Introduction to Robotics

Ing. Tomas Borovicka

Department of Theoretical Computer Science (KTI), Faculty of Information Technology (FIT)
Czech Technical University in Prague (CVUT)

BIE-ZUM, LS 2013/14, 12. lecture

https://edux.fit.cvut.cz/courses/BIE-ZUM/

Summary of Previous Lecture

- Artificial Neural Networks
- Perceptron
- Multilayer perceptron
- Neuroevolution
- Self Organizing Maps

The term **robot** was first introduced by famous Czech writer Karel Čapek in his play "R.U.R. Rossum's universal robots" (in 1921).

- People have tendency to think about robot as some anthropomorphic and intelligent artificial creature that can replace human in any job as an universal worker. That is still science fiction, robots have many different appearances and purposes and humanoid robot is the least common in practical use.

Robotics

- Design, construction, and use of machines (robots) to perform tasks done traditionally by human beings.
- Robotics is multidisciplinary field which includes:
    - mathematics,
    - physics,
    - biology,
    - mechanical engineering,
    - electrical engineering,
    - computer science.

Definition of a Robot

Robot is physical agent that performs actions to accomplish given tasks by manipulating the physical world. Robot is equipped with:

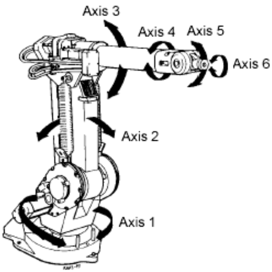
- **sensors** to perceive the environment,
  (camera, sonar, laser, gyroscope, accelerometer)
- **effectors** to assert physical forces on the environment.
  (legs, wheels, joints, grippers . . .)

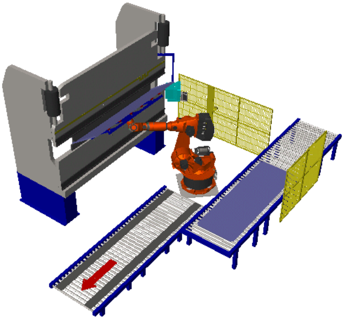
Categories of Robots

- Manipulators
- Mobile robots
- Mobile manipulators
- Humanoid robots

Manipulators

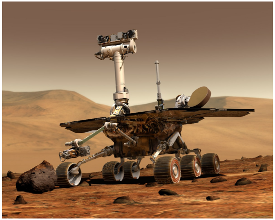
- Physically anchored to their workplace.
- Motion usually involves a chain of controllable joints, enabling robot to place its effectors in any position within the workplace
- the most common type of industrial robots.
- Usually operate in highly controlled environment, for instance manufacturing.





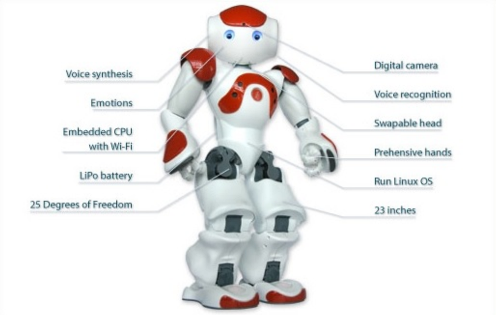
Mobile Robots

- Move in the environment using wheels, legs, or similar mechanisms.
- Have more autonomy.
- Need to know about its position and orientation.
- **Mobile manipulators** combines mobility with manipulation.



Humanoid Robots

- Humanoid robots mimic the human appearance and behavior.
- Generally have a torso, a head, two arms, and two legs.
- Designed for:
  - perform human tasks and interacting with human tools,
  - study of bipedal locomotion,
  - entertainment.



Robot Sensors

- Perceptual interface between a robot and the environment.
- Allow them to perceive the environment.
- Diverse set of sensors including cameras, sonars, lasers...

- **Passive & Active**
  - Passive sensors capture signals that are generated by other sources in the environment (camera, microphone)
  - Active sensors emit some form of energy into the environment and measures reflection caused by environment.
- **Proprioceptive & Exteroceptive**
  - Proprioceptive are sensors which measure robot's internal state.
  - External are sensors which measure the world around a robot.

Parameters of sensors

- **Field of view and range.**
  - Vertical and horizontal view angle, view distance.
- **Accuracy, repeatability, and resolution.**
  - Error in measurements, error variance, minimal increment, sensitivity.
- **Responsiveness in the target domain.**
  - Environment absorption/deflection, signal-to-noise ratio.
- **Power consumption.**
  - Passive/active, life on battery.
- **Hardware reliability, complexity.**
  - Physical limitations, work cycles, calibration requirements.
- **Size.**
  - Size matters!
- **Price.**
  - $x or $xxx . . .

Types of Sensors

- **Location sensors**
  - Give the robot information about its position in the environment (GPS).
- **Proximity sensors**
  - Give the robot the ability to detect presence of objects in the environment or measure the distance between the sensor and object (sonar, infrared).
- **Tactile sensors**
  - Provide the robot with the ability to touch and feel (whiskers, bump panels).
- **Force/Pressure sensors**
  - Provide the robot with a sense of the force being applied and the direction of the force (includes also torques and moments).
- **Vision & voice sensors**
  - Vision includes mainly cameras, which produce images of the environment that can then be analyzed using computer vision and image recognition techniques.
  - Voice includes microphones, which produce sound signal that can then be analyzed using speech recognition etc.
- **Proprioceptive sensors**
  - gyroscope, accelerometer, shaft decoders (odometer) . . .

Effectors

- Effectors are the means by which robots move, manipulate or interact with the environment.
  Two basic ways of using effectors:
  - to move the robot around $\rightarrow$ locomotion
  - to move other object around $\rightarrow$ manipulation
- **Locomotion effectors**:
  - legs (for walking/crawling/climbing/jumping/hopping)
  - wheels (for rolling)
  - arms (for swinging/crawling/climbing)
  - flippers (for swimming)
- **Manipulation effectors** are connected by joints, that allows particular movement:
  - rotary (rotation around a fixed axis)
  - prismatic (linear movement)
- **End-effectors**:
  - grippers,
  - special purpose tools.

Effectors

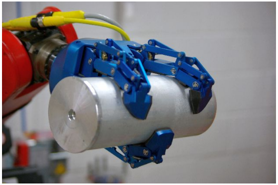
Power sources/actuators

- Electric motors
- Hydraulic cylinders
- Pneumatic cylinders
- McKibben Artificial Muscles
- Piezoelectric materials

Effectors

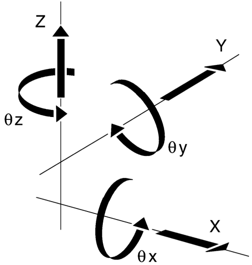
End-effectors

- Grippers
- Machine Tools
- Measuring Instruments
- Laser/Water Jet Cutters
- Welding Torches



Robot Motion

- To describe motion abilities of a robot and its effectors we use the concept of **degrees of freedom** (DOF).
  - Each independent direction in which a robot, or one of its effectors, can move is one degree of freedom.
- Degrees of freedom describes **kinematic state** of the robot; if we add velocity for each DOF we are talking about **dynamic state**.



Rotation about x axis:

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\varphi) & -sin(\varphi) \\ 0 & sin(\varphi) & cos(\varphi) \end{pmatrix}$$

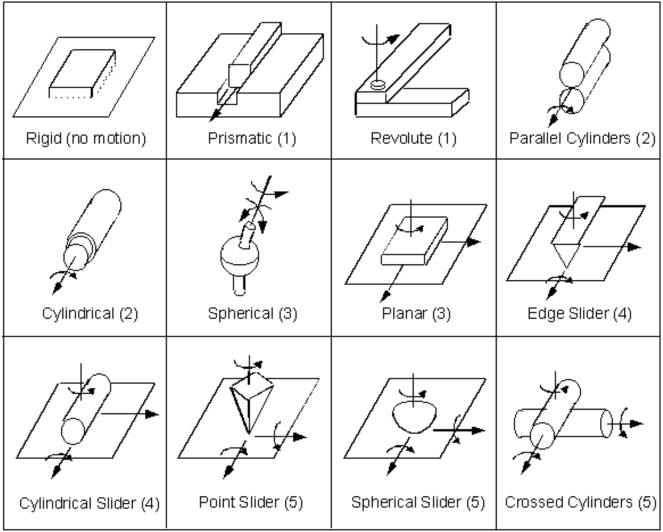
Translation on x axis:

$$t_x(\Delta) = \begin{pmatrix} \Delta x \\ 0 \\ 0 \end{pmatrix}$$

Joints

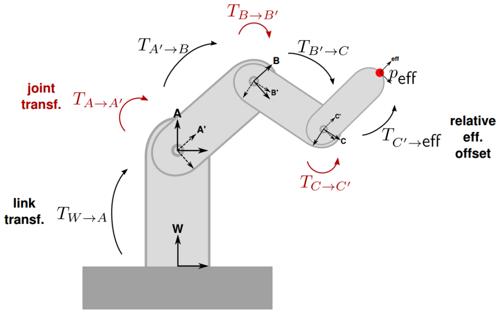
- A manipulator may be seen as a set of parts (called links) connected in a chain by joints.
- We distinguish between two basic types of joints:
  - **revolute joints** for rotation about the joint axis,
  - **prismatic joints** for linear translation along the joint axis.
- Both, revolute and prismatic, joints have one degree of freedom.
- Any joint with *n* degrees of freedom can be modeled as *n* joints of one degree of freedom.

Types of Joints



Forward Kinematics

- The forward kinematics of a manipulator refers to the calculation of the position and orientation of its end-efector relative to its base position as a function of the joint variables.



(http://ipvs.informatik.uni-stuttgart.de/)

Forward Kinematics

Transformation Matrix

Homogeneous Transformation Matrix

Let x^B be coordinates of some point in a frame B, then coordinates of the same point in a frame A (i.e. relative to A) are

$$x^A = t + Rx^B,$$

where t denotes translation vector and R rotation matrix. We define transformation matrix $T_{A \to B} \in \mathcal{R}^{4x4}$ that describes the coordinate transformation from x^A to x^B as

$$T_{A \to B} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

Then the coordinates of point x relative to frame A are

$$x^A = T_{A \to B}\, x^B.$$

Kinematic Chain

- Kinematic chain is composed of links and joints.
- For the purpose of obtaining kinematic equations a link is considered only as a rigid part which defines the relationship between two neighboring joints.

Kinematic chain

To describe coordinates of a point x in frame W, knowing the coordinates in frame C and relative transformation matrices $T_{W \to A}$, $T_{A \to B}$, $T_{B \to C}$, we can combine them into the kinematic chain written as a product of these matrices

$$T_{W \to C} = T_{W \to A}\, T_{A \to B}\, T_{B \to C}.$$

Coordinates can be then calculate as

$$x^W = T_{W \to C}\, x^C = T_{W \to A}\, T_{A \to B}\, T_{B \to C}\, x^C.$$

Inverse kinematics

- Inverse kinematic of a manipulator refers to calculation of the set of joint angles which will achieve desired position and orientation of the tool relative to the base position.
- Much more difficult problem then forward kinematics.
  - Solving nonlinear equations.
  - Solution may not exist (not reachable workspace) or has multiple solutions (might be infinite, need to choose).
- Closed-form solutions and numerical solutions.
  - Algebraic solution - manipulating given equations into a form for which a solution is known.
  - Geometric solution - try to decompose the spatial geometry of the arm into several plane-geometry problems.
  - Jacobian inversion - transforms differential angle changes to the differential motions of the end effector.
  - Optimization methods - takes the inverse kinematics equations as minimization problem.

Robotics and AI

- Robots are manufactured as hardware.
- Artificial intelligence is a theory, principles and techniques.
- The connection between those two is that the **control of the robot** is a software agent that reads data from the sensors, decides what to do next and then directs the effectors to act in the physical world.

Robot Control

- **Lead-Through Programming**
  - The human operator physically grabs the end-effector and shows the robot exactly what motions to make for a task, while the computer memorizes the motions.
- **Teach Programming**
  - Human moves robot to required positions via controller box. Computer memorizes these configurations and plays them back in robot motion sequence.Adequate for simple, non-intelligent tasks.
- **Off-Line Programming**
  - Motions are planed and programmed in using software tools without the use of robot hardware.
- **Teleoperation**
  - Human-directed motion, via a joystick. Special joysticks that allow the human operator to feel what the robot feels are called haptic interfaces.
- **Autonomous**
  - Robots are controlled by computer, with sensor feedback, without human intervention.

Robot AI Paradigms

- Set of assumptions and/or techniques which characterize an approach.
- There are currently three paradigms for organizing intelligence in robots:

Hierarchical paradigm

Reactive paradigm

Hybrid deliberative/reactive paradigm

Robot AI Paradigm Primitives

- **Sense**

  Input: Sensor data

  Output: sensed information
- **Plan**

  Input: information (sensed or cognitive)

  Output: directives
- **Act**

  Input: Sensed information or directives

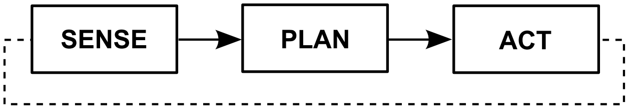
  Output: actuator commands
- **Learn**

  Input: sensed information

  Output: new interpretation

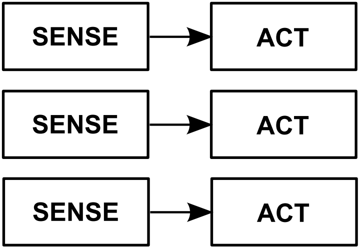
Hierarchical Paradigm

- Operates in a top-down fashion.
  - The robot senses the world, plans the next action, and then acts.
- Planning requires search, which is slow.
- Requires a (precise) world model.
  - World model based on First Order Predicate Logic.
  - Frame problem, need for closed world assumption.
  - Error recovery by updating world model.



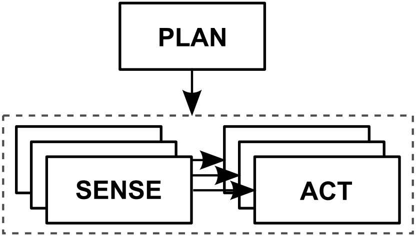
Reactive Paradigm

- Does not build world models nor plan.
- Requires high observability of the environment.
- **Pure Reactive**
  - Close connection of sensors to actuators.
  - Pre-programmed patterns of behavior.
  - Inherently concurrent.
- **Behavior based**
  - Very specific controllers good at one or few tasks.
  - Compute the best action to take independently of what the other processes are doing.
  - Combined decision.



Hybrid deliberative/Reactive Paradigm

- Mixture of hierarchical and reactive paradigm.
  - Reactive system on the bottom.
  - Deliberative system on the top.
- The robot first plans (deliberates) how to best decompose a task into subtasks (mission planning) and then what are the suitable behaviors to accomplish each subtask.



Robot Perception

- Perception provides robots with information about the environment by interpreting response of sensors.
- Each sensor measures some specific aspect of the environment that is used to describe current state.
- Includes many fields such as
  - Signal processing
  - Image and video processing
  - Feature extraction
  - Image segmentation
  - Pattern recognition
  - Visual object tracking
  - Classification
  - 3D image reconstruction

Path and Motion Planning

The **motion planning** is a problem of finding a robot motion from a start state to a goal state while avoiding obstacles in the environment, joint limits, and torque limits.

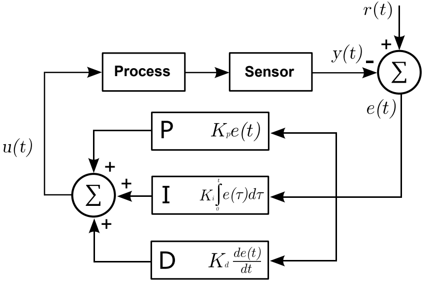
The **path planning** problem is a sub-problem of the general motion planning problem. Path planning is a geometric problem of finding a collision-free path from a start state to a goal state.

- General problem solvers
  - STRIPS
- Variations of search
  - A*
  - Dijkstra's
  - BFS, DFS
- However, such methods require discretization of the search space (cell decomposition, skeletonization).

Dynamics and Control

- The robot has a goal and may have a plan how to reach it based on the kinematic model.
- However the environment affects the robot by external forces that causes deviations (error) from the plan.
  - We can include dynamics that includes the effect of forces and make plan based on dynamic model.
  - Planning becomes too difficult due to high dimensionality.
- **Kinematic model with controller**
- The robot is sensing current state and measures difference between desirable state (error).
- Controller corrects the actions to aim towards the goal.

Basic Feedback Controllers



- **P: Proportional control**
  $u(t) = K_p \cdot e(t)$
- **PI: proportional integral control**
  $u(t) = K_p \cdot e(t) + K_i \int_{t_0}^{t} e(\tau) d\tau$
- **PD: proportional derivative control**
  $u(t) = K_p \cdot e(t) + K_d \cdot \frac{d}{dt} e(t)$
- **PID: proportional integral derivative control**
  $u(t) = K_p \cdot e(t) + K_i \int_{t_0}^{t} e(\tau) d\tau + K_d \cdot \frac{d}{dt} e(t)$

Basic Feedback Controllers

- **P: Proportional control**

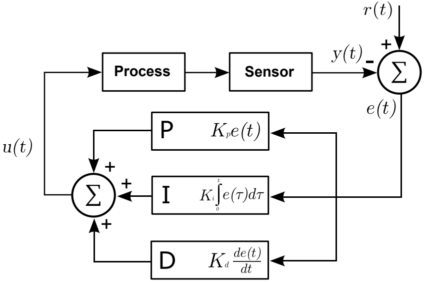
  $u(t) = K_p \cdot e(t)$

- **PI: proportional integral control**

  $u(t) = K_p \cdot e(t) + K_i \int_{t_0}^{t} e(\tau) d\tau$

- **PD: proportional derivative control**

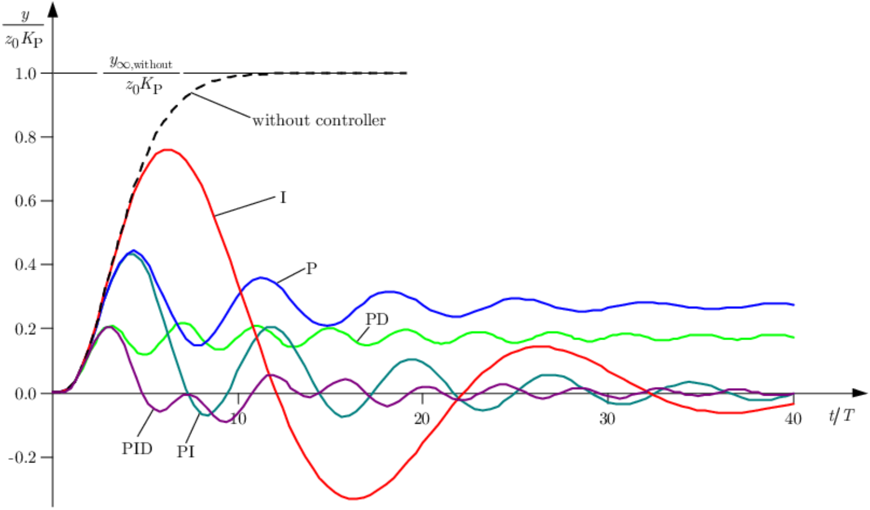
  $u(t) = K_p \cdot e(t) + K_d \cdot \frac{d}{dt} e(t)$



- **PID: proportional integral derivative control**

  $u(t) = K_p \cdot e(t) + K_i \int_{t_0}^{t} e(\tau) d\tau + K_d \cdot \frac{d}{dt} e(t)$

Basic Feedback Controllers



Reinforcement Learning

- Reinforcement learning is a method of machine learning used to solve complex control and optimization tasks in an interactive manner.
- Consider an **agent** in an **environment** that represents the problem.
- Agent performs **actions** and environment responds by giving the agent a **reward** for what it has done based on how good or how bad the action is.
- Based on the reward the agent **adapts** its behavior.