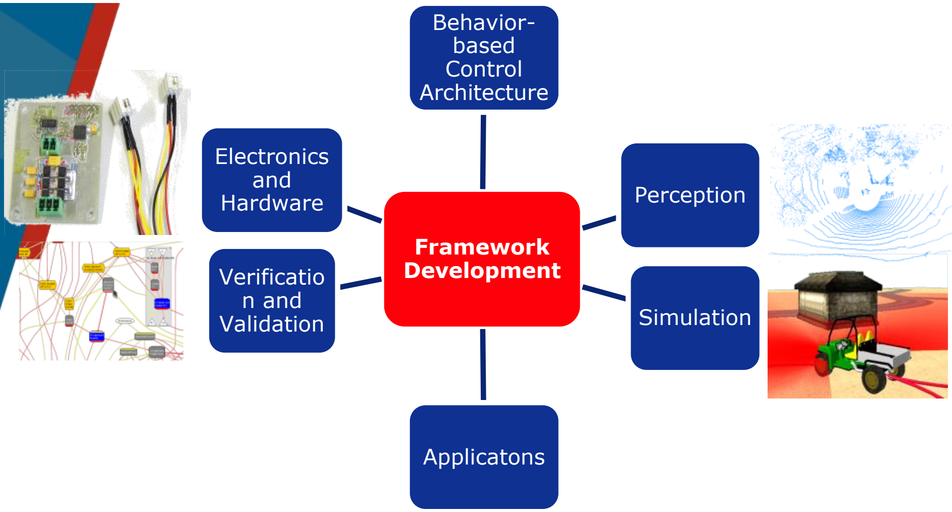
Foundations of Robotics – Subsystems and Components



Prof. Dr. Karsten Berns
Robotics Research Lab (RRLab)
Department of Computer Science
TU Kaiserslautern, Germany

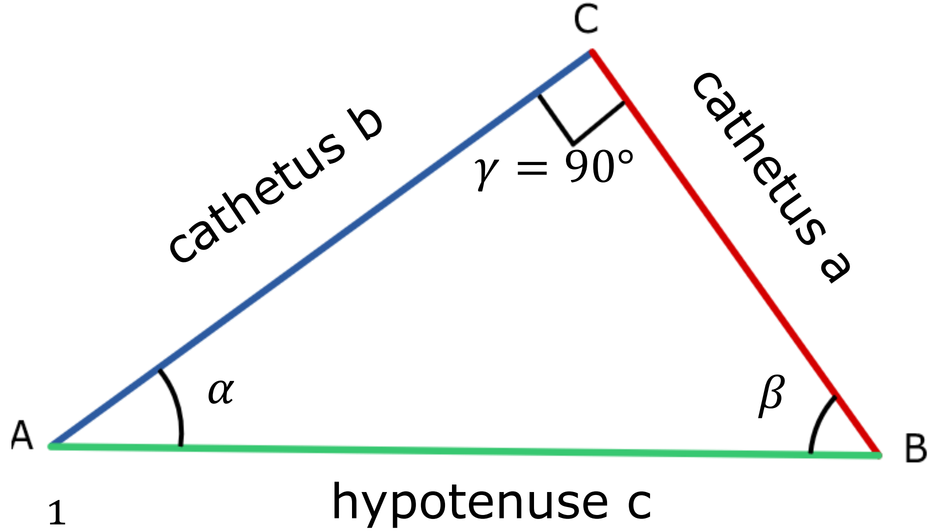
Content

- Mechanical components
  - Workspace
  - Joints
  - Basic configuration for a robot
  - Robotic Wrists
  - Actuators
- Open and closed loop control
- Sensors
- Hardware Architecture
- Simulation

Reminder: Trigonometric Functions

The trigonometric functions $\sin$, $\cos$, $\tan$, $\cot$ are defined as the ratios of corresponding sides in a right triangle (between $0°$ and $90°$).

- $\cos\alpha = \dfrac{\text{adjacent}}{\text{hypotenuse}} = \dfrac{b}{c}$
- $\sin\alpha = \dfrac{\textit{opposite}}{\text{hypotenuse}} = \dfrac{a}{c}$
- $\tan\alpha = \dfrac{\textit{opposite}}{\text{adjacent}} = \dfrac{a}{b} = \dfrac{a/c}{b/c} = \dfrac{\sin\alpha}{\cos\alpha}$
- $\cot\alpha = \dfrac{\text{adjacent}}{\textit{opposite}} = \dfrac{b}{a} = \dfrac{b/c}{a/c} = \dfrac{\cos\alpha}{\sin\alpha} = \dfrac{1}{\tan\alpha}$

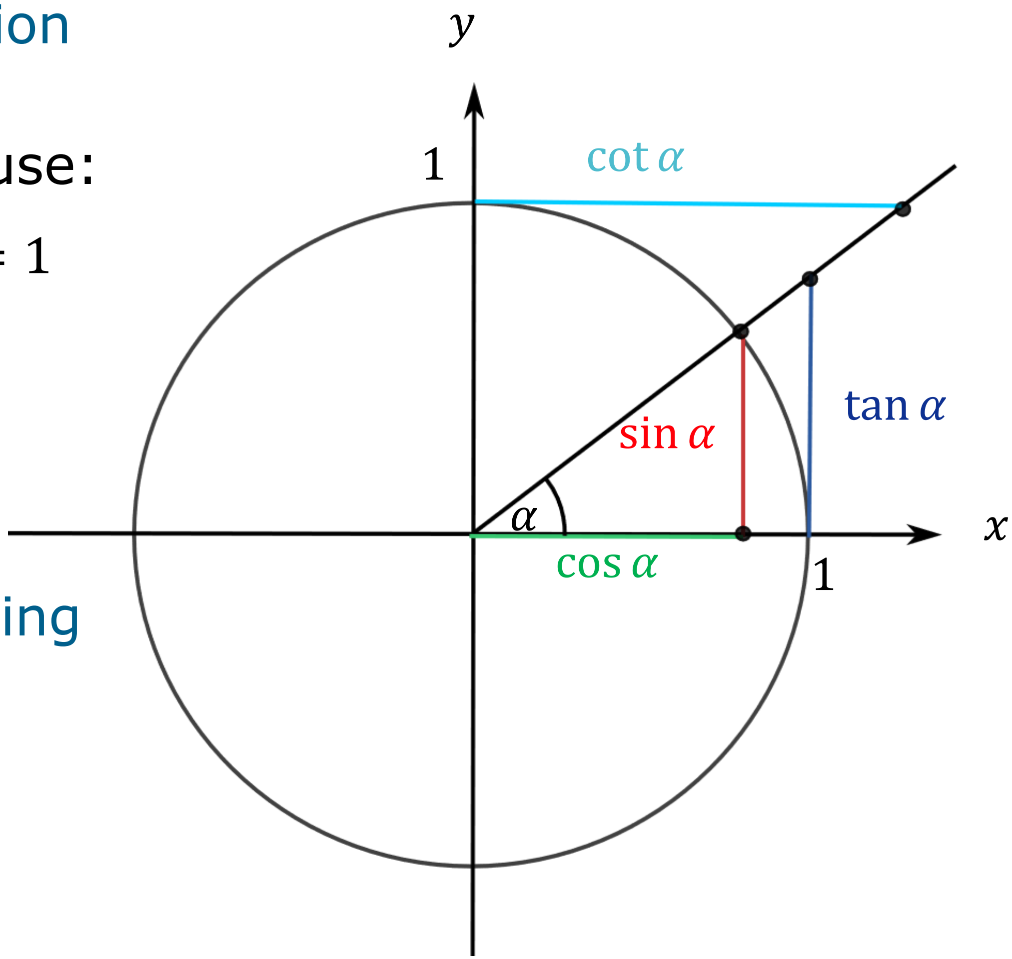


Reminder: Trigonometric Functions

- Unit circle for the definition of sin, cos, tan, cot
  - Length of the hypotenuse:

$\sqrt{x^2+y^2}=1$, or $x^2+y^2=1$

  - $\sin\alpha = y$
  - $\cos\alpha = x$
- Radian of an angle α: Length of the arc of the unit cicle corresponding to angle α



Reminder: Properties of Sine and Cosine

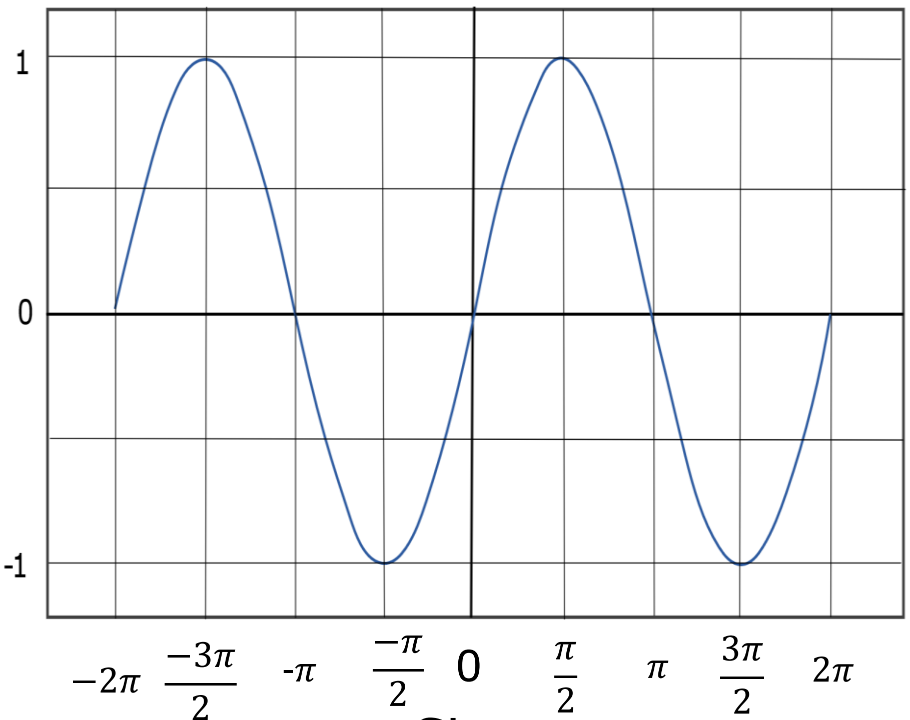
	$\sin \boldsymbol{x}$	$\cos \boldsymbol{x}$
Domain	$-\infty < x < \infty$	
Codomain	$-1 \leq \sin x \leq 1$	$-1 \leq \cos x \leq 1$
Period	2π	
Symmetry	Odd	Even
Roots	$x_k = k \cdot \pi$	$x_k = \frac{\pi}{2} + k \cdot \pi$
Maxima	$x_k = \frac{\pi}{2} + k \cdot 2\pi$	$x_k = k \cdot 2\pi$
Minima	$x_k = \frac{3\pi}{2} + \mathrm{k} \cdot 2\pi$	$x_k = \pi + k \cdot 2\pi$
Transformations	$\sin(90° - \alpha) = \cos \alpha$	$\cos(90° - \alpha) = \sin \alpha$

Reminder: Table of Values

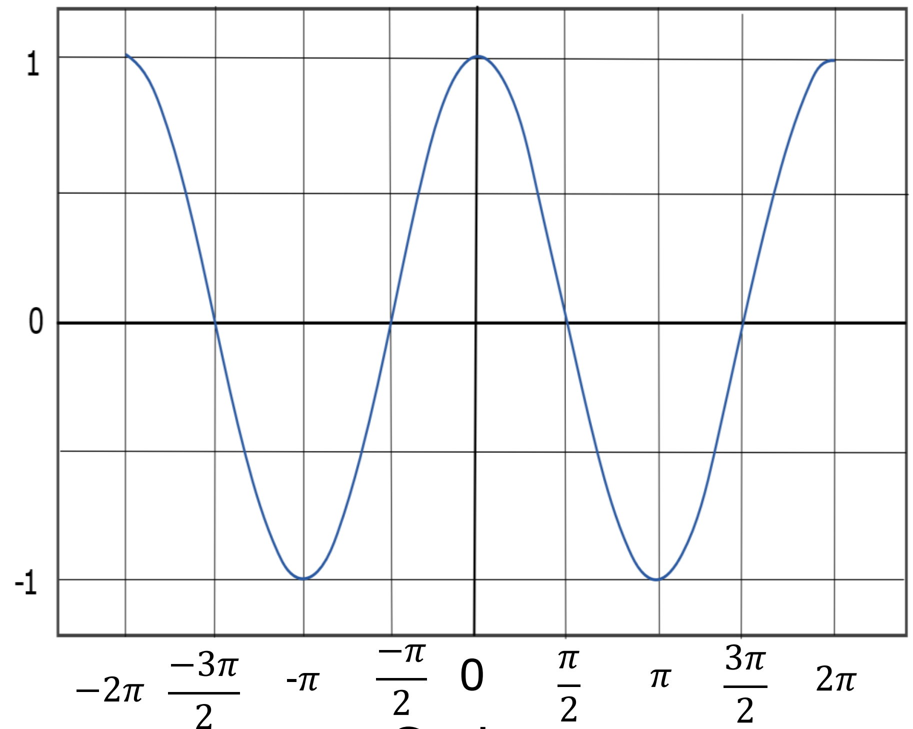
	Sine	Cosine
$0°$	$\frac{1}{2}\sqrt{0} = 0$	$\frac{1}{2}\sqrt{4} = 1$
$30°$	$\frac{1}{2}\sqrt{1} = \frac{1}{2}$	$\frac{1}{2}\sqrt{3}$
$45°$	$\frac{1}{2}\sqrt{2} = \frac{1}{\sqrt{2}}$	
$60°$	$\frac{1}{2}\sqrt{3}$	$\frac{1}{2}\sqrt{1} = \frac{1}{2}$
$90°$	$\frac{1}{2}\sqrt{4} = 1$	$\frac{1}{2}\sqrt{0} = 0$

Reminder: Additional Theorem and Graphs

- $\sin(x_1 \pm x_2) = \sin x_1 \cdot \cos x_2 \pm \cos x_1 \cdot \sin x_2$
- $\cos(x_1 \pm x_2) = \cos x_1 \cdot \cos x_2 \mp \sin x_1 \cdot \sin x_2$
- $\tan(x_1 \pm x_2) = \frac{\tan x_1 \pm \tan x_2}{1 \mp \tan x_1 \cdot \tan x_2}$



Sine



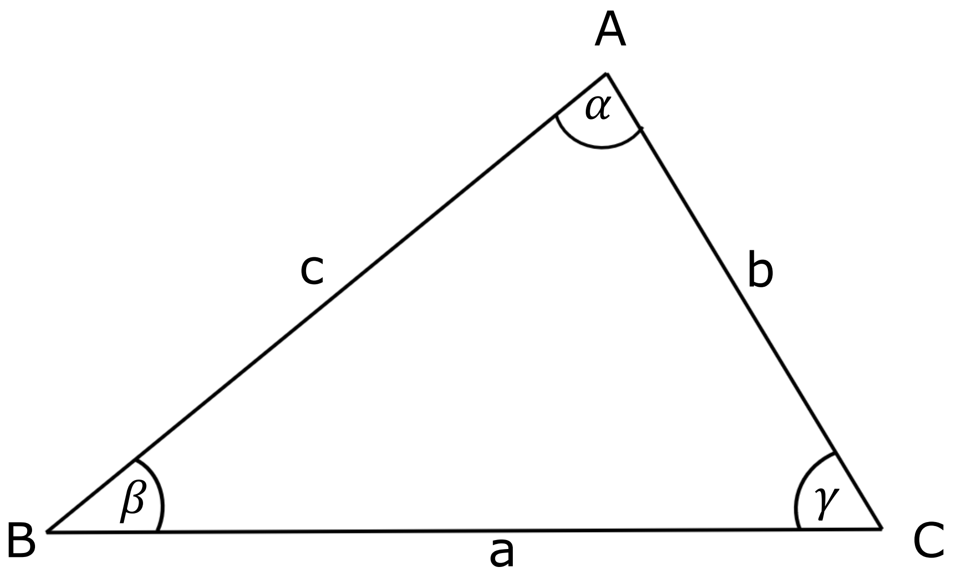
Cosine

Cosine/Sine Rule

β

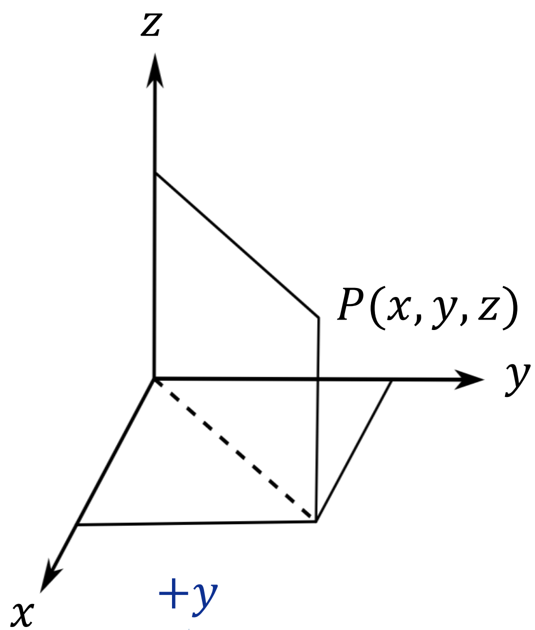
- Cosine Rule:
  - $a^2 = b^2 + c^2 - 2bc\cos(\alpha)$
  - $b^2 = a^2 + c^2 - 2ac\cos(\beta)$
  - $c^2 = a^2 + b^2 - 2ab\cos(\gamma)$

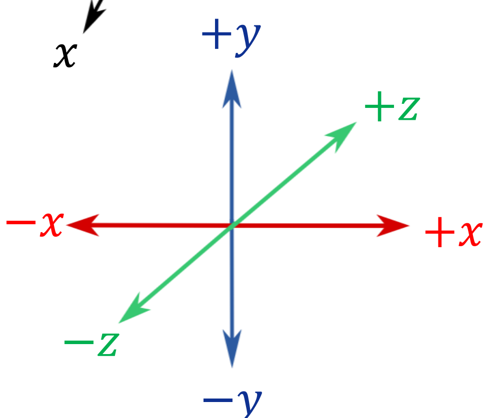
- Sine Rule: β
  - $\frac{\sin\alpha}{a} = \frac{\sin(\beta)}{b} = \frac{\sin(\gamma)}{c}$



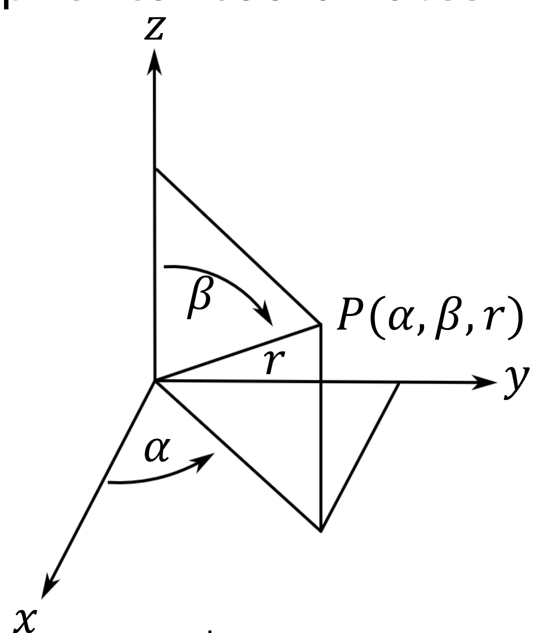
Reminder: 3D Coordinate Systems

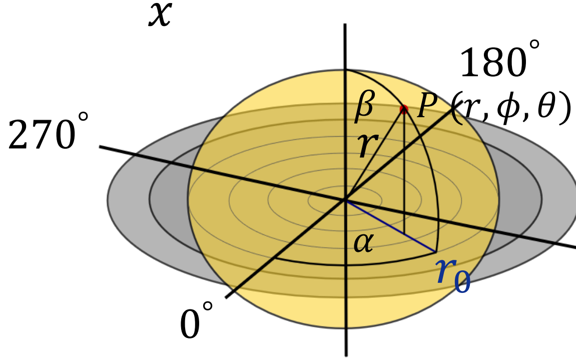
Cartesian coordinates



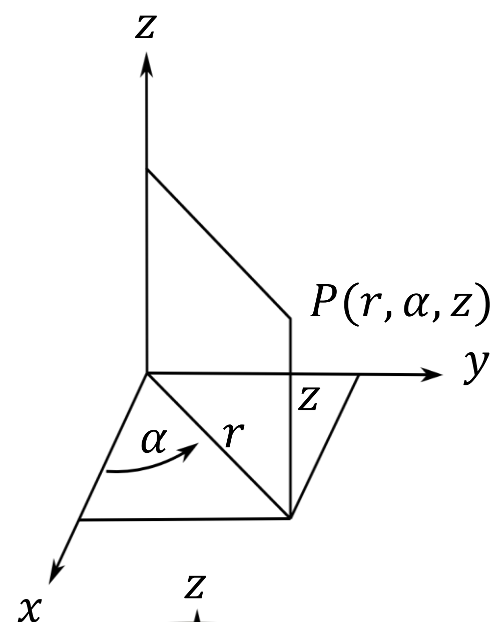


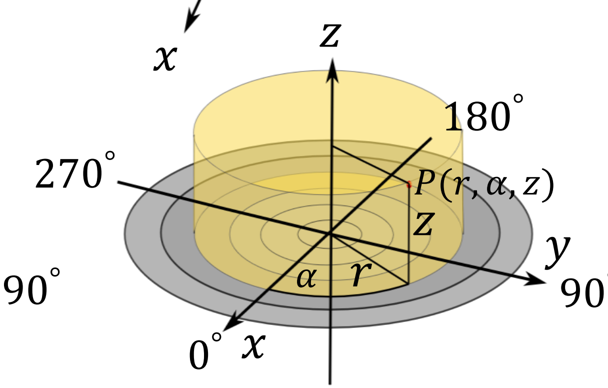
Spherical coordinates





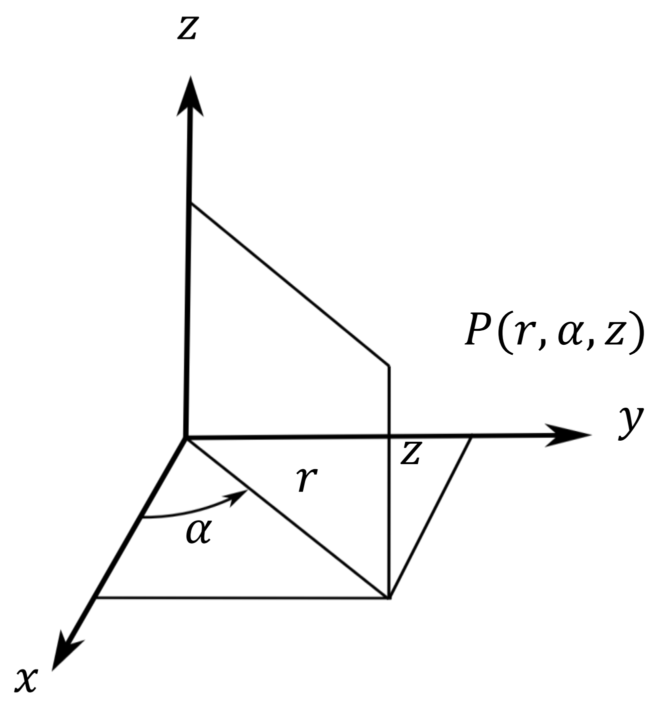
Cylindrical coordinates





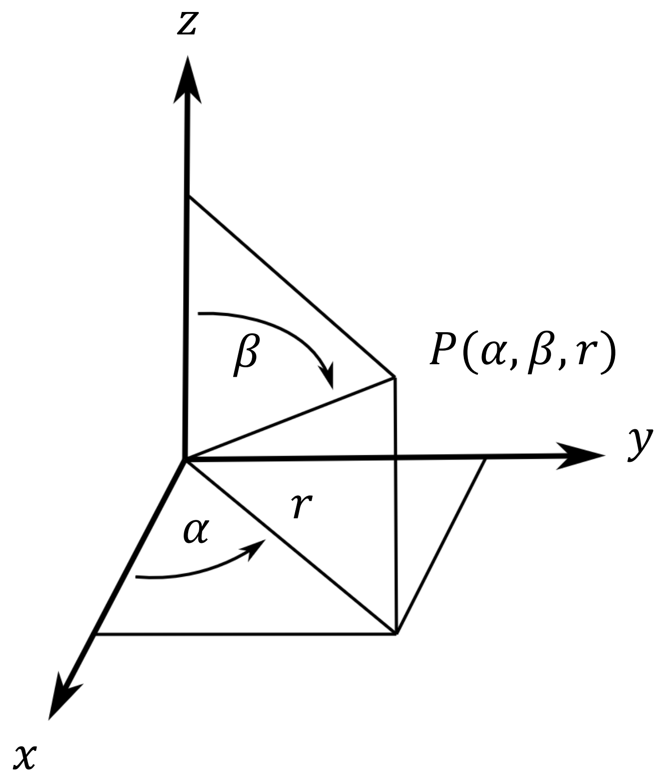
Transformation of Coordinate Systems

- Cartesian coordinates → Cylindrical coordinates
  - $(x, y, z) \rightarrow (r, \alpha, z)$
  - $r = \sqrt{x^2 + y^2}$
  - $\tan \alpha = \frac{y}{x}$
  - $z = z$
- Cylindrical coordinates → Cartesian coordinates
  - $(r, \alpha, z) \rightarrow (y, x, z)$
  - $x = r \cdot \cos \alpha$
  - $y = r \cdot \sin \alpha$
  - $z = z$



Transformation of Coordinate Systems

- Cartesian coordinates → Spherical coordinates
  - $(x, y, z) \rightarrow (r, \alpha, \beta)$
  - $r = \sqrt{x^2 + y^2 + z^2}$
  - $\cos\beta = \frac{z}{r}$
  - $\tan\alpha = \frac{y}{x}$
- Spherical coordinates → Cartesian coordinates
  - $(r, \alpha, \beta) \rightarrow (x, y, z)$
  - $x = r \cdot \sin\beta \cdot \cos\alpha$
  - $y = r \cdot \sin\beta \cdot \sin\alpha$
  - $z = r \cdot \cos\beta$



Definition: Workspace

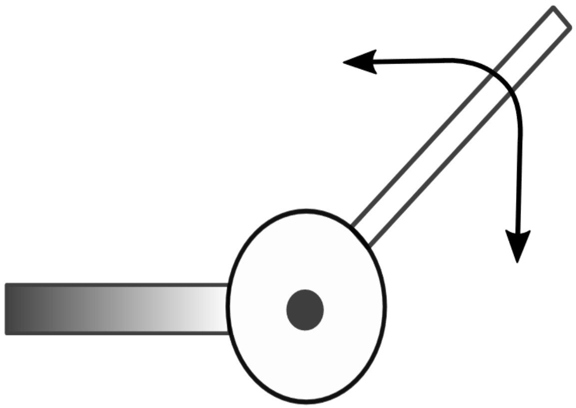
- Workspace consists of all points, which are reachable by the robot hand
  - At least 3 DOF (degree of freedom) necessary for a 3-D space
  - At least 3 basic joints necessary for a 3-D space

- Basic shape of workspace consists of all points, which are reachable by the robot hand without considering any restrictions by the joints or obstacles

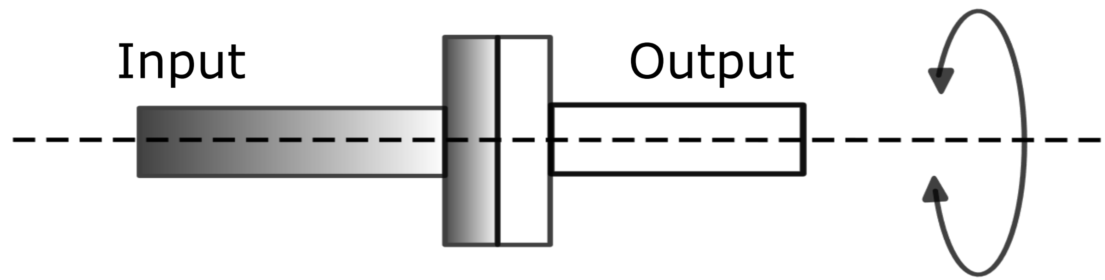
Relation between Degrees of Freedom and Joints

- Number of possible independent movements of an object in relation to a fixed coordinate system
- Pose of an object that can move freely in space is defined by
  - Position (3 values)
  - Orientation (3 values)
- Joints necessary to achieve the degree of freedom (DOF)
- Rotary joints are required for orientation, as liner joints would not change the orientation of the wrist
- Active (actuated) and passive DOF

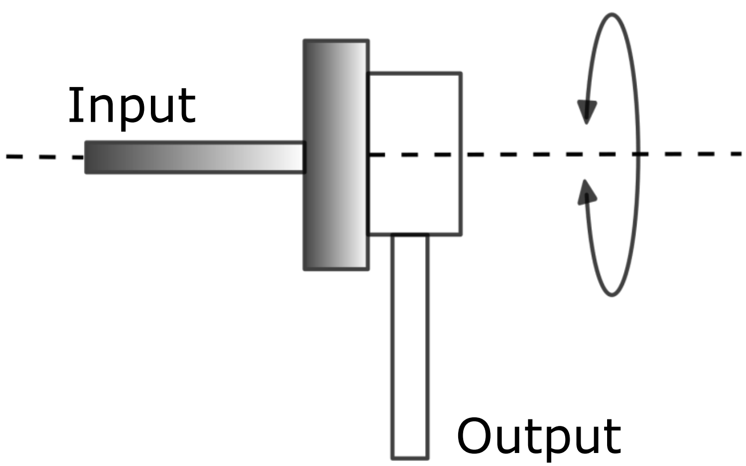
The 4 Basic Joint Types of Robotic Systems



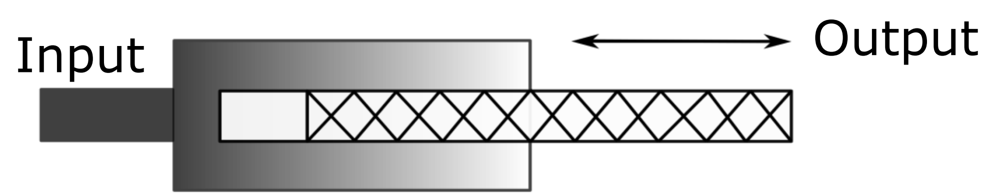
Rotational Joint (R)



Torsion Joint (T)



Revolute Joint (V)



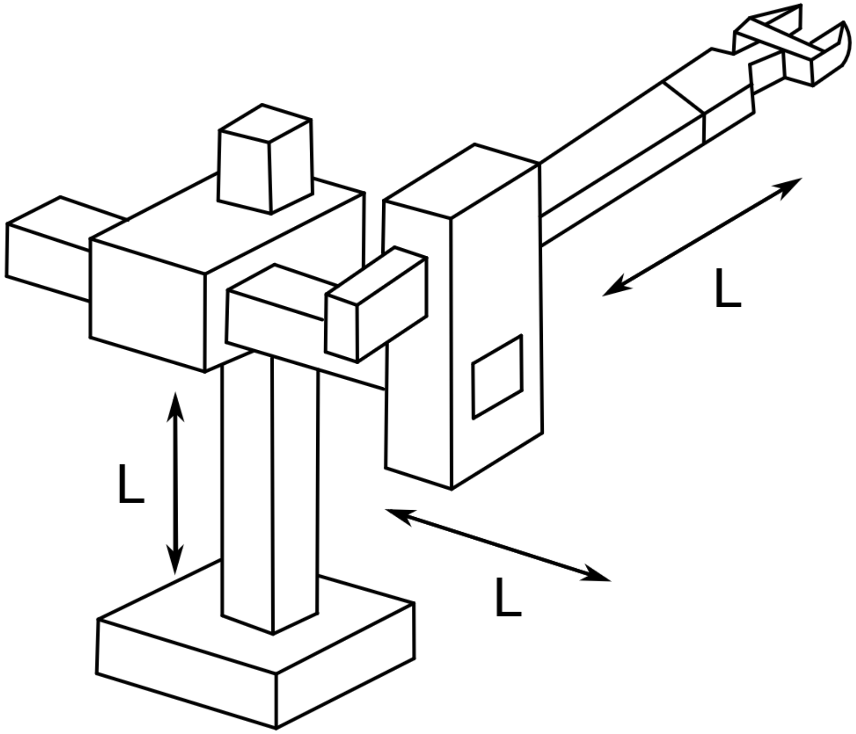
Linear Joint (L)

see also [Siegert, Bocionek 96]

Basic Robot Types

- Cartesian Robot (LLL)

  Basic shape of operational space: Cuboid

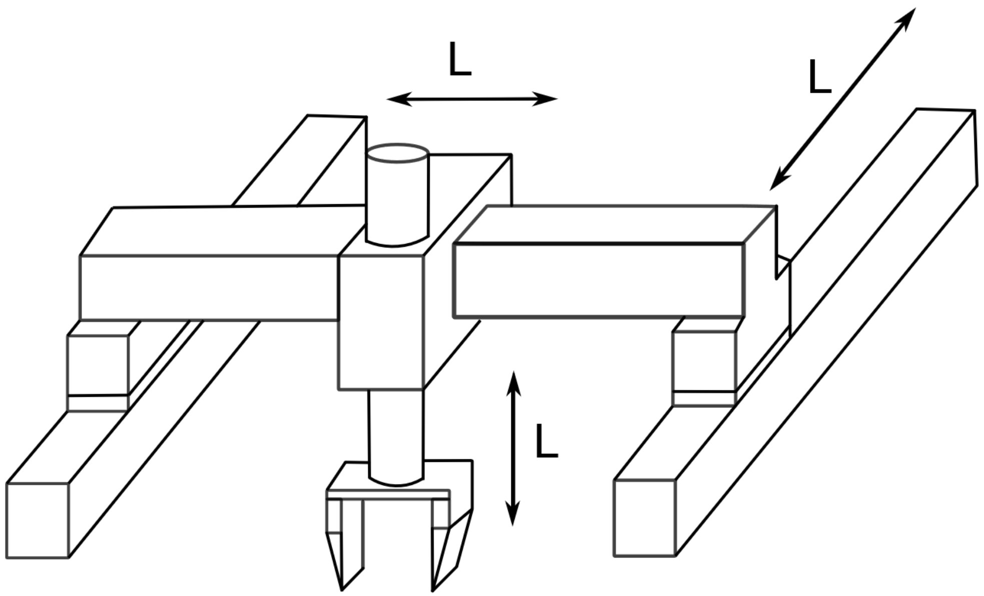


[Siegert, Bocionek96]

Basic Robot Types

- Cartesian Robot (LLL)

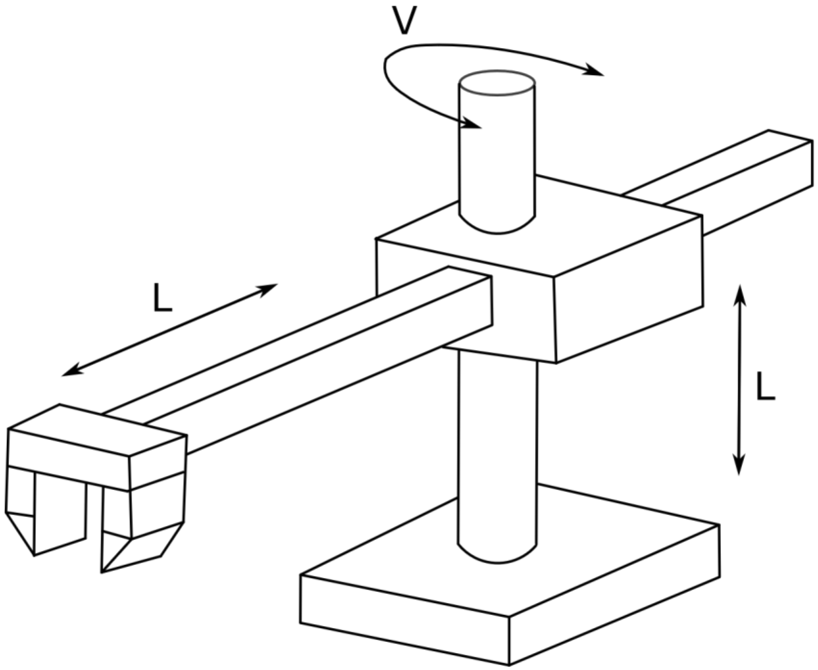
  Basic shape of operational space: Cuboid



[*Siegert, Bocionek96*]

Basic Robot Types

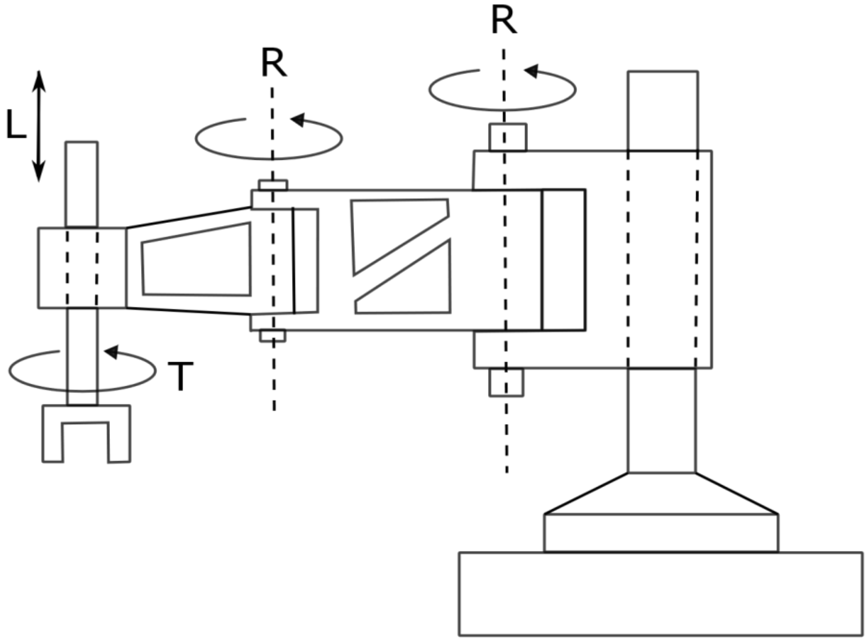
- Robot Arm (LVL)
  - Basic shape of operational space: Cylinder
  - Other possibilities: TLL, LTL



[Siegert, Bocionek96]

Basic Robot Types

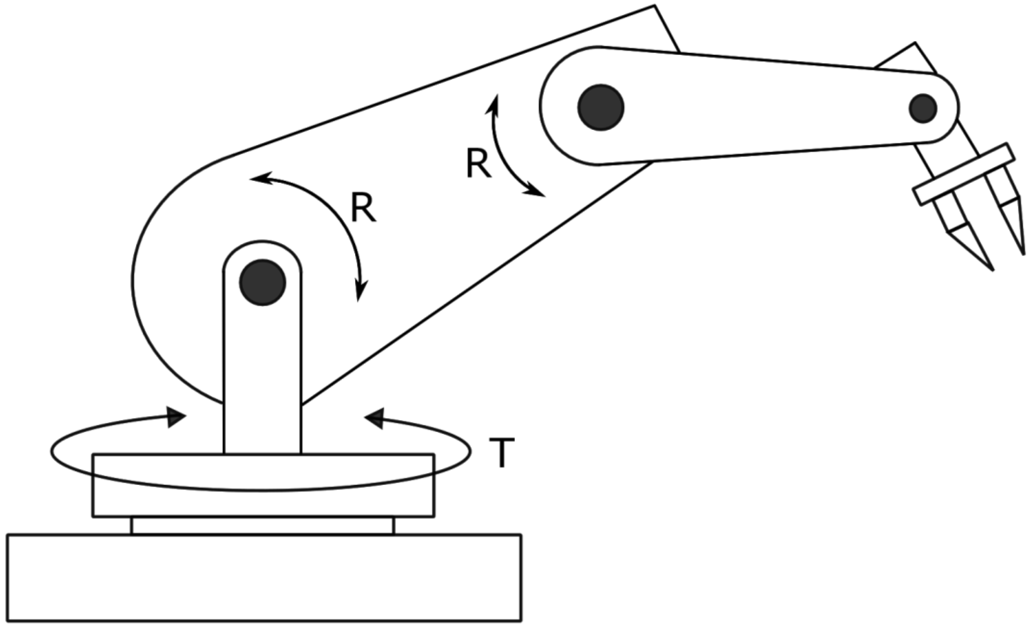
- SCARA-Robot (RRLT)
  - **S**elective **c**ompliance **a**ssembly **r**obot **a**rm
  - Basic shape of operational space: Cylinder



[Siegert, Bocionek96]

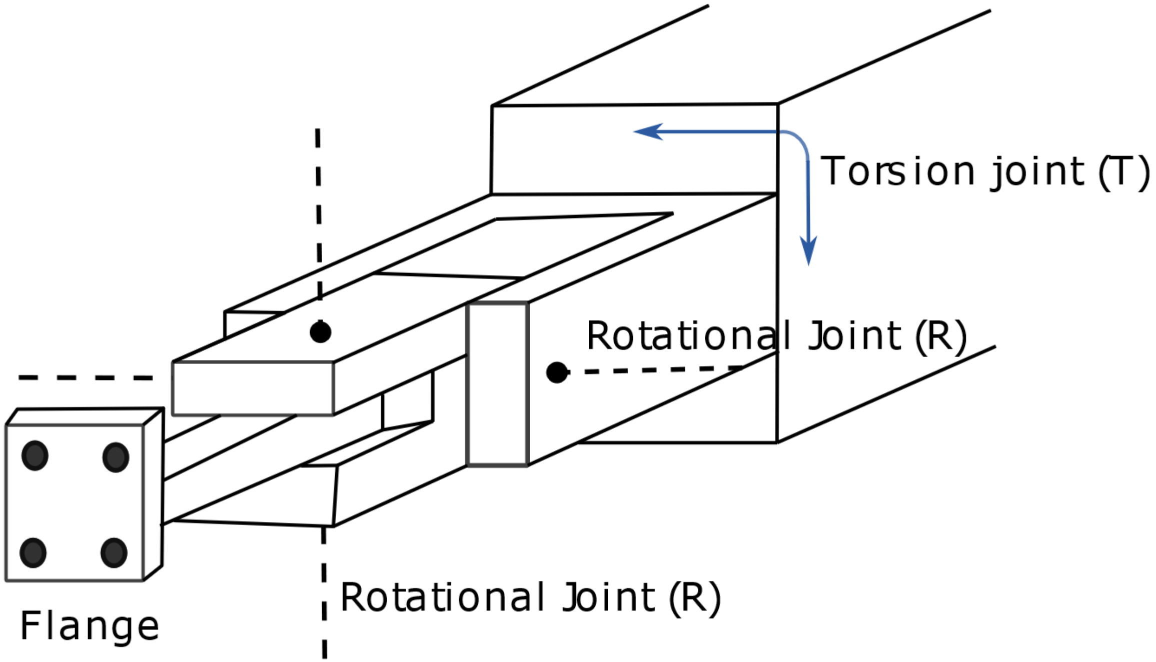
Basic Robot Types

- Universal Robot Arm (TRR)
  - Basic shape of operational space: Sphere
  - Other possibilities: VVR



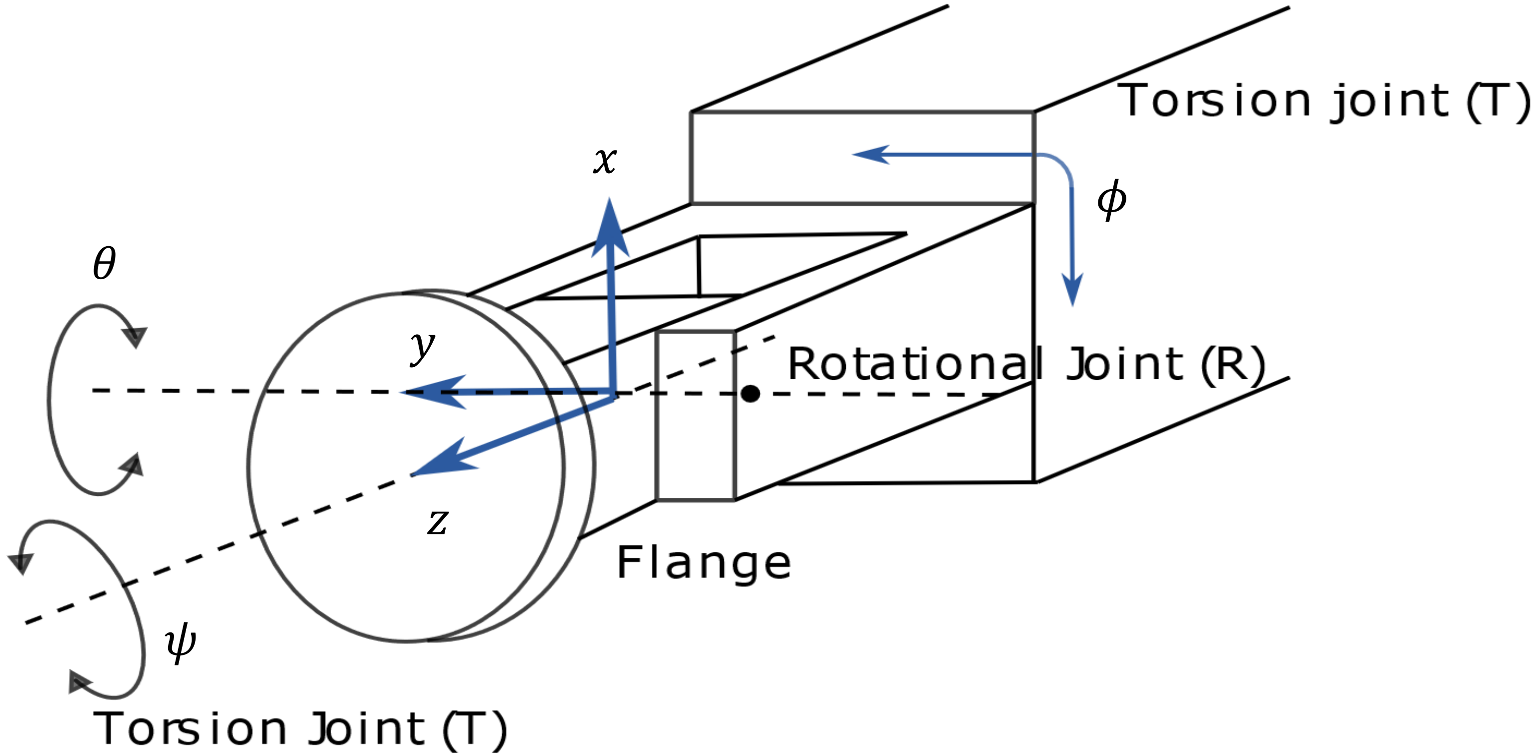
Robotic Wrists:

- Basic Shape of Wrist (TRR)



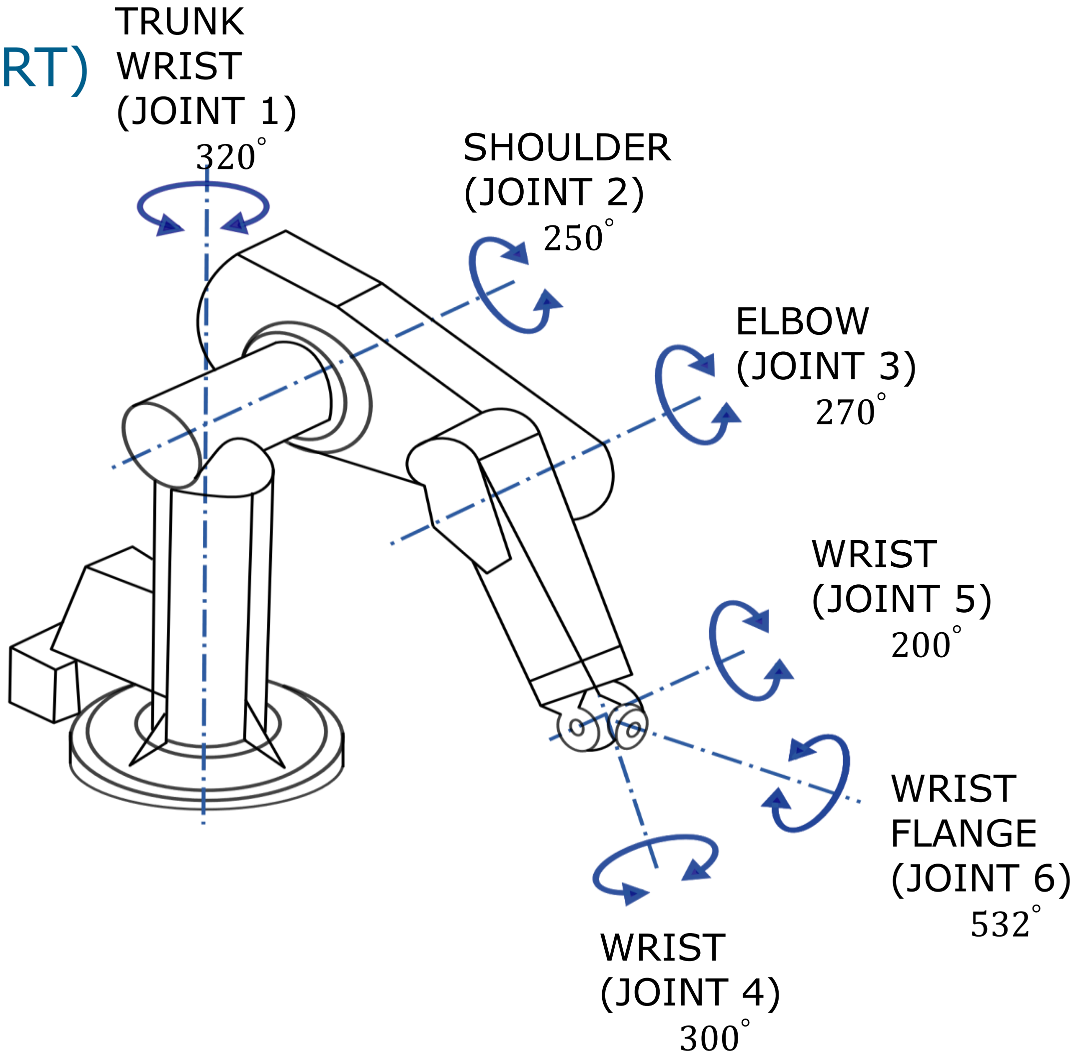
Robotic Wrists

- Basic Shape of Wrist (TRT)

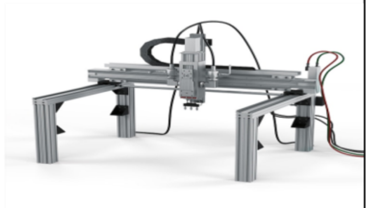
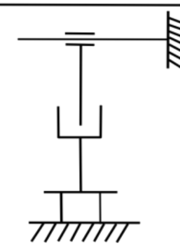
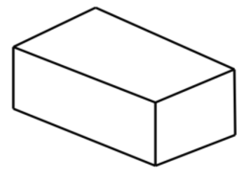
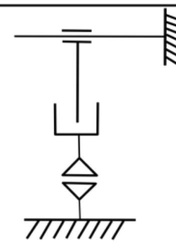
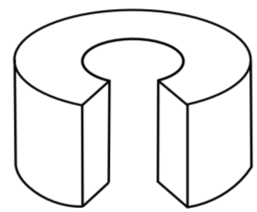
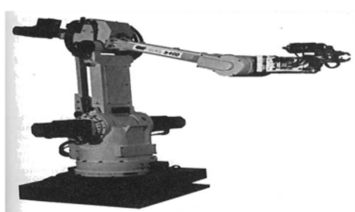
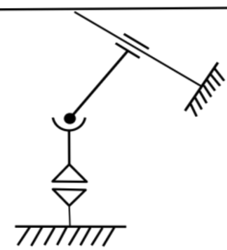


Degrees of Freedom and Joints

- Puma Robot (TVRRRT)

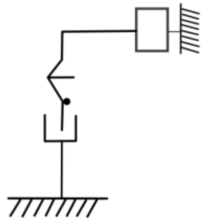
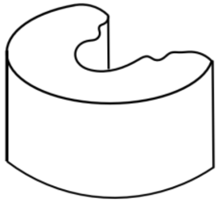
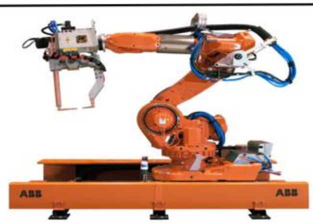
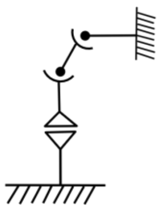
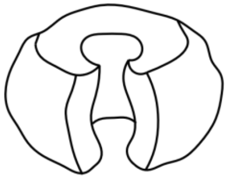
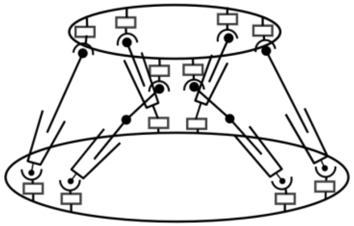


Robot Kinematics

Robot	Axes	
Photo	Kinematic Stucture	Workspace
 Cartesian Robot		
 Cylindrical Robot		
 Spherical Robot		

[*World Robotics 2003*]

Robot Kinematics

Robot	Axes	
Photo	Kinematic Structure	Workspace
 Scara Robot		
 Articulated Robot		
 Parallel Robot		

[*World Robotics 2003*]

Actuators for the Realization of active Joints

- Fluid actuators (pneumatic/hydraulic)
  - Linear
  - Rotational
  - Muscle principle
- Electric actuators
  - Linear
  - Rotational
  - DC motor (brushless, brushed)
  - AC motor
  - Stepper motors
  - Servo motor

Pneumatic Actuators

- Used energy and control
  - Compressed air, no gear box
- Pros
  - Cheap, easy setup, fast reaction times
  - Usable in rough environments
- Cons
  - Noisy
  - Difficult control
  - Mostly just point to point
  - Bad accuracy
- Usage
  - Small robots with fast cycles and low force

Hydraulic Actuators

- Used energy and control
  - Oil pressure, controllable valves
- Pros
  - Very high forces
  - Average speed
- Cons
  - Noise
  - Leakage of oil
  - Additional space for hydraulics are needed
  - Slow and inaccurate due to viscosity of oil
- Usage
  - Big robots

Electric Actuators

- Used energy and control
  - Electric energy, current (voltage) control
- Pros
  - Small
  - Easy to control
  - High precision
- Cons
  - Small forces
- Usage
  - Small robots for tasks which require high accuracy

Kinematics Module

Kinematics module (control module) of a robot allows the joints to be positioned.

The basic tasks:

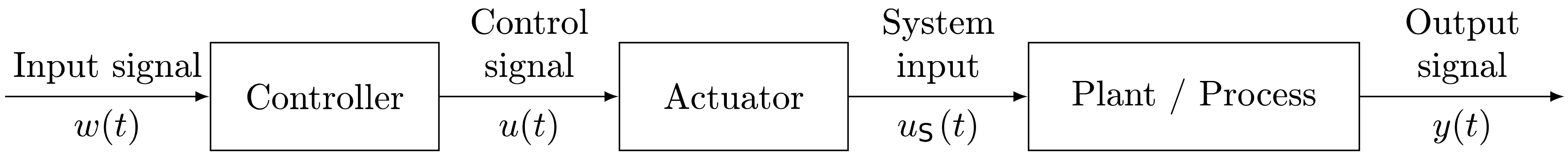
- Forward calculation:
  User directly specifies the joint coordinates (joint angle)
- Backward calculation:
  User specifies the pose of the effector that the robot should approach or move through with a certain speed and acceleration
- Teaching path or path points:
  User manually controls the robot arm in a sequence of target positions.

Requirements for Joint Control

- The following requirements are often used:
  - Arm movements as fast as possible
  - Movement along the specified path without swinging, in particular no overshooting in tight curves or at the target position
  - Adaptation to loads in hand
  - Holding the arm with the load at the target position (no drifting due to the weight of the load)

Open Loop Control

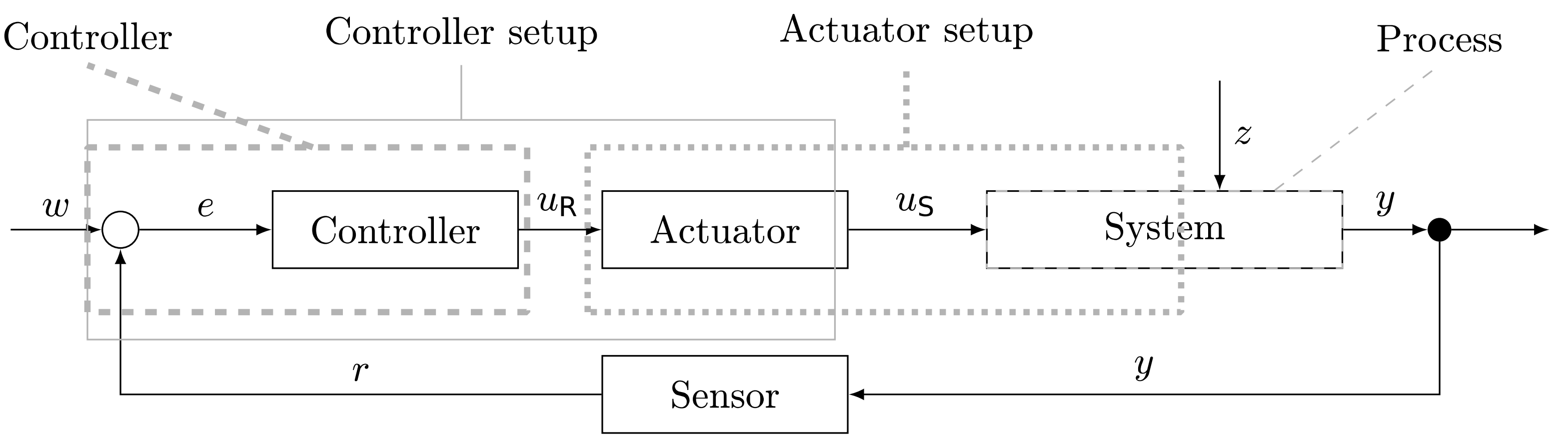
- Control variable are set without any knowledge of the current state
- No Feedback
  - No correction of input variables
  - Noise leads to deviations
  - Needs a exact model of the process
  - Result of the control is unknown



Differences between the goal and the current state can not be detected. Damage to the system is possible.

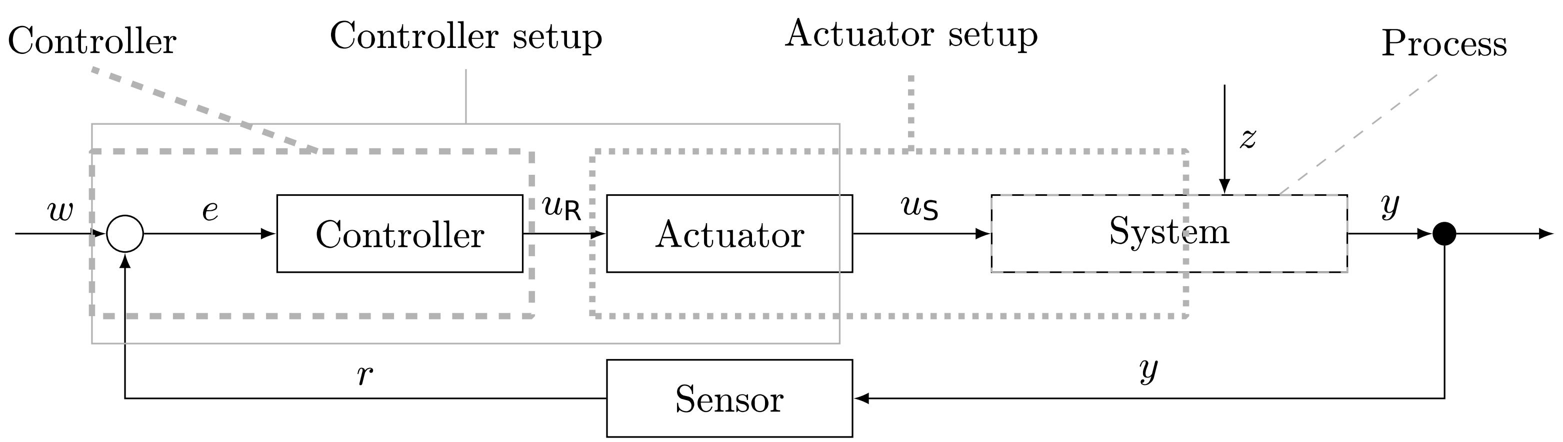
Closed Loop Control

- Observes process via feedback (Closed Loop)
- Can react to Noise
- Process: Part of the system which needs to be controlled
- Reference variable w: value which the output should trace
- Control difference e: Difference between w and process variable r



Closed Loop Control

- Comparator: Calculates control difference e
- Controller: Calculates from control difference signal e the output variable, which leads to a possible tracing
- Actuator: Manipulates energy flow or mass flow. Output is the system input u_S.
- Controller variable/Manipulated variable u_R: Input to actuator

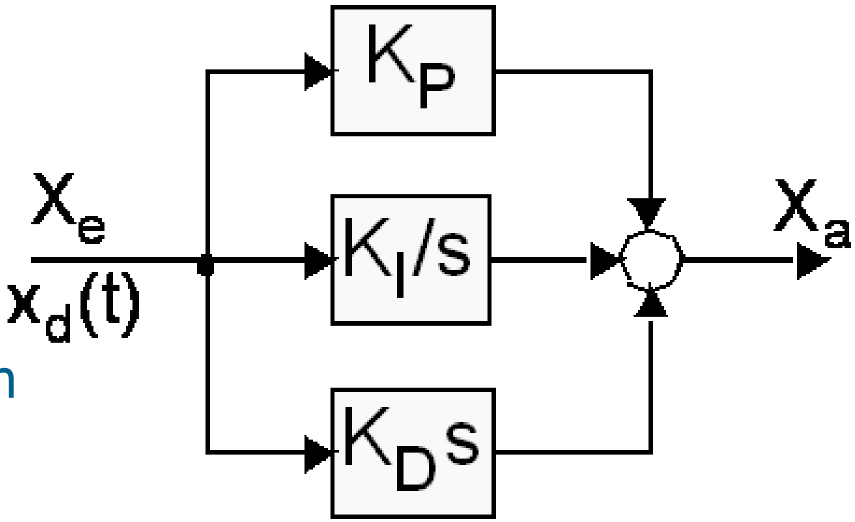


Requirements of a Closed Loop Control

Selection of controller type and calculation
of its parameters for …

- Steady state accuracy: control difference vanishes for $t \rightarrow \infty$
- Speed: actual value follows desired value as good/fast as possible
- Stability: no instability of control system due to feedback of system output
- Robustness: small changes of parameters of the control plant do not change the properties of the control system
$\rightarrow$ approximations for calculation of control parameters are feasible

PID-Controller (often used)



- Stationary accurate
- Fast
- Tends to oscillated when the reference value is reached
- Oscillation is damped by D-term
- Frequency domain

$$G_{PID}(s) = K_P + \frac{K_I}{s} + K_D \cdot s$$

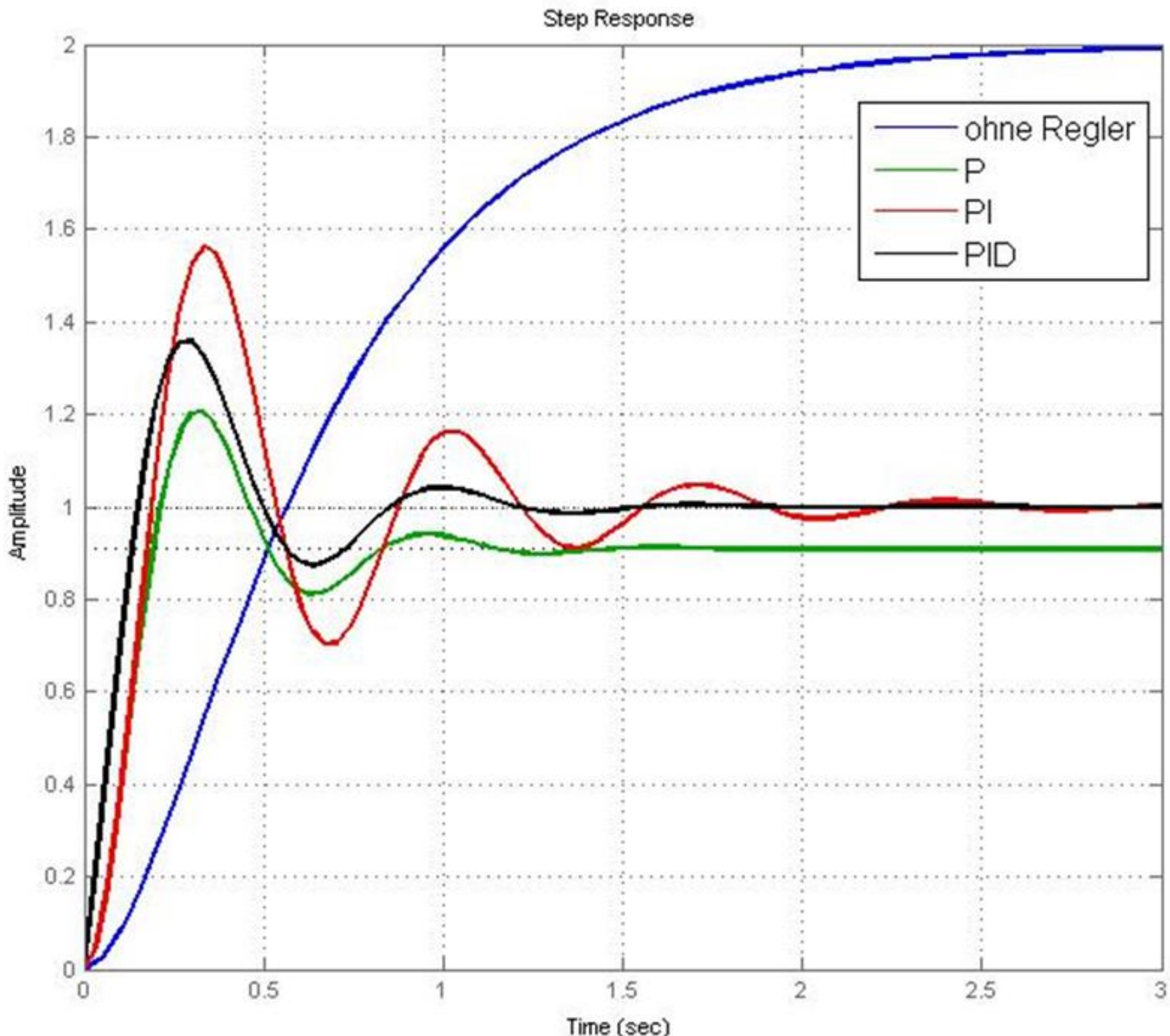
$$\Rightarrow X_a(s) = K_P X_e(s) + \frac{K_I}{s} X_e(s) + K_D \cdot s X_e(s)$$

$$= K_P \left(1 + \frac{1}{T_N s} + T_D s\right) X_e(s)$$

- Time domain

$$y(t) = K_P x_d(t) + K_I \int_0^t x_d(t)\, dt + K_D \cdot \dot{x}_d(t)$$

PID-Controller (Example)



Sensor Classification: Proprioceptive

Acquisition of internal states of a robot/machine e. g.: Joint position, joint velocity, joint acceleration, orientation

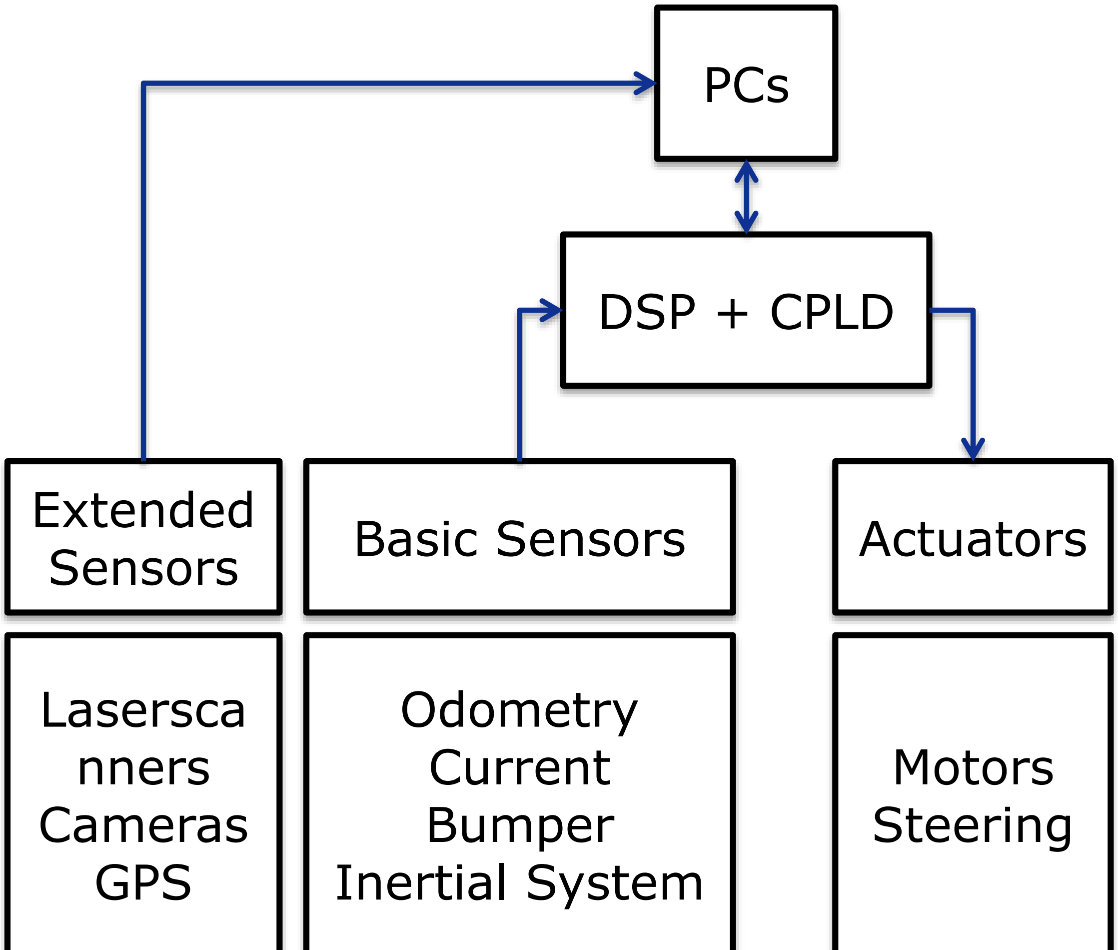
- Position
  - Potentiometer
  - Optical encoder
  - Differential transformer transducer
  - Magnetic-inductive encoder
- Velocity
  - Speed generator
  - Optical encoder
- Acceleration
  - Si-sensor
  - Piezo-electric sensor
- Orientation
  - Gyroscope
  - Geomagnetic sensor

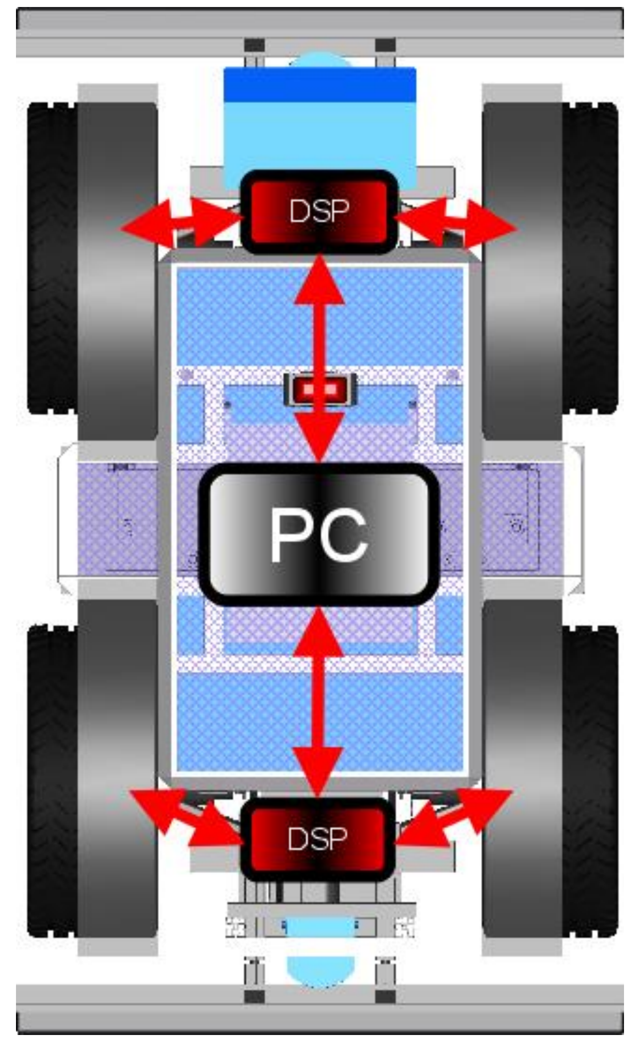
Sensor Classification: Exteroceptive

Acquisition of external states (⇒ environment) e. g.: obstacle distance, object identification, object position

- “Feel”
  - Artificial skin
  - Sliding sensors
  - Force-torque-sensors
- Approach
  - Inductive, capacitive sensors
  - Optical sensors
  - Acoustic sensors

Electronics and Computing Architecture





Simulation

Why simulation?

- Control and perception algorithms can be developed before the robot exists
- Safe testing of algorithms
- Tests in simulation are faster (several tests in parallel on a computer cluster)
- Test can be repeated under absolutely the same conditions
- Test environments can be exchanged
- Different light and weather conditions can be generated

Simulation

Problems with simulation?

- High effort for a realistic simulation of sensor systems (real-time requirements often not fulfilled)
- Physical Engine are weak in the modelling of dynamics
- High effort for the development of simulators
- Adequate interfaces to the control system must be implemented
- Still differences between simulation and real robots in its operational environment

Simulations using Unreal Engine

Literature

- [Siegert, Bocionek 96] Siegert, H.-J. and Bocionek, S. (1996) Robotik: Programmierung intelligenter Roboter. Springer Verlag
- [World Robotics 2003] International Federation of Robotics, United Nation, New York and Geneva, 2003

Coming up next ...

Spatial Kinematics

–

Foundations I