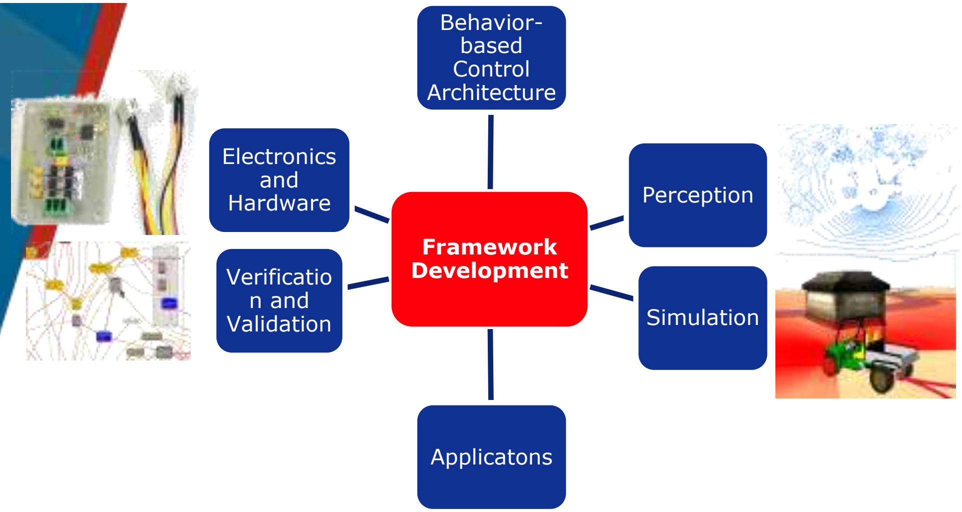
# Foundations of Robotics – Subsystems and Components





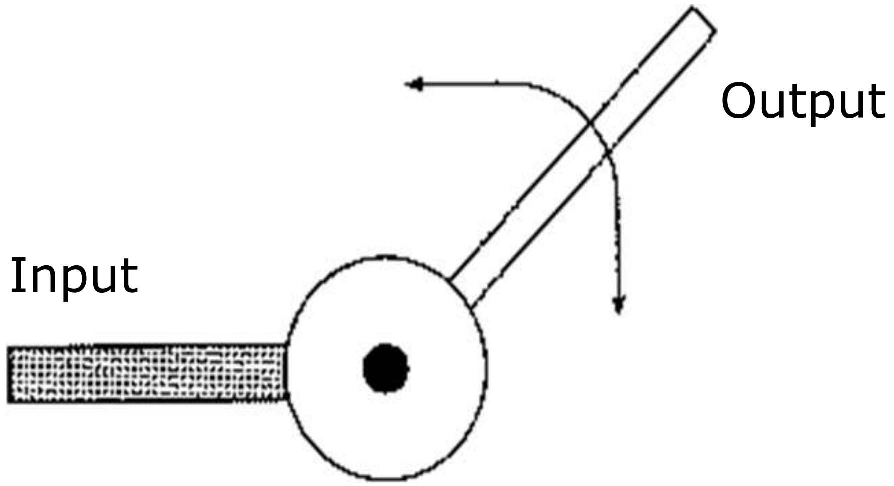
**Prof. Dr. Karsten Berns**
Robotics Research Lab (RRLab)
Department of Computer Science
TU Kaiserslautern, Germany

# Content

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

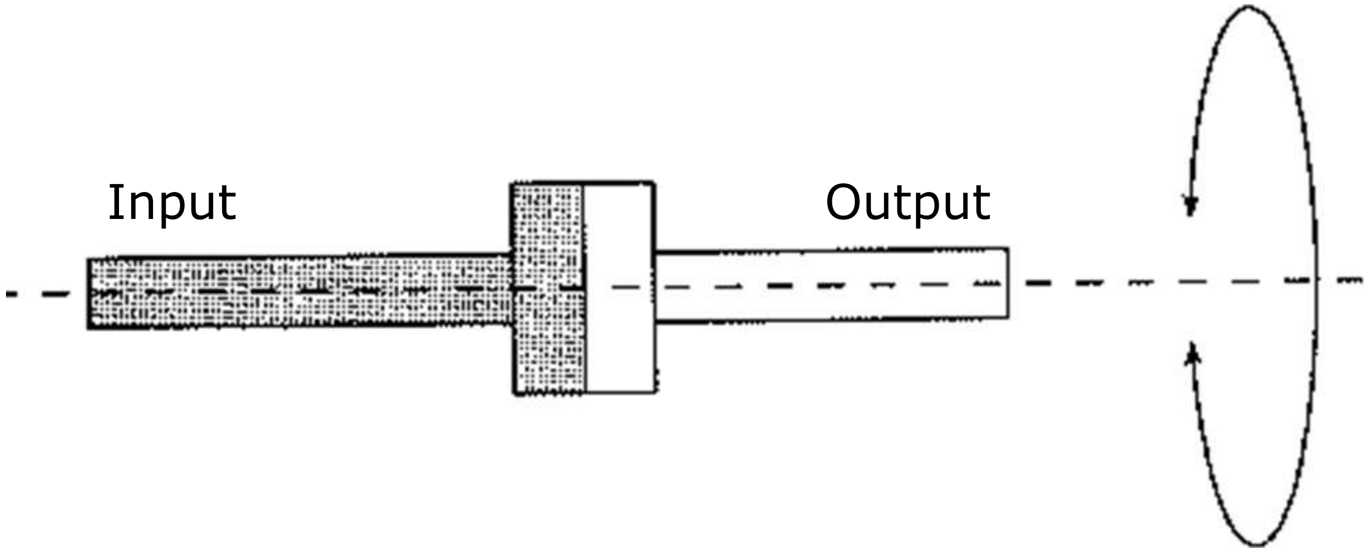
# Basic Joint Types of Robotic Systems

# Rotational Joint (R)



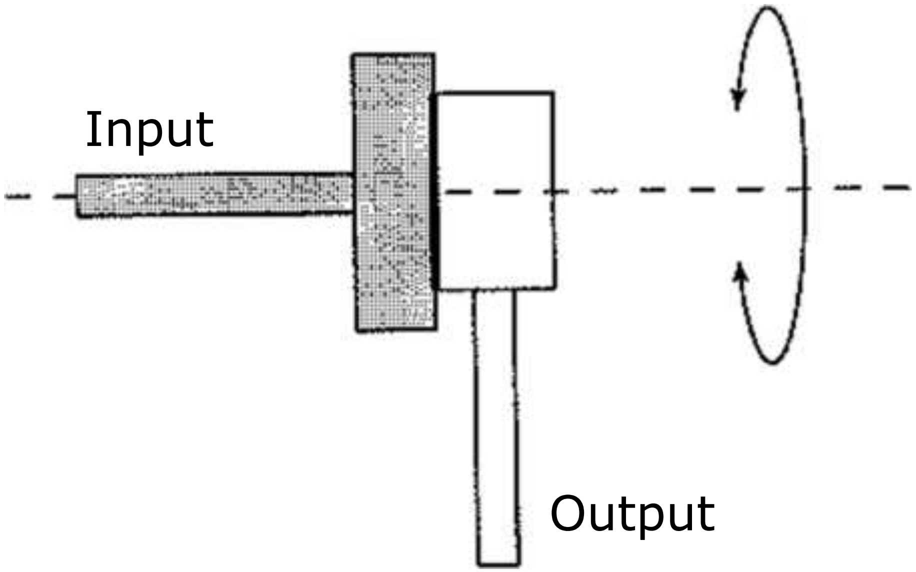

[*Siegert, Bocionek 96*]

# Torsion Joint (T)



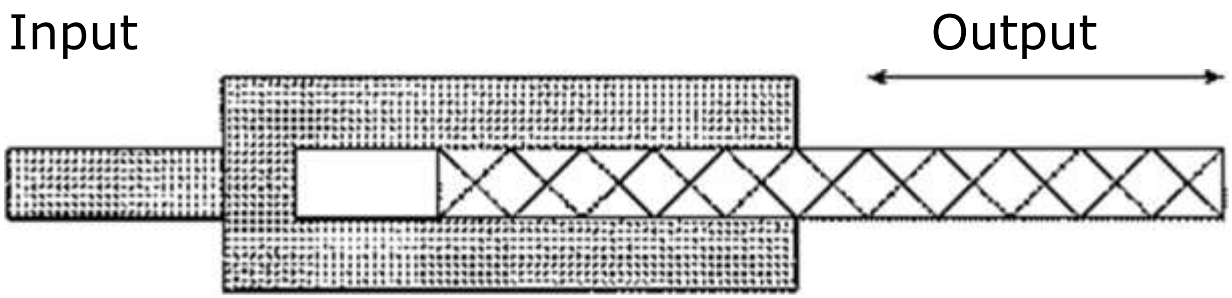

[*Siegert, Bocionek 96*]

# Revolute Joint (V)




[*Siegert, Bocionek 96*]

# Linear Joint (L)




[*Siegert, Bocionek 96*]

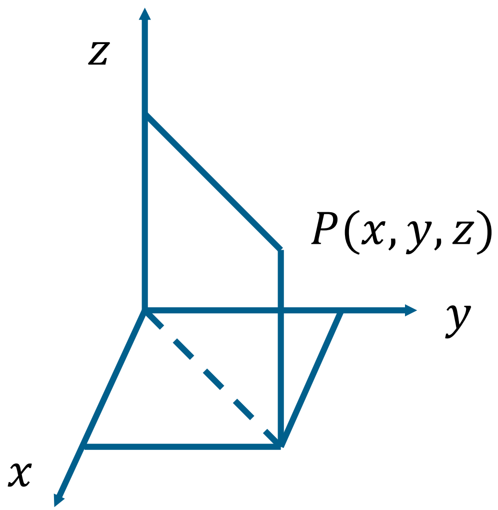
# Definition Workspace

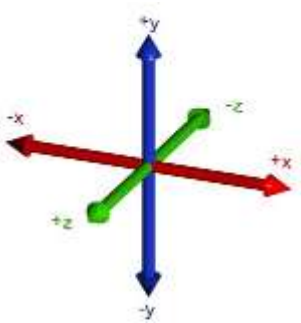
- Workspace consists of all points, which are reachable by the robot hand
  - At least 3 DOF (degree of freedom) necessary for a 3-D space
  - At least 3 basic joints necessary for a 3-D space

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

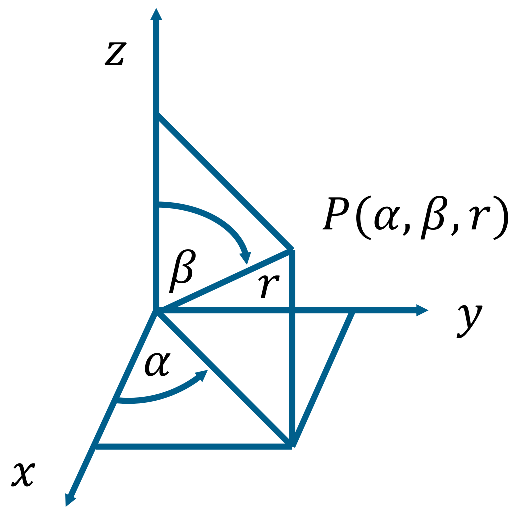
# Reminder: 3D Coordinate Systems

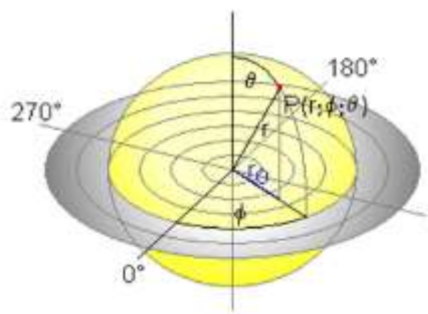
Cartesian coordinates



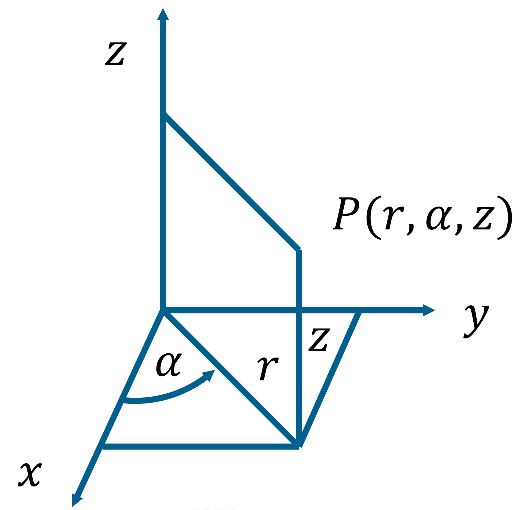



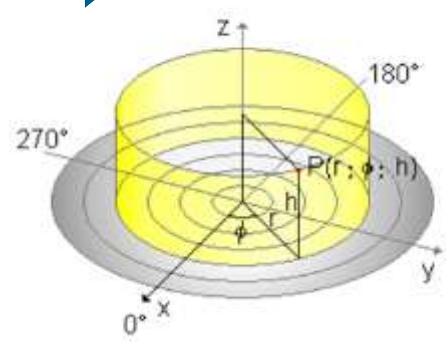

Spherical coordinates




Cylindrical coordinates

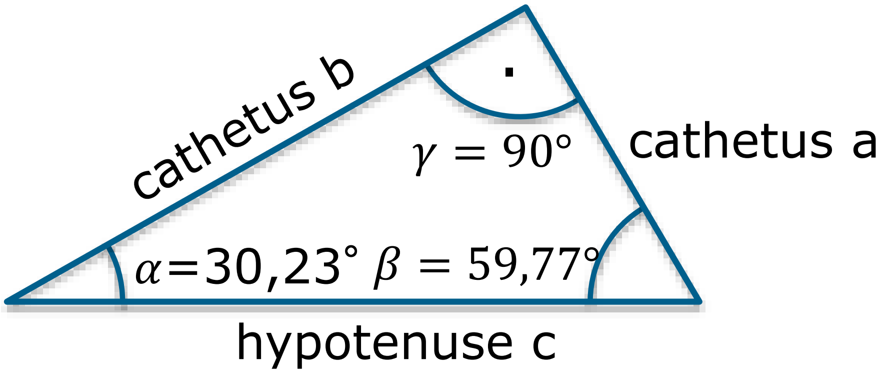


# Reminder: Trigonometric Functions

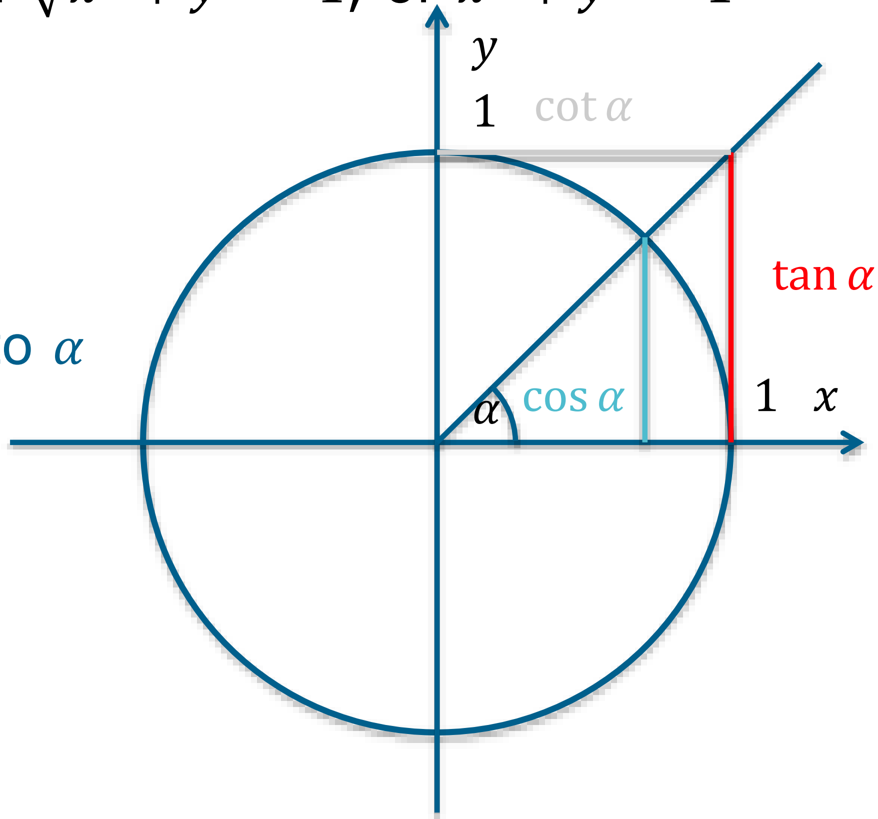
The trigonometric functions $\sin$, $\cos$, $\tan$, $\cot$ are defined as the ratios of corresponding sides in a right triangle (between $0°$ and $90°$).

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



# Reminder: Trigonometric Functions

- For arbitrary angles we define them for the unit circle
  - 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 $\alpha$:
  Length of the arc of
  a unit circle corresponding to $\alpha$



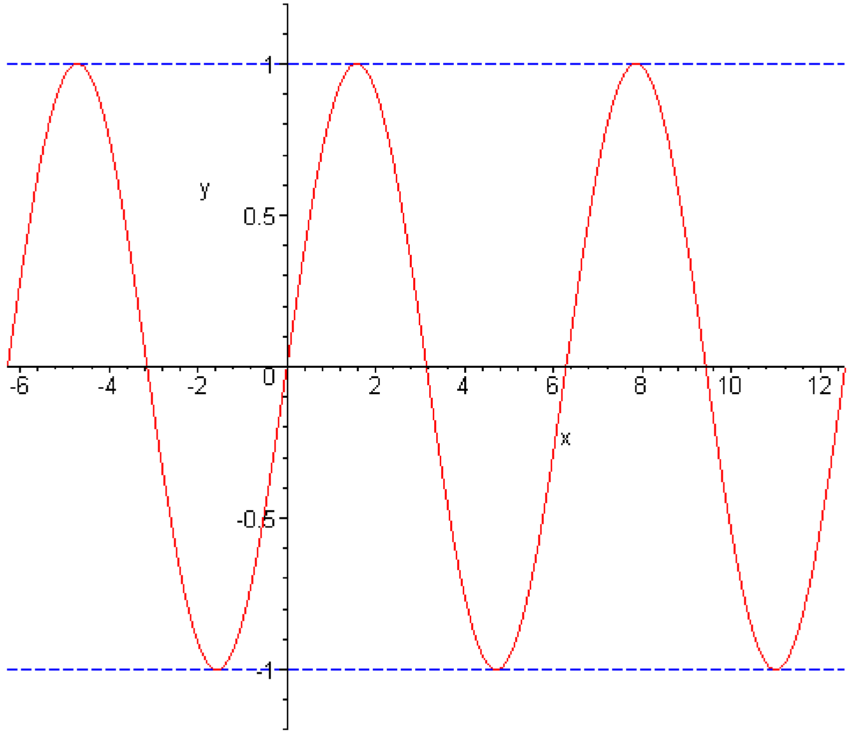
# Reminder: Properties of Sine and Cosine

| | $\sin x$ | $\cos x$ |
|---|---|---|
| Domain | $-\infty < x < \infty$ | |
| Codomain | $-1 \leq \sin x \leq 1$ | $-1 \leq \cos x \leq 1$ |
| Period | $2\pi$ | |
| 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 | $\sin(90° - \alpha) = \cos \alpha$ | $\cos(90° - \alpha) = \sin \alpha$ |

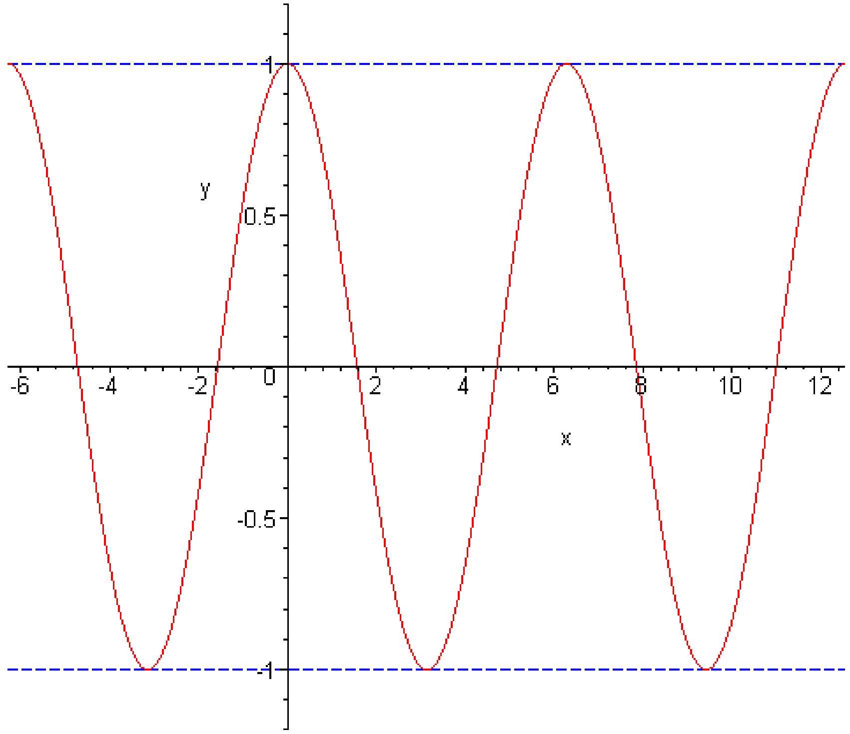
# Reminder: Table of Values

| | 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: Graphs of Sine and Cosine




Sine



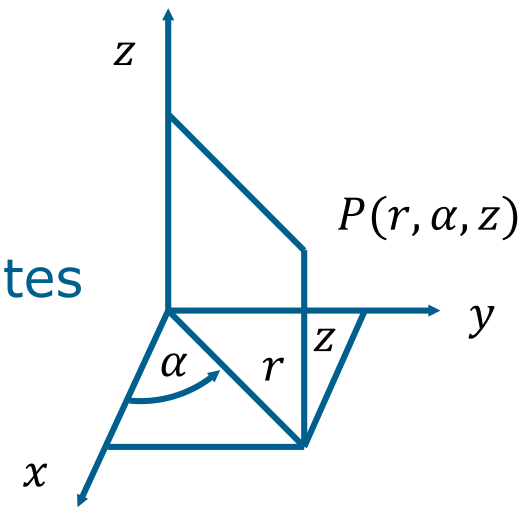

Cosine

# Reminder: Additional Theorem

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

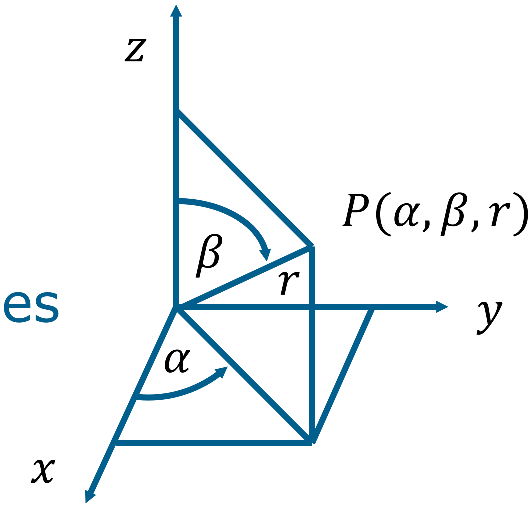
# Transformation of Coordinate Systems

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



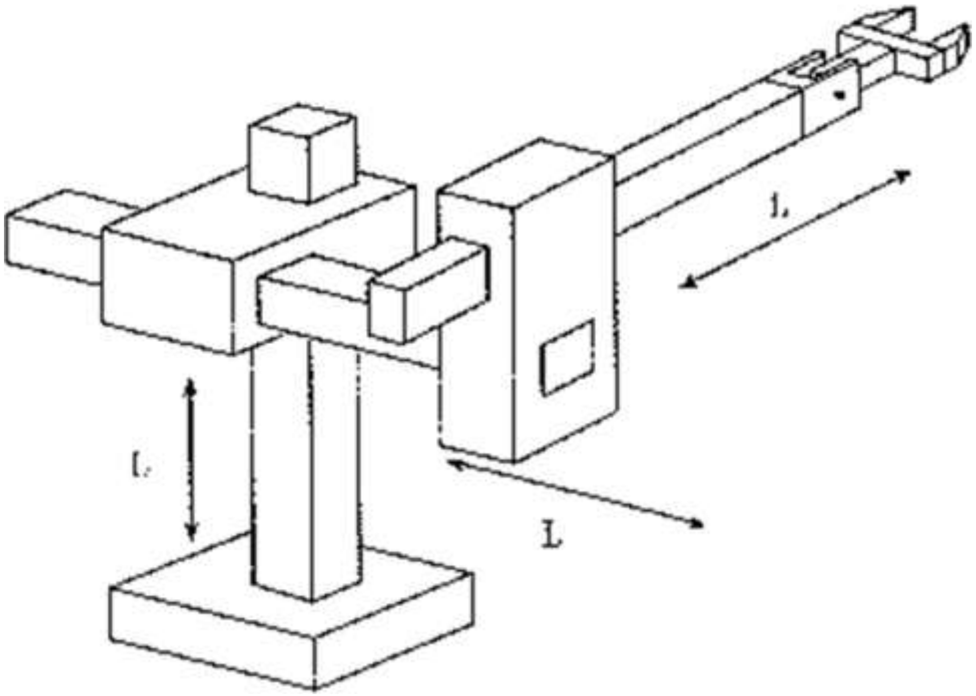
# Transformation of Coordinate Systems

- 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
  - $(r, \alpha, \beta) \rightarrow (x, y, z)$
  - $x = r \cdot \sin\beta \cdot \cos\alpha$
  - $y = r \cdot \sin\beta \cdot \sin\alpha$
  - $z = r \cdot \cos\beta$



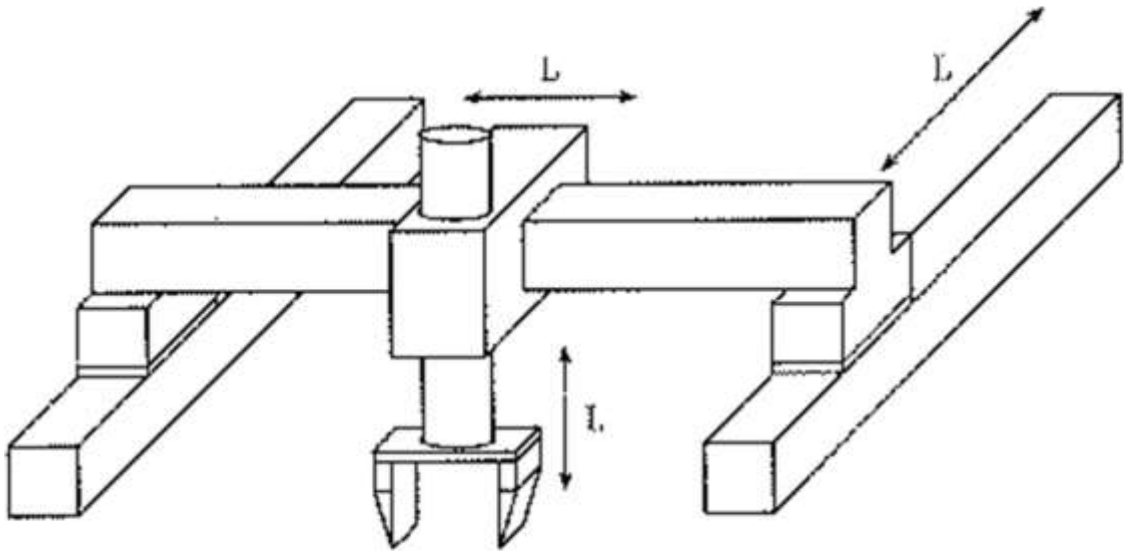
# Cartesian Robot (LLL)

- Basic shape of operational space: Cuboid
[*Siegert, Bocionek96*]



# Cartesian Robot (LLL)

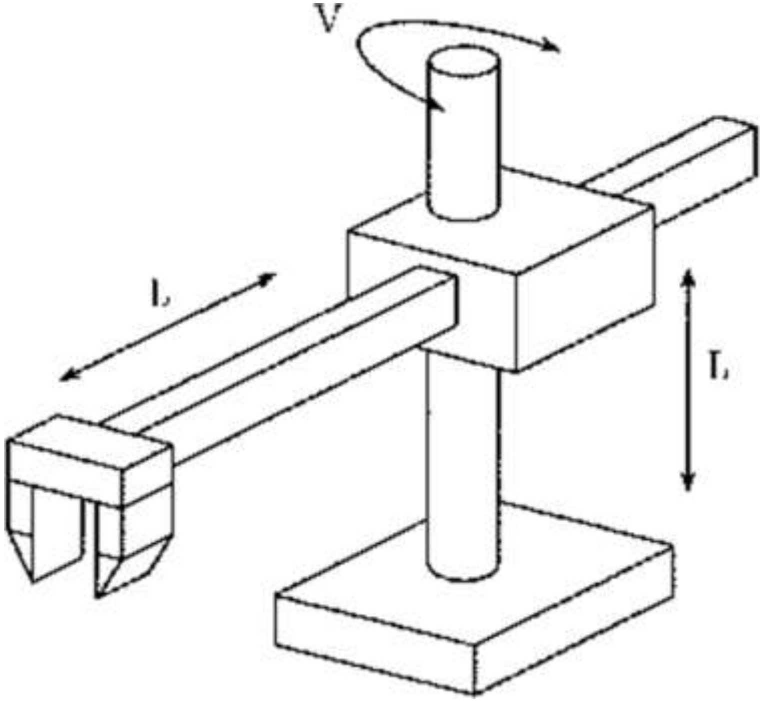
- Basic shape of operational space: Cuboid
  [*Siegert, Bocionek96*]



# Robot Arm (LVL)

- Basic shape of operational space: Cylinder
- Other possibilities: TLL, LTL

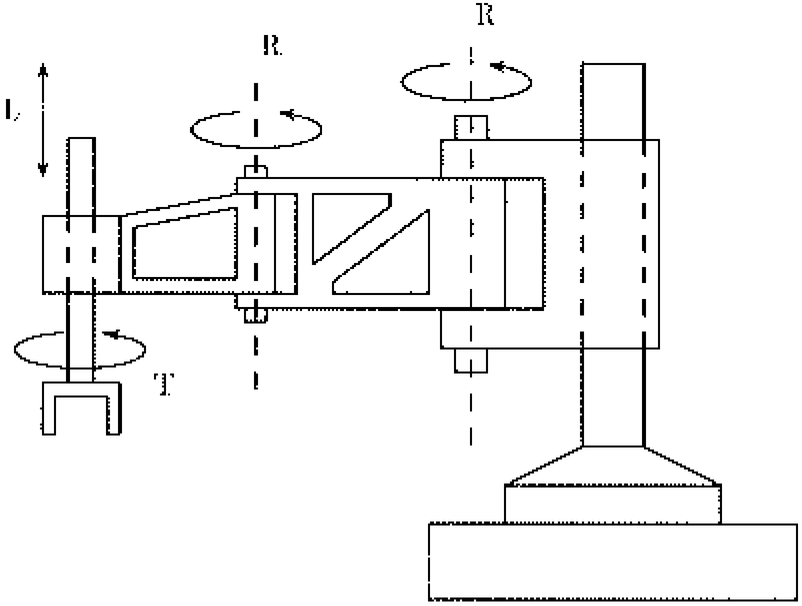
[*Siegert, Bocionek96*]



# SCARA-Robot (RRLT)

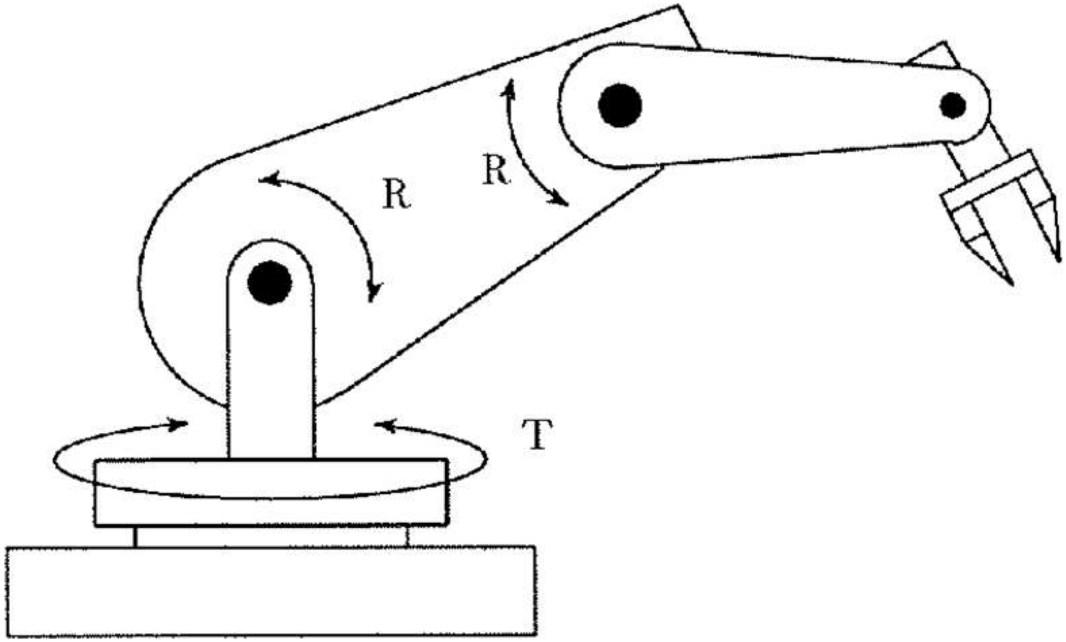
- **S**elective **c**ompliance **a**ssembly **r**obot **a**rm
- Basic shape of operational space: Cylinder

[*Siegert, Bocionek96*]

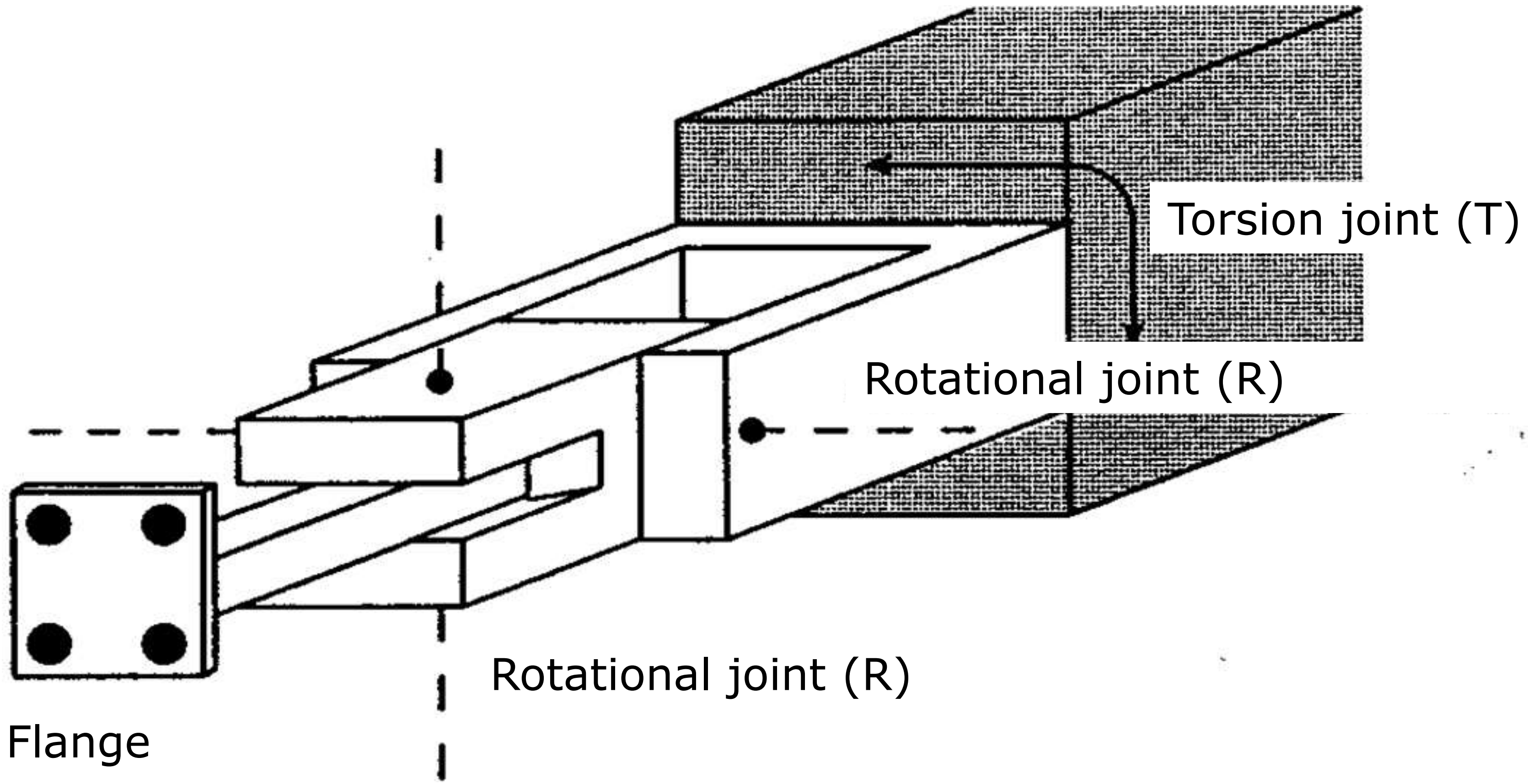


# Universal Robot Arm (TRR)

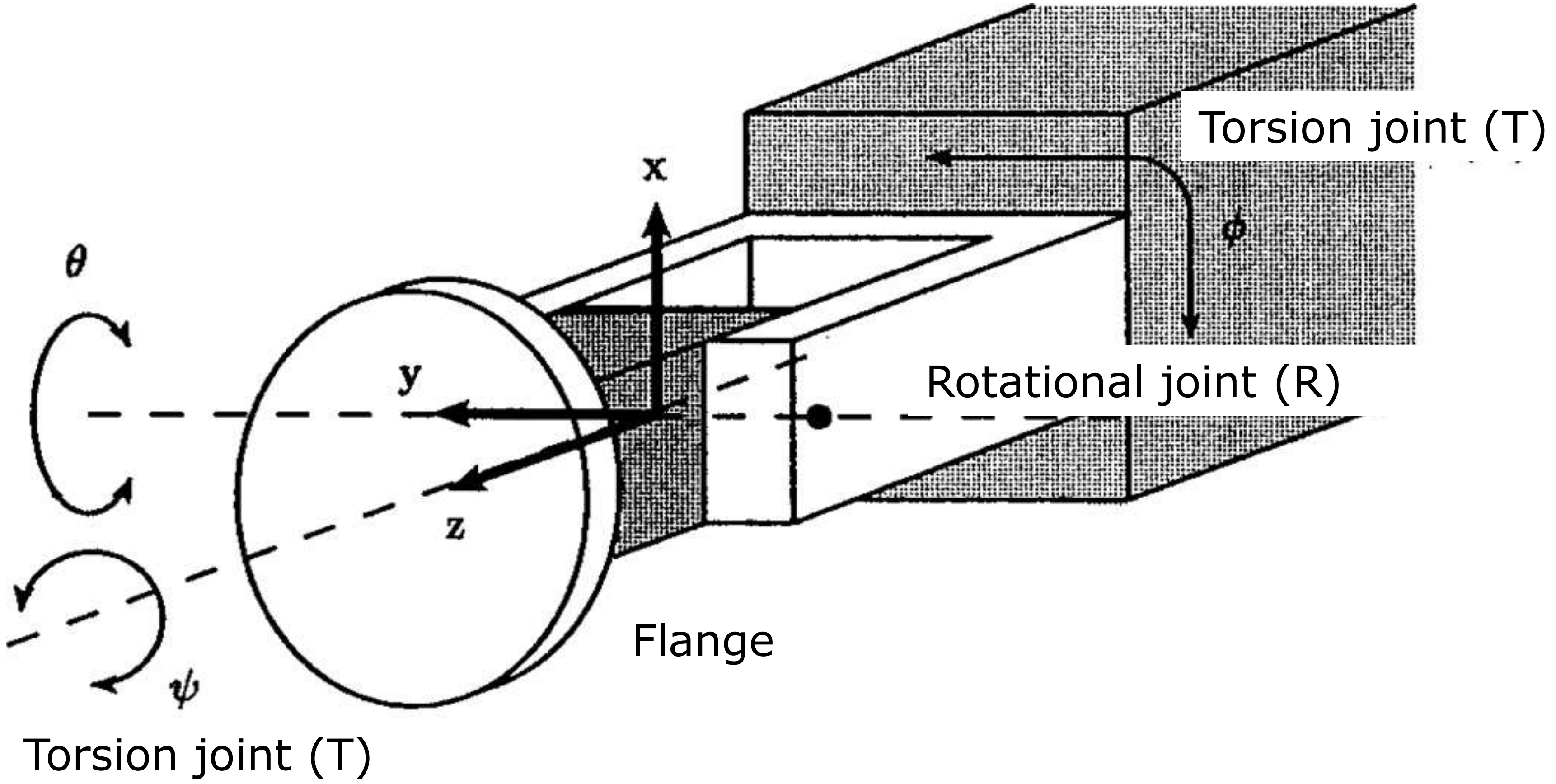
- Basic shape of operational space: Sphere
- Other possibilities: VVR



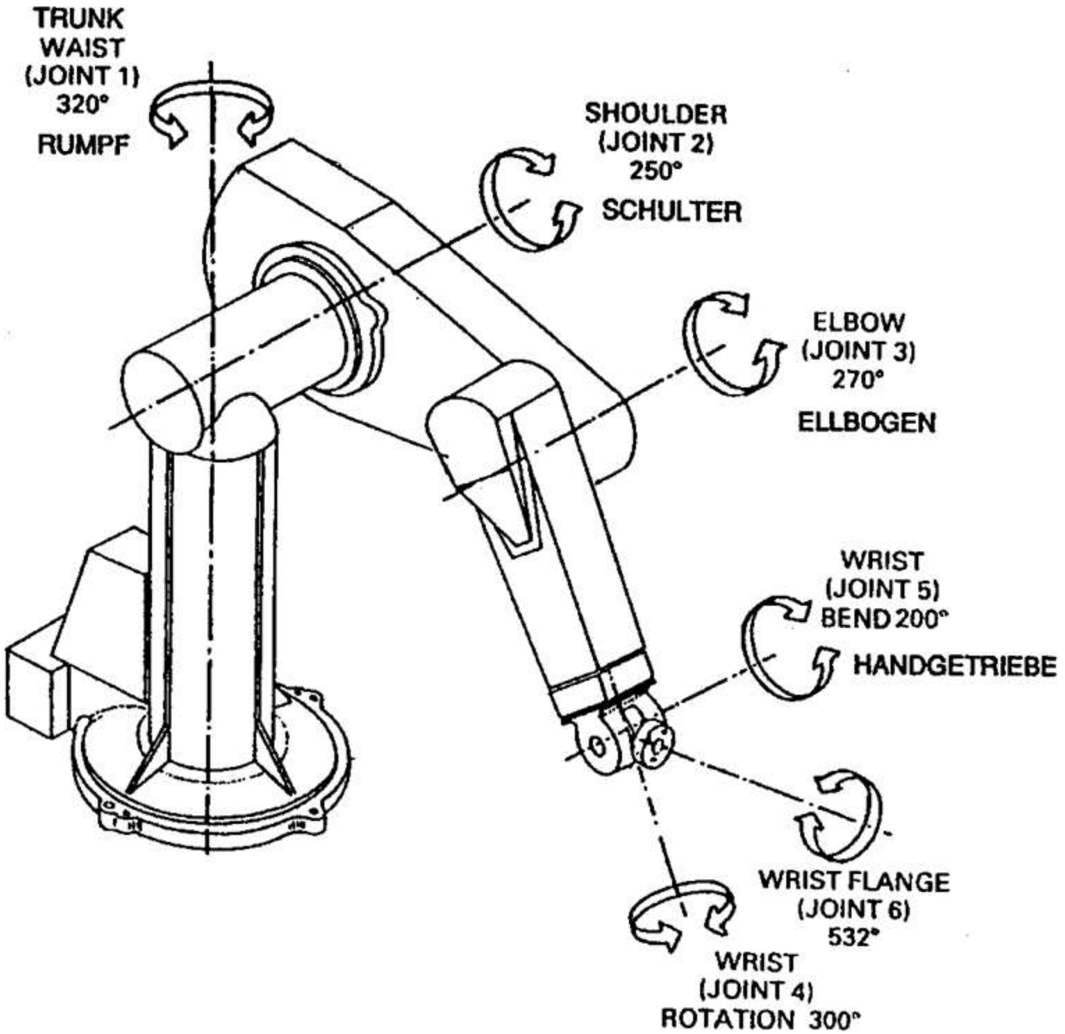
# Basic Shape of Wrist (TRR)



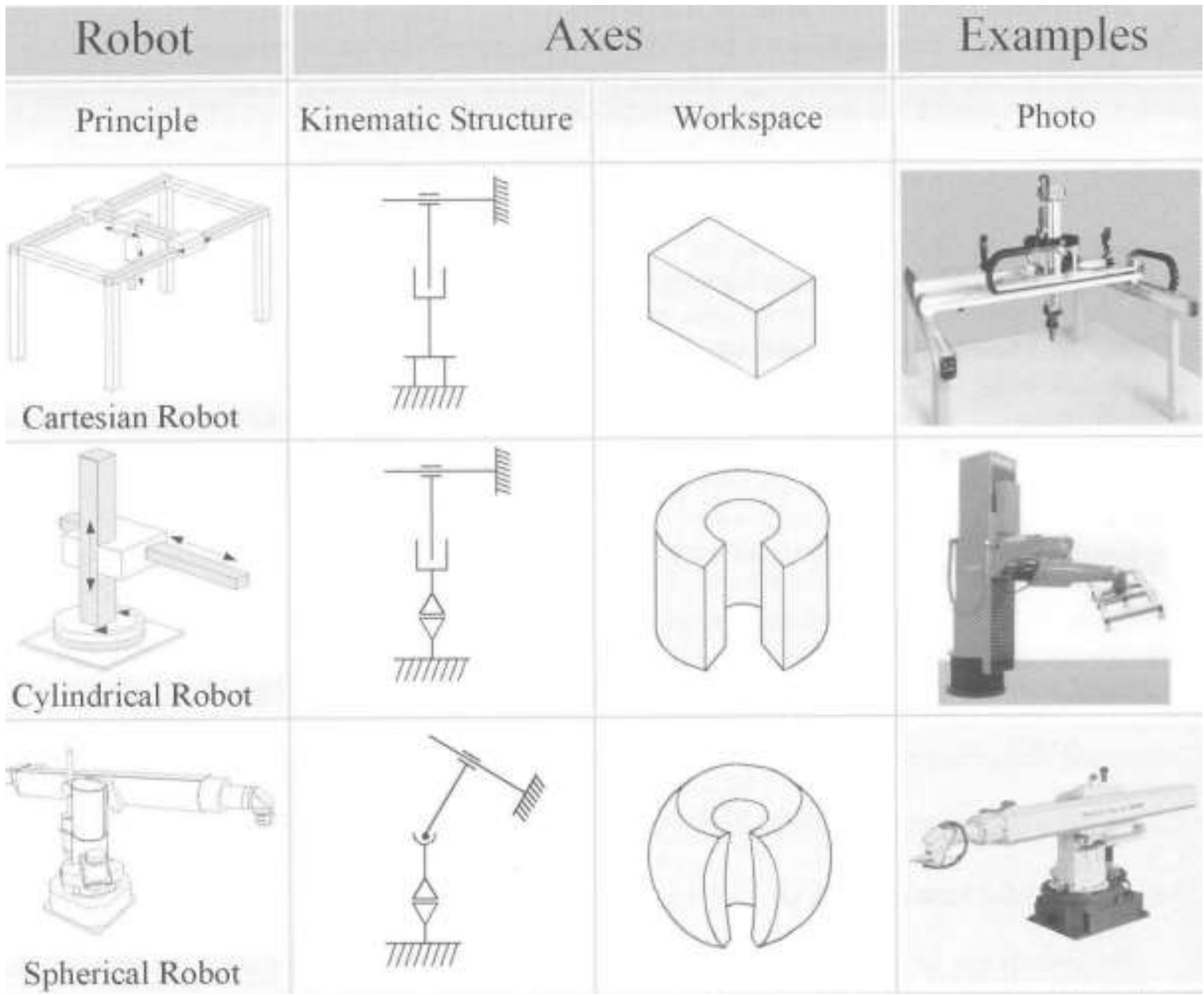
# Basic Shape of Wrist (TRT)



# Puma Robot (TVRRRT)

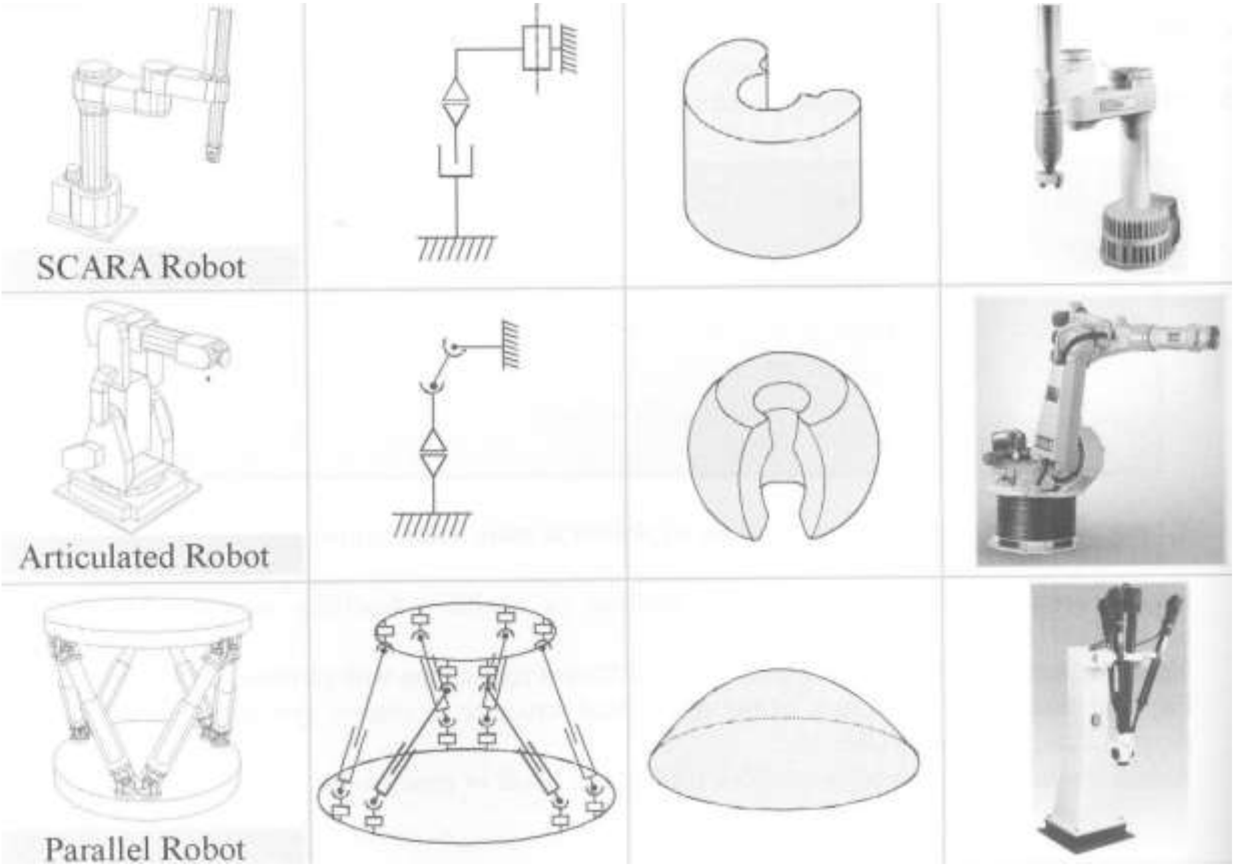


# Robot Kinematics




[*World Robotics 2003*]

# Robot Kinematics




[*World Robotics 2003*]

# Actuators

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

# Pneumatic Actuators

- Adjustments: 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: Oil pressure and controllable valves
- Pros
  - Very high forces
  - Average speed

- Cons
  - Noise
  - Additional space for hydraulics are needed
  - Leakage of oil
  - Slow and inaccurate due to viscosity of oil
- Usage
  - Big robots

# Electric Actuators

- Used energy: Electric energy
- Pros
  - Small
  - Easy to control
  - High precision
- Cons
  - Small forces
- Usage
  - Small robots for tasks which require high accuracy

# Sensor Classification: Proprioceptive

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

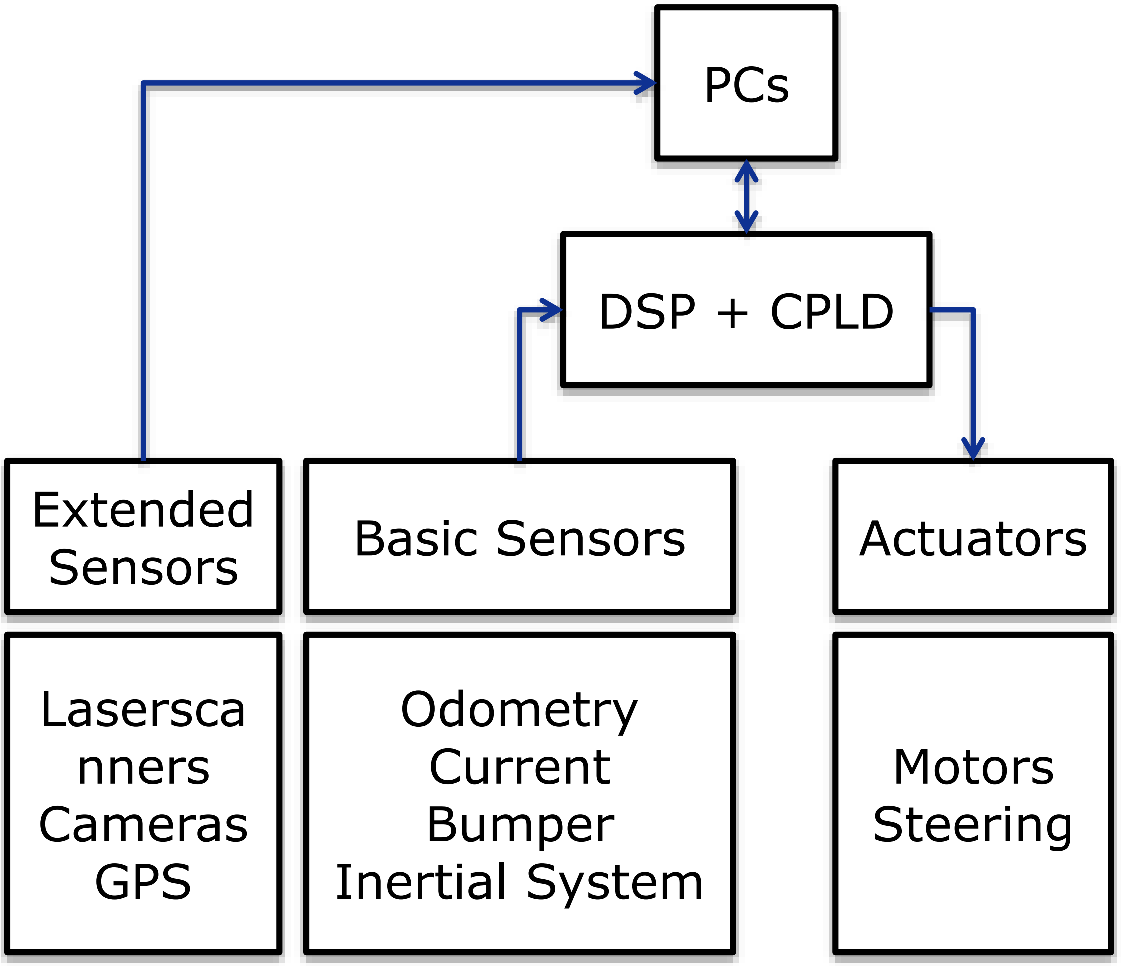
- Position
  - Potentiometer
  - Optical encoder
  - Differential transformer transducer
  - Magnetic-inductive encoder
- Velocity
  - Speed generator
  - Optical encoder
- Acceleration
  - Si-sensor
  - Piezo-electric sensor
- Orientation
  - Gyroscope
  - Geomagnetic sensor

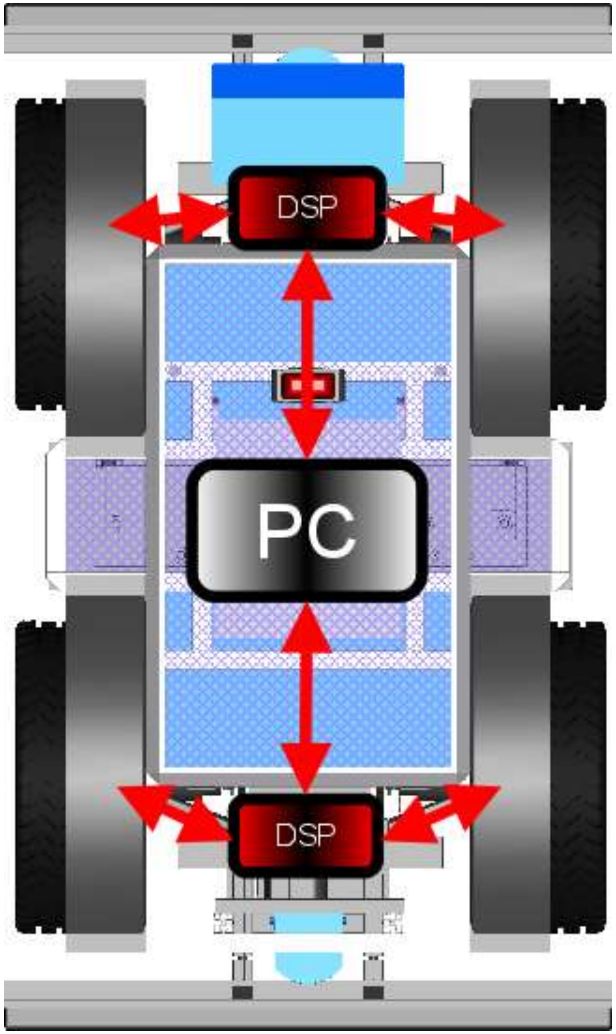
# Sensor Classification: Exteroceptive

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

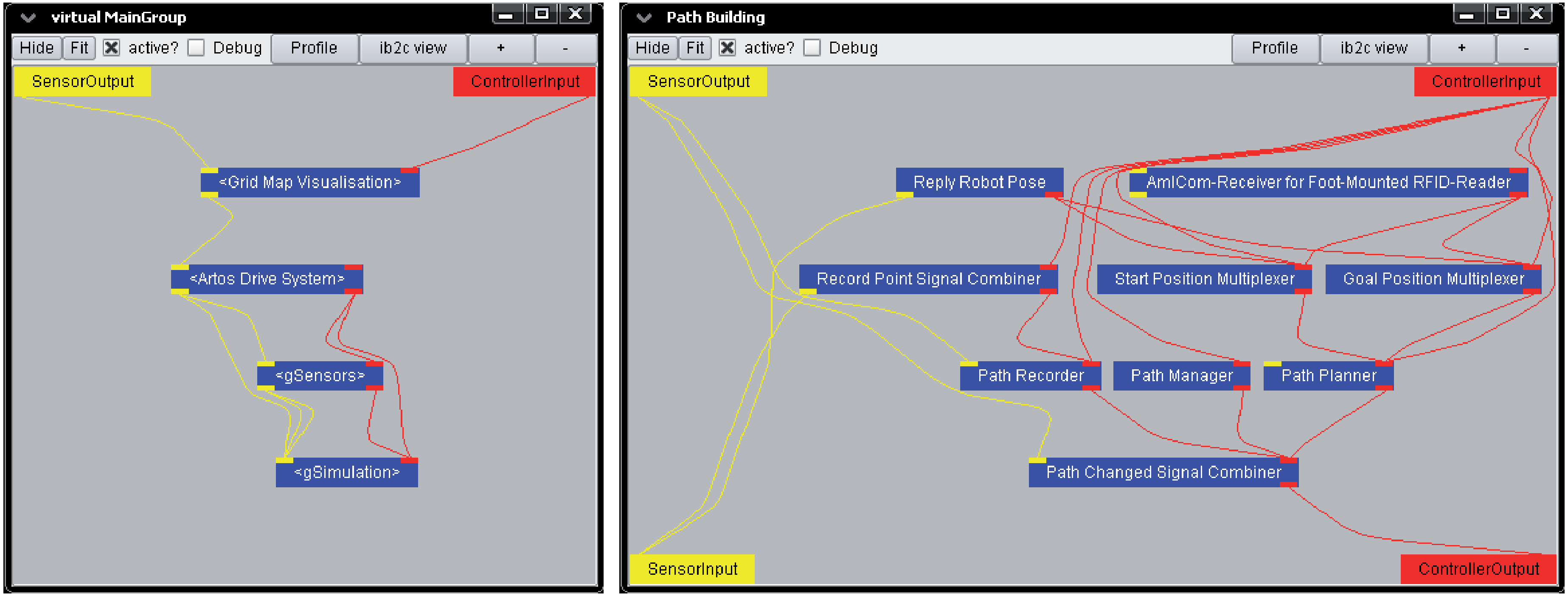
- "Feel"
  - Artificial skin
  - Sliding sensors
  - Force-torque-sensors
- Approach
  - Inductive, capacitive sensors
  - Optical sensors
  - Acoustic sensors

# Electronics and Computing Architecture



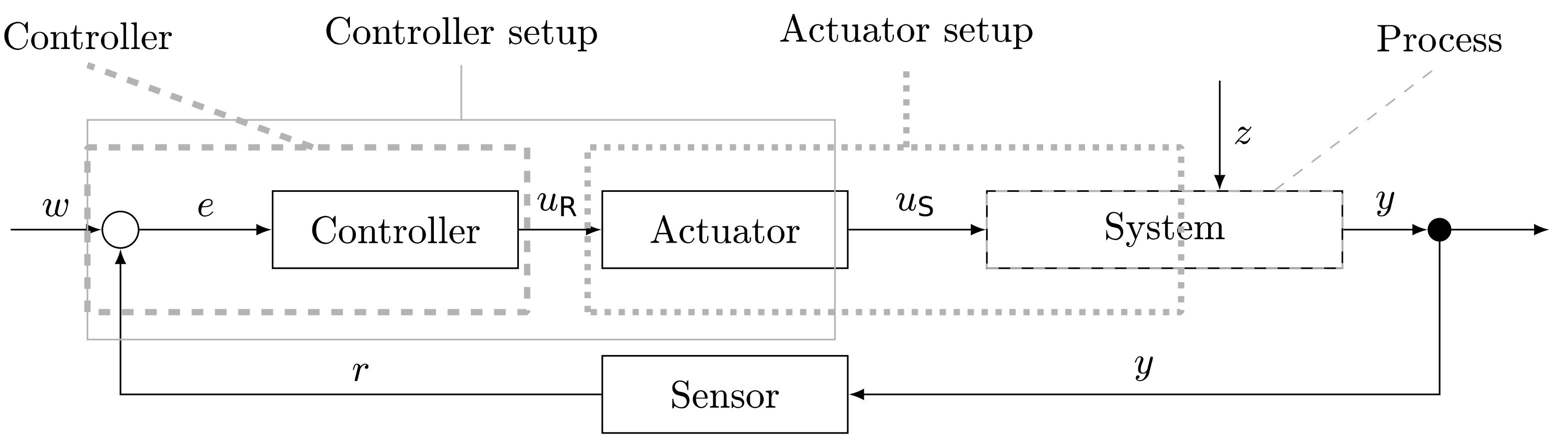
# Software Architecture and Middleware



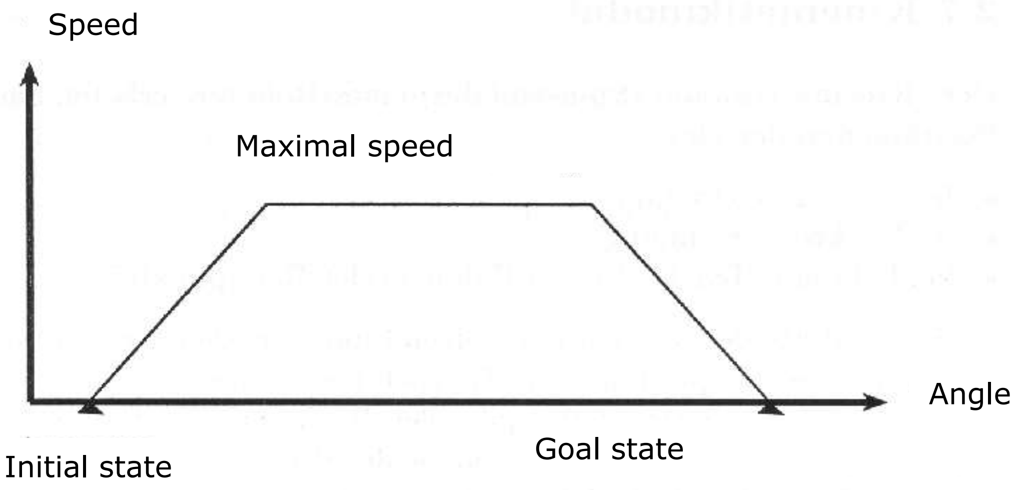

MCABrowser visualizing MCA2 application

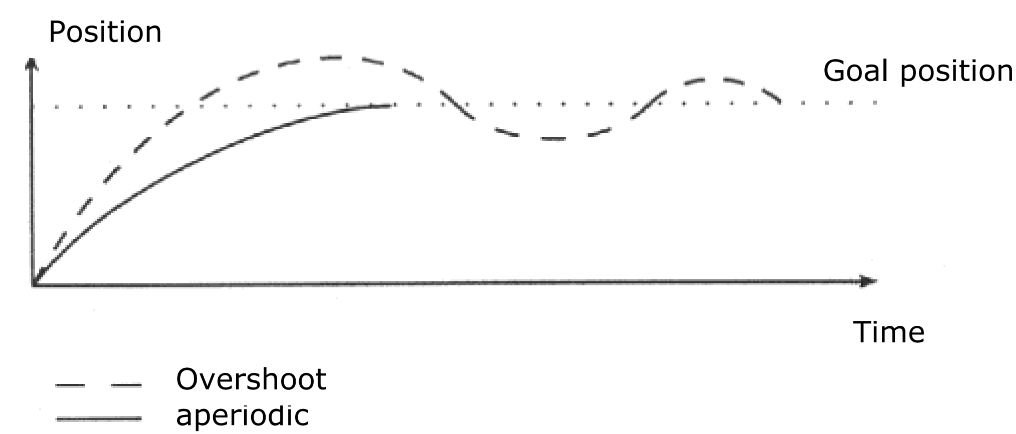
# Closed Loop Control

- 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



# Goals and Attributes of a Closed Loop Control

Selection of controller type and calculation
of its parameters for …

- Steady state accuracy: control difference vanishes for $t \to \infty$
- 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
  $\to$ approximations for calculation of control parameters are feasible

# 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 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 large 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

# Next Lecture …

Spacial kinematics

- Position of objects
- Orientation