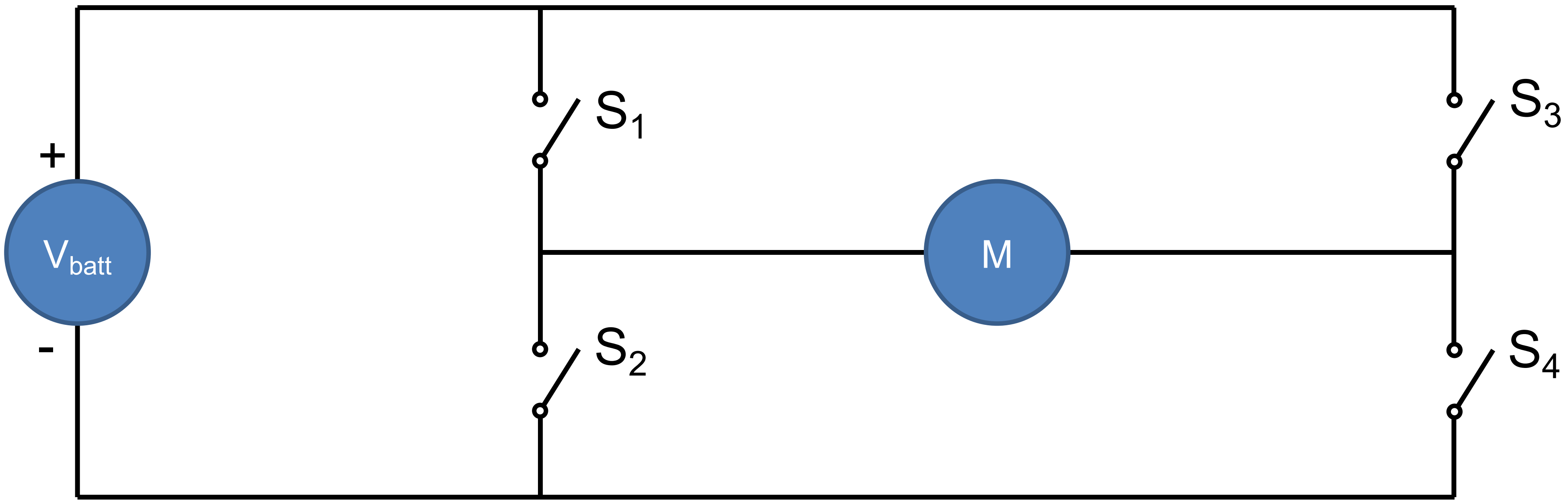
How does a Motor Driver work? H-Bridge

- H-Bridge:
  - Change direction of energy flow -> change direction of motor
  - Also: switch motor on and off



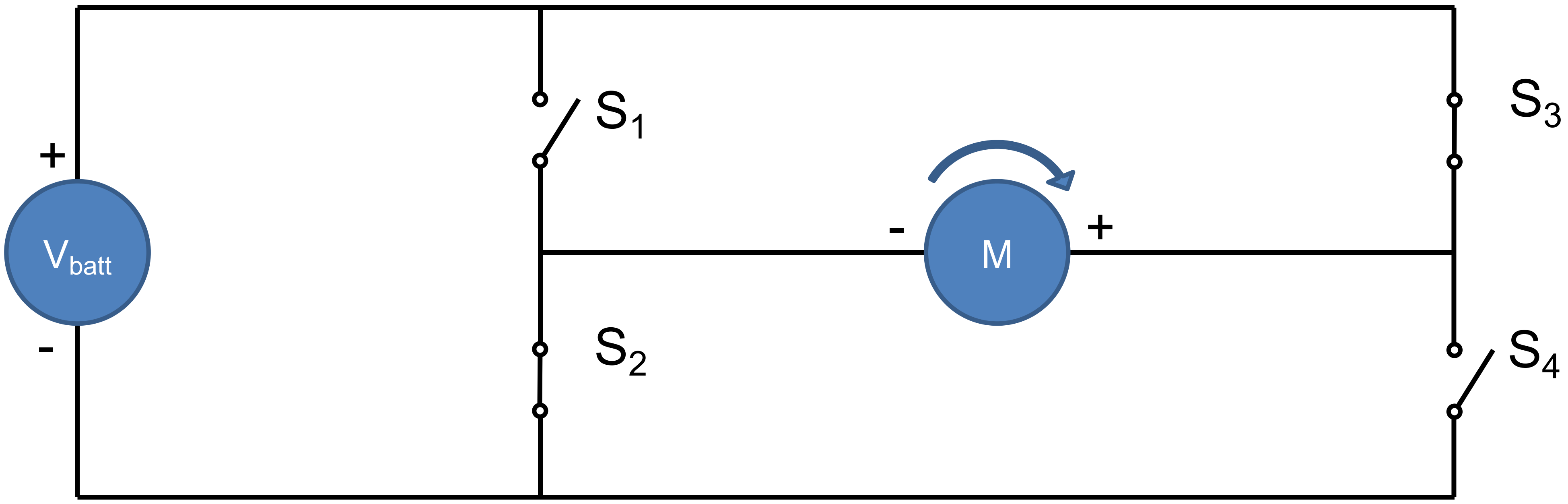
How does a Motor Driver work? H-Bridge

- H-Bridge:
  - Change direction of energy flow -> change direction of motor
  - Also: switch motor on and off



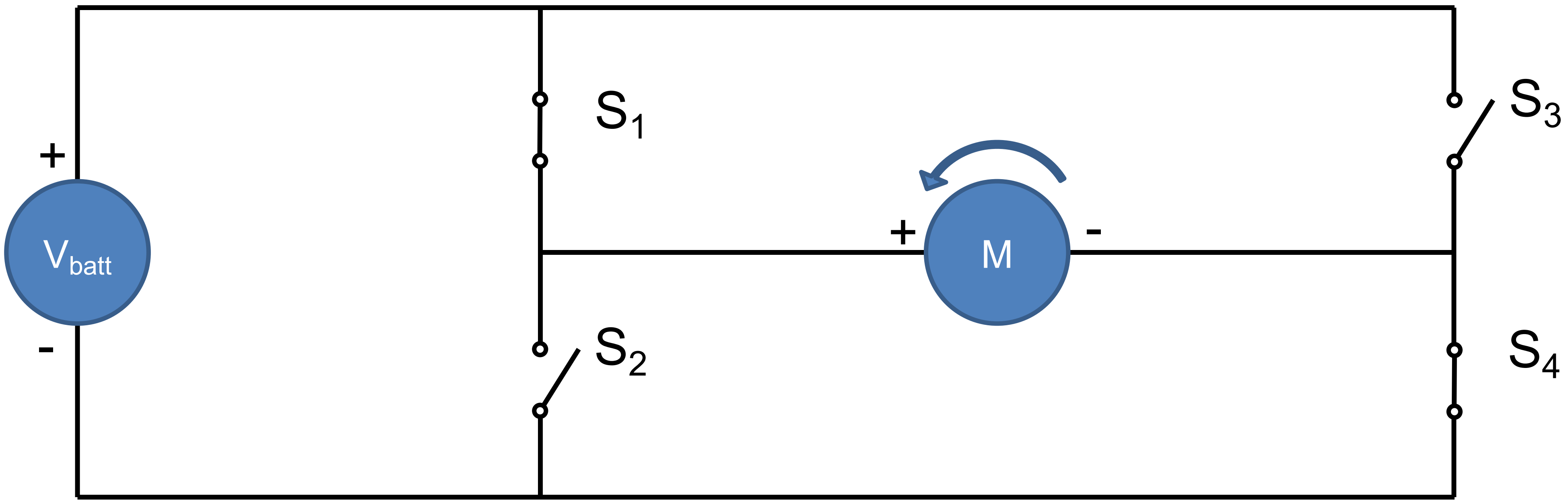
How does a Motor Driver work? H-Bridge

- H-Bridge:
  - Change direction of energy flow -> change direction of motor
  - Also: switch motor on and off



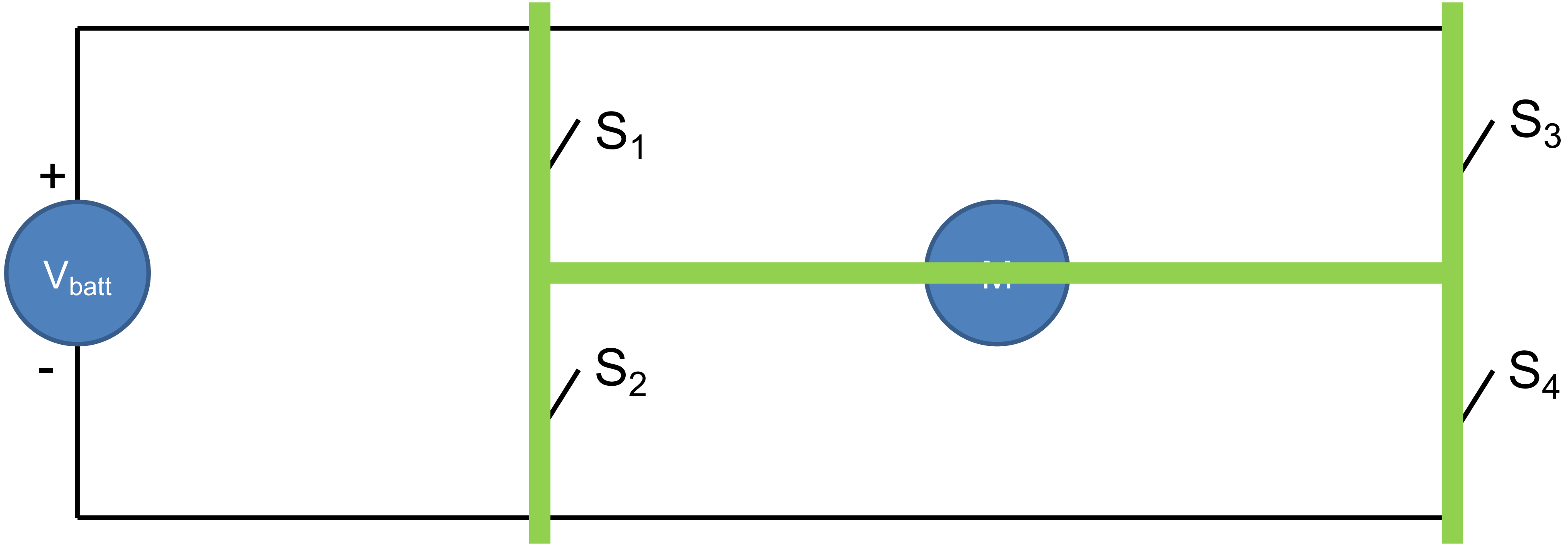
How does a Motor Driver work? H-Bridge

- H-Bridge:
  - Change direction of energy flow -> change direction of motor
  - Also: switch motor on and off



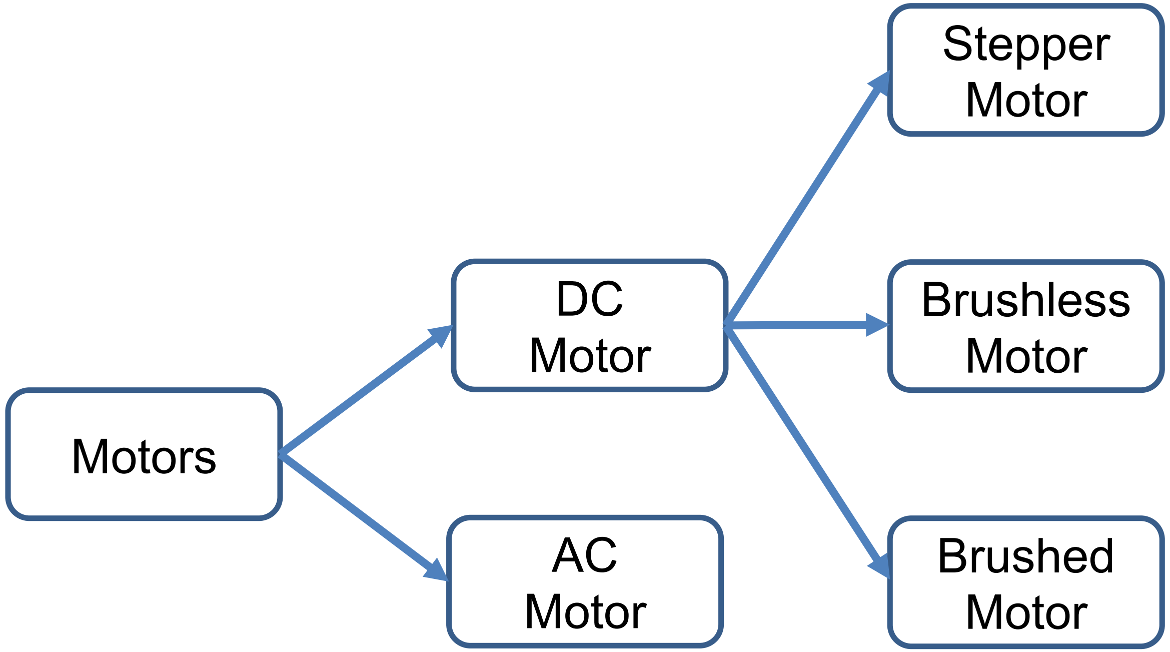
How does a Motor Driver work? H-Bridge

- H-Bridge:
  - Change direction of energy flow -> change direction of motor
  - Also: switch motor on and off



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

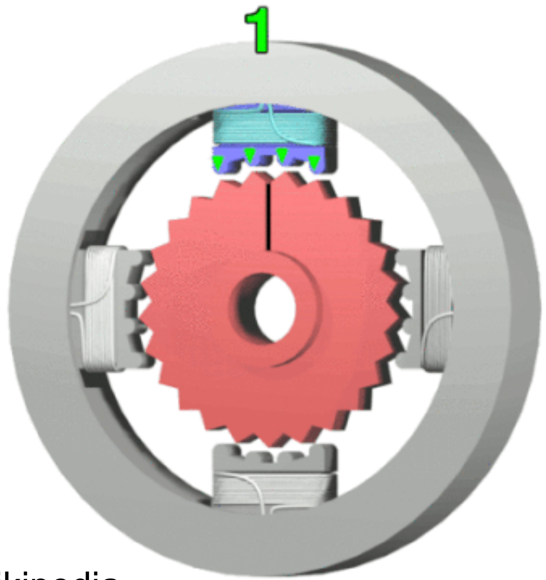
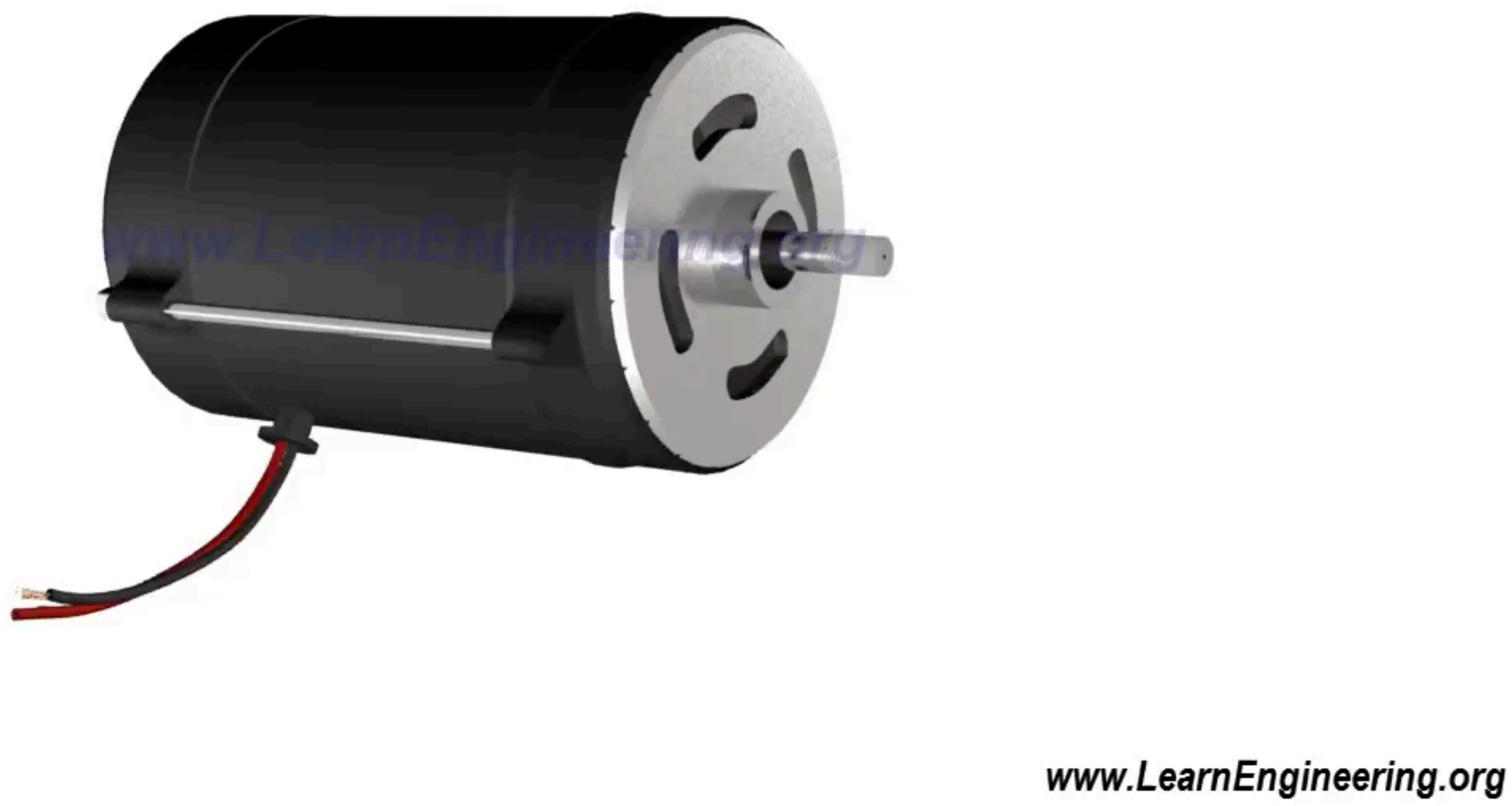


Image: Wikipedia

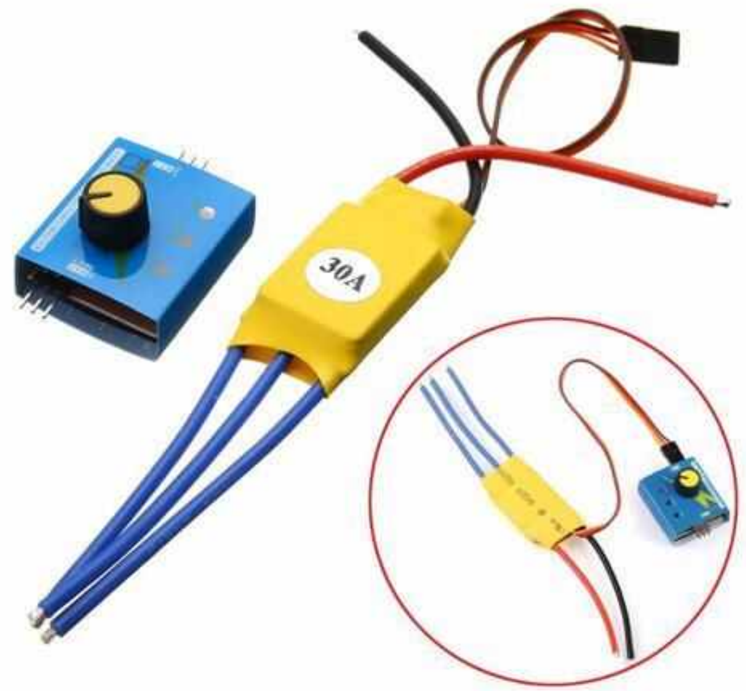


https://www.youtube.com/watch?v=CWulQ1ZSE3c

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

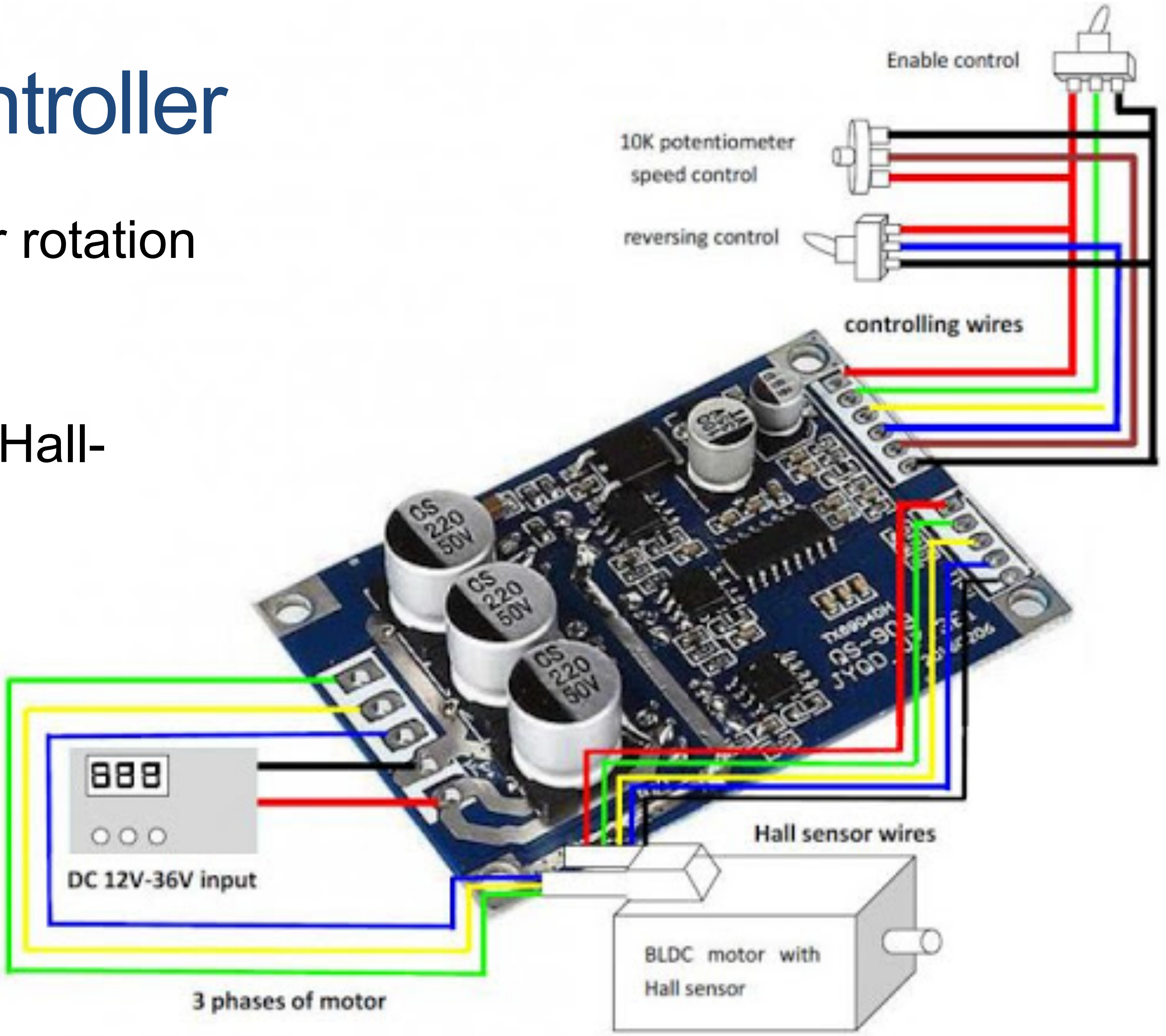
Brushless Motor Controller

- 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



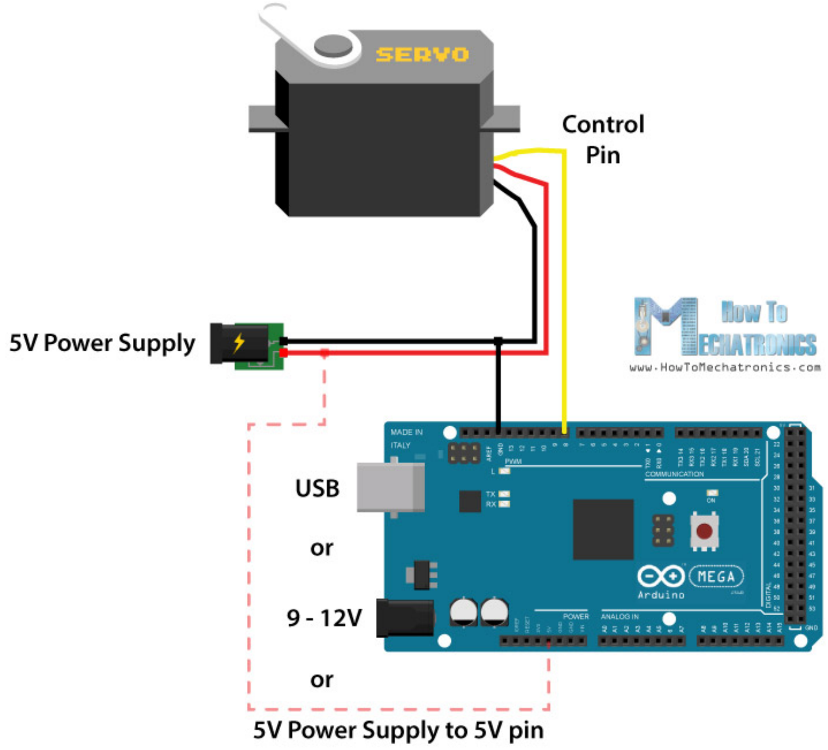
Brushless Motor Controller

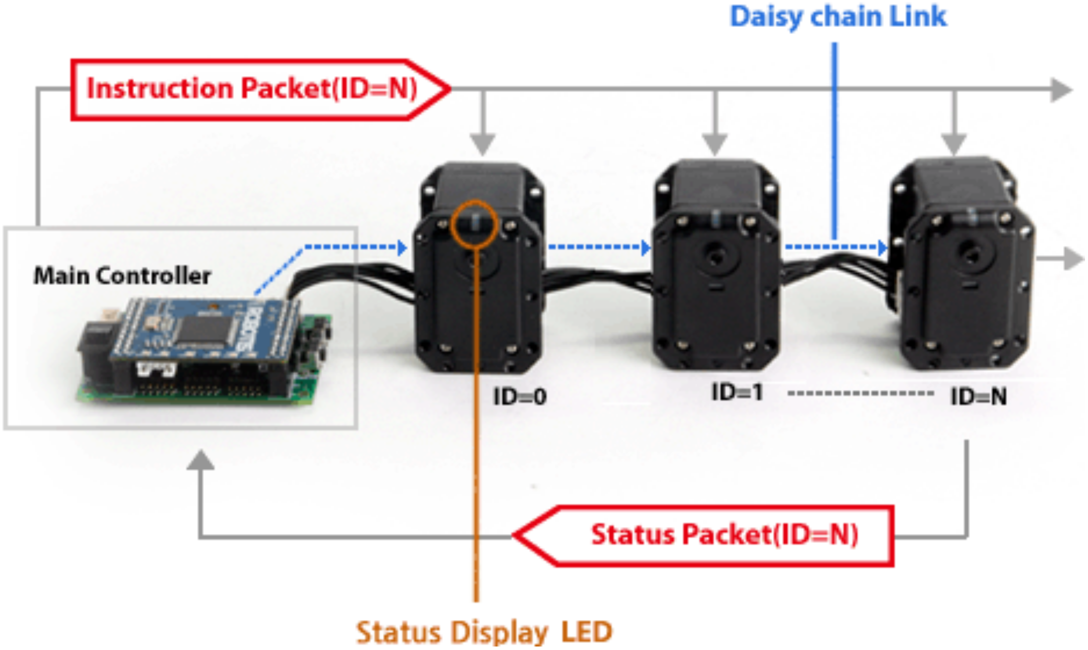
- Hall sensor only 3 positions per rotation
  - Quadrature encoder: up to 4096
- For high torque; low speeds: 3 Hall-effect sensors needed!
- External PID speed control may still be needed!
- Brushless: 20%-30% better efficiency



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

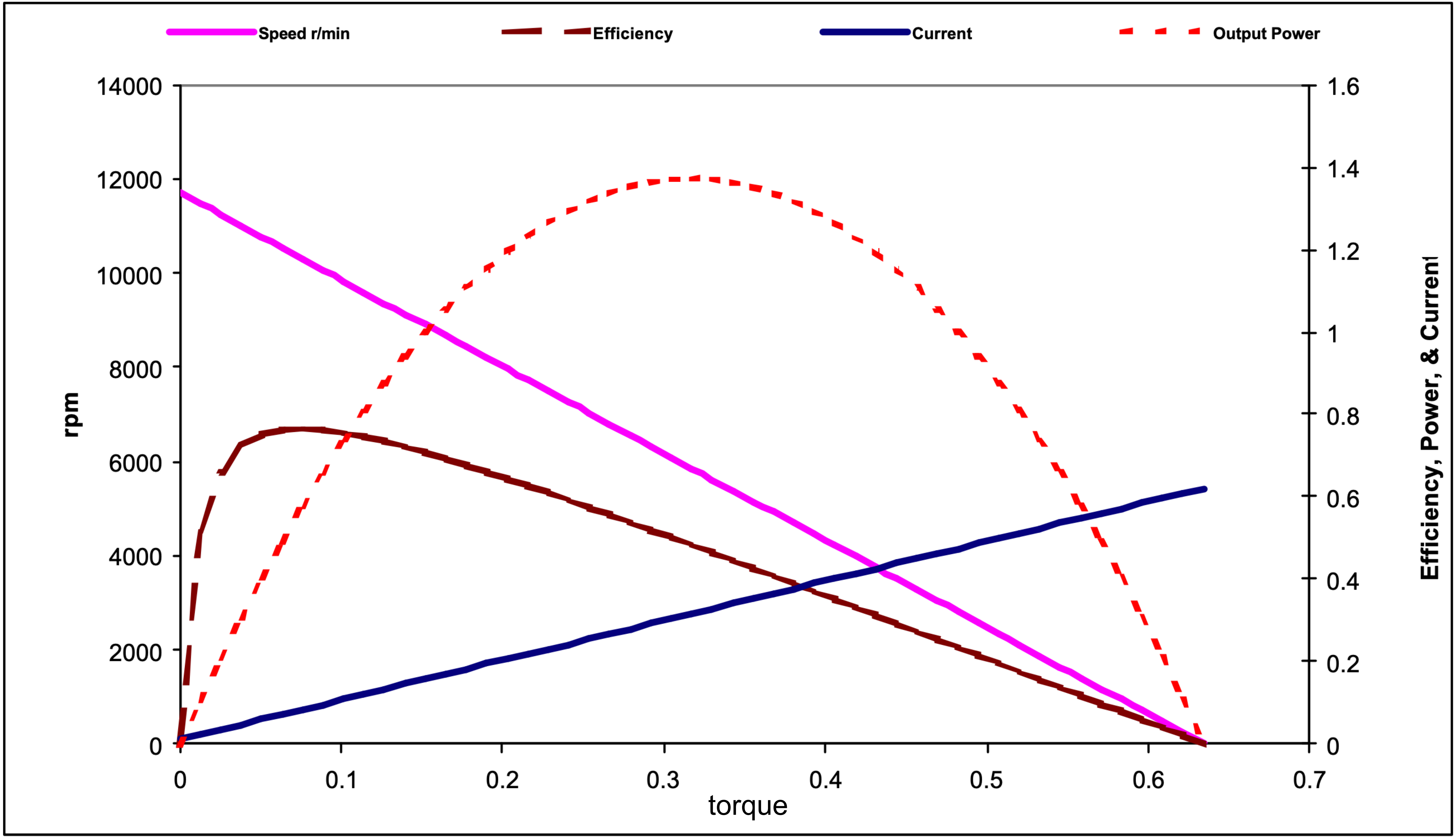




DC Motor Characteristics

- 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

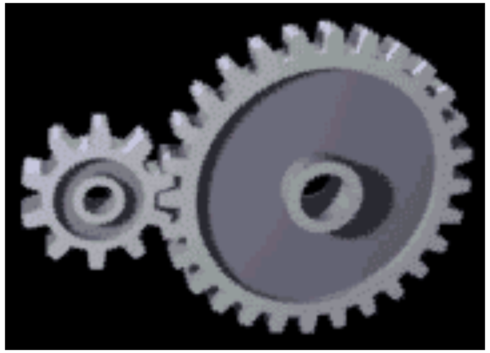
- High-power motors (e.g. humanoid robots) get very hot/ need cooling!


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

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…



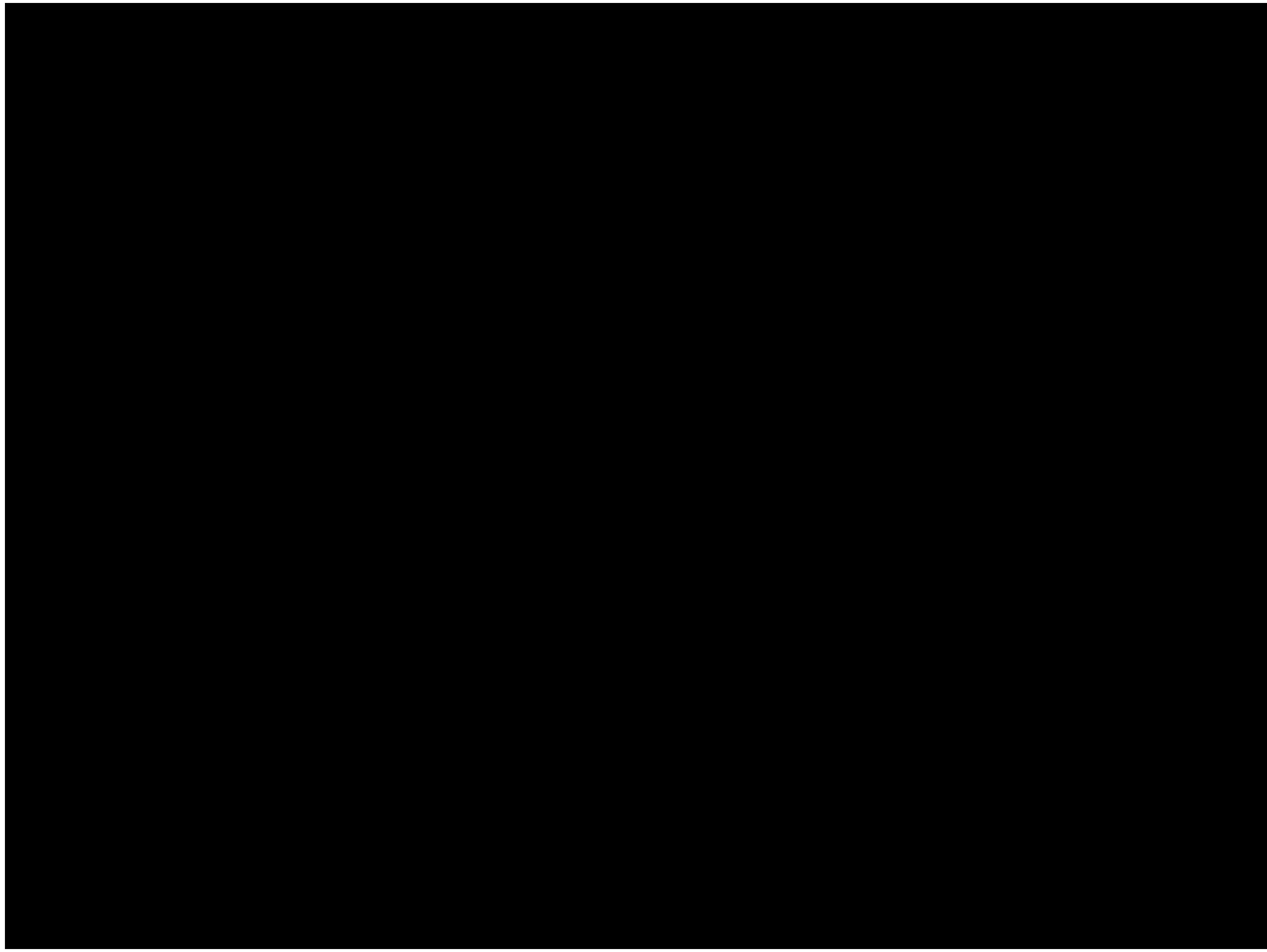
Gears

- Must be well designed to provide constant force transmission
  - Low wear/ low noise

- Back drivable: Can the wheel move the motor?

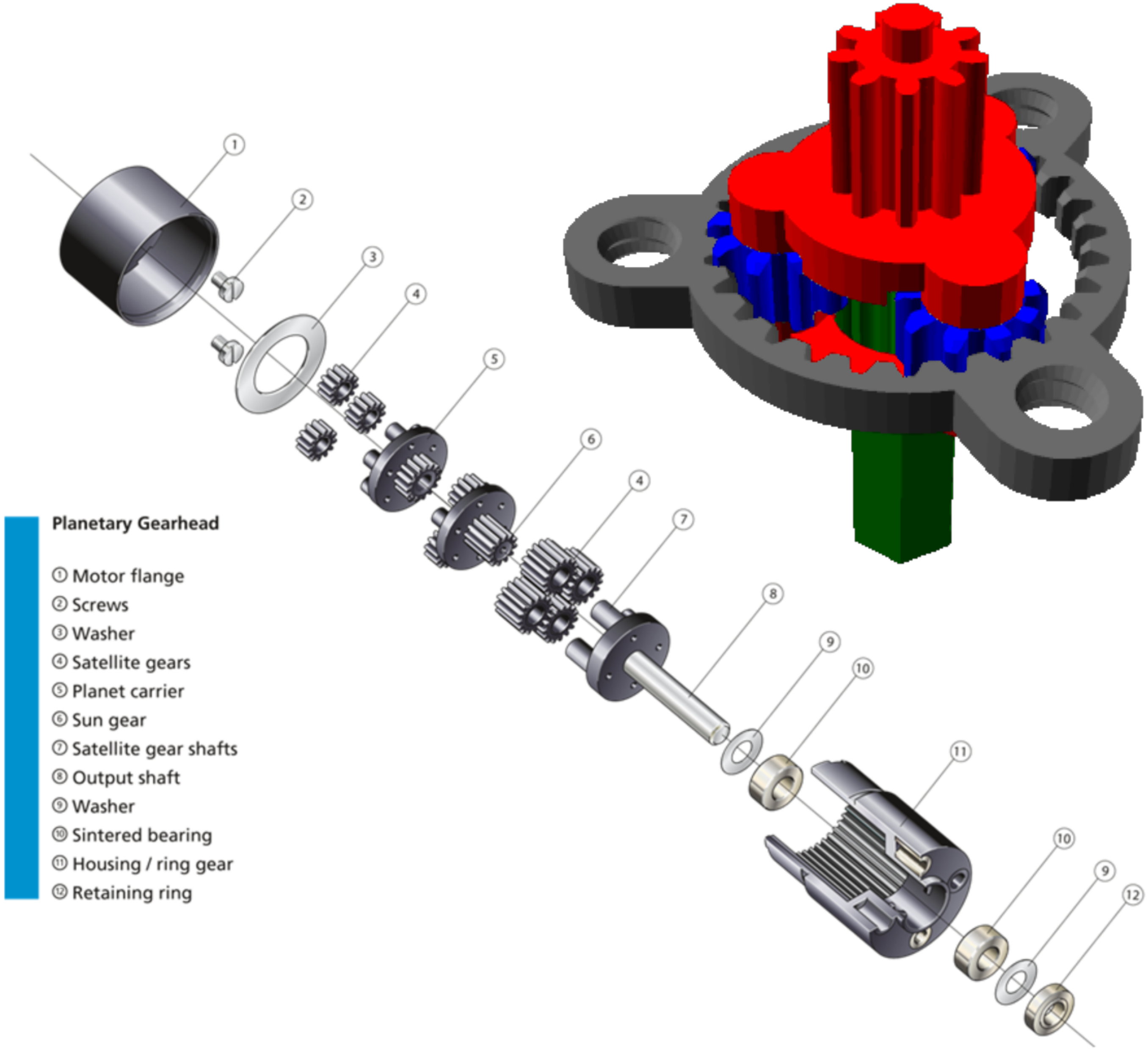
- Spur Gear reverses rotation direction!

- Backlash: when reversing direction: short moment of no force transmission => error in position estimate of wheel!



Planetary Gear

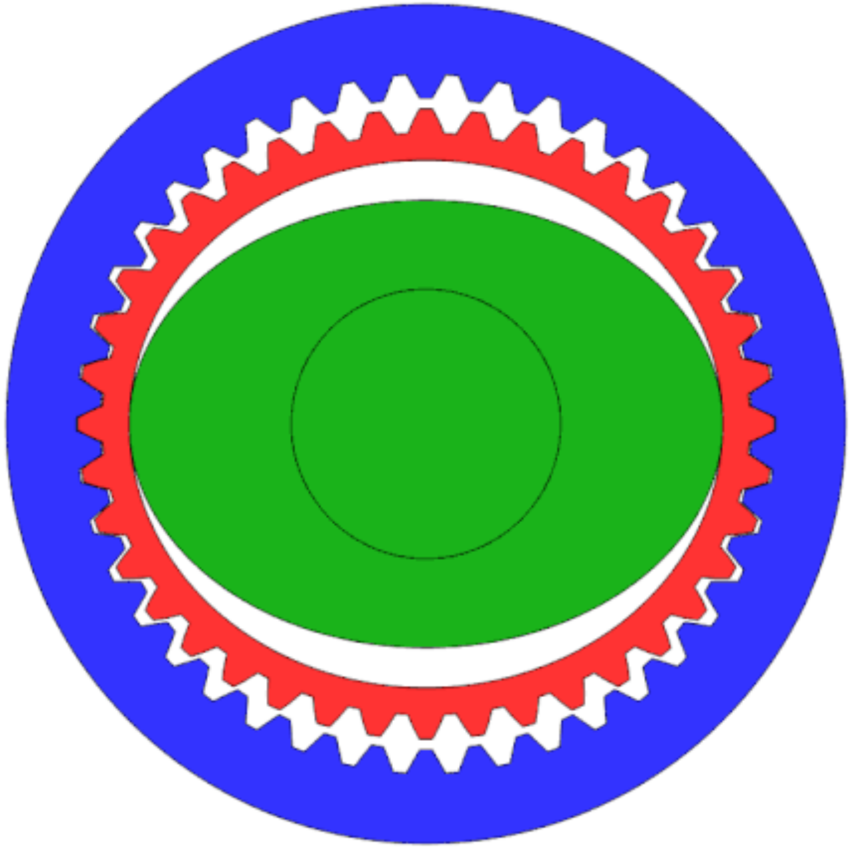
- Aka epicyclic gear train
- Quite common!
- Ratios: 3:1 … 1526:1
- Typical setup:
  - Sun (green) to motor
  - Carrier (red) output
  - Planets (blue): support
  - Ring (black): constraints the planets
  - => Ratio = 1:(1 + N_{Ring}/N_{Sun})



Harmonic Drive

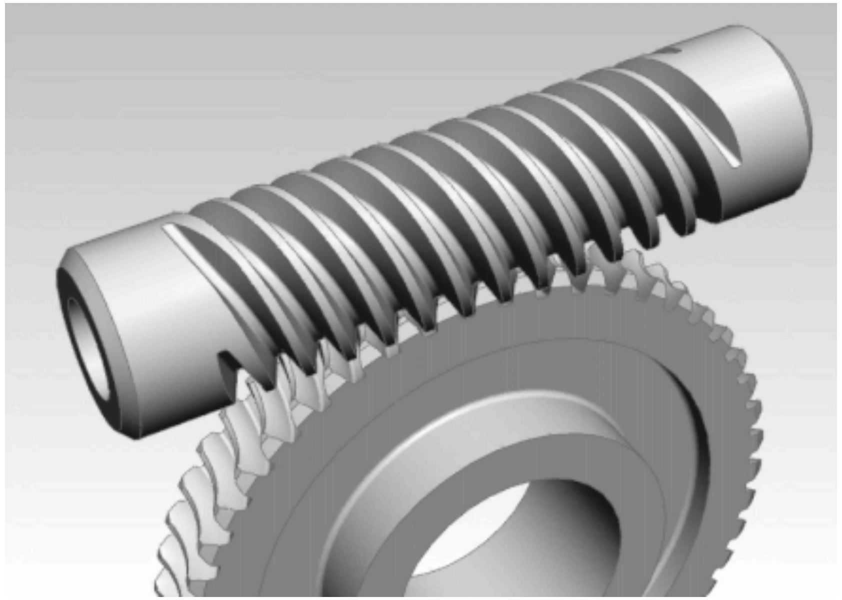
- High reduction in small volume (30:1 to 320:1)
- No backlash
- Light weight

- 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



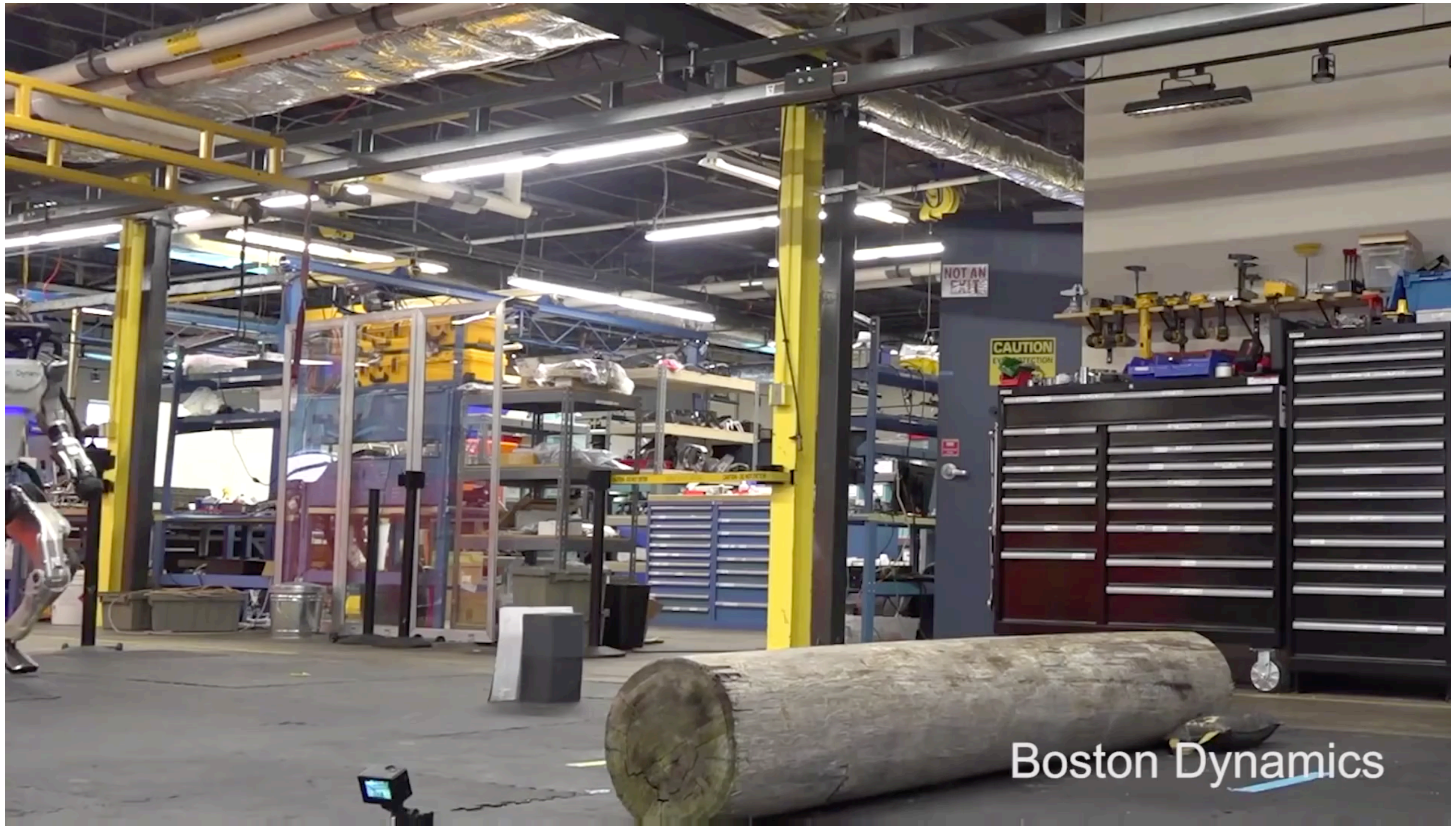
- Rack and pinion
  - linear drive
- Worm drive
  - Very high torque
  - Ratio: N_{Wheel} : 1
  - Locking (not back-drivable) gear)
- Bevel gear
  - Mainly to change direction

Summary: Mechatronics of Controlling a Wheel

1. Use Encoder and PID to control the speed of the motor
2. Send PWM signals to Motor Driver using dedicated circuits in CPU
3. Use Motor Driver to send high voltage & high current to motor
4. Use DC Brushed or Brushless motor to drive gear
5. Use Gear to get the required torque/ speed
6. Connect wheel to gear

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: …

Others

- Piezoelectric actuation
  - Small motions only
  - Very fast and precise
- Pneumatic actuator
  - Uses compressible gas
- Thermal-driven actuation