Slides have been created by Prof. Michele Focchi

webpage: https://mfocchi.github.io/Teaching/

Introduction to Robotics

Instructors:
Matteo Saveriano
Marco Camurri

Information

- Name of the course: **Introduction to robotics**
- **Instructors**:
  - Matteo Saveriano (michele.focchi@unitn.it)
  - Marco Camurri (marco.camurri@unitn.it)
- **Schedule**:
  - Thursday 16:30 – 18:30 (room A110)
  - Friday 9:30 – 11:30 (room A210)

Course organization

- **Duration**
  - February 27th to June 7th
    - 12 weeks, 4h / week
    - 48h = 36h theory + 12h lab
- **Credits**: 6 or 5(CFU)
- **Course material** (pdfs, lab sessions):
  - Moodle (code: **145959 / 145866**)
- **Slides uploaded (and updated) along with the course**

Course prerequisites

- **Self-contained course without:**
  - **Mathematics:**
    - multivariable differential calculus (gradient, Jacobian, Taylor series, chain rule)
  - **Linear algebra**:
    - Intuitive grasp of concepts from linear algebra
    - Eigenvalues, eigenvectors, rank, nullspace, and positive definite matrices. (to refresh Gilbert Strang's lectures on linear algebra)
  - **Dynamics:**
    - Newtonian dynamics in 3D, work, kinetic potential energy
  - **Control Systems (desirable):**
    - Elementary knowledge in automatic control
    - State space representation of linear systems
    - Stability criteria for linear systems

Course overview

- **Goals**:
  - Basic understanding of robotics
    - Taxonomy of robots
    - Functional units of a robot, sensors/actuators
  - Theoretical tools for kinematic analysis, motion planning, and motion control of articulated robots
    - Forward/inverse kinematics
    - Dynamic modeling of robots
    - Motion control
  - Practical implementation of theory
    - Programming of motion tasks for robot manipulators (in Python)

Lab sessions (2h each)

- Activities
  - Write code (not much)
  - Analyze results (a lot), viewer, plots
  - Discuss (a lot)
  - Use real robots (if possible…)

Evaluation

- **Final Exam (Oral):** open questions and numerical exercises on course material
- **Bonus points:**
  - Up to 3 bonus points for optional project

Resources (some are free)

Robotics:

- **Book: Robotics Modelling, Planning and Control - Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.**
- Sistemi di Automazione Industriale..Architetture e Controllo, Bonivento, Gentili, Paoli, 2010.
- Robotics 1, A. De Luca: http://www.diag.uniroma1.it/~deluca/rob1_en.php
- Robotics 2, A. De Luca: http://www.diag.uniroma1.it/deluca/rob2_en.php
- Course: Introduction to Robotics - Oussama Khatib's: http://videolectures.net/stanfordcs223aw08_introduction_robotics/
- Book: Introduction to Robotics: Mechanics and Control – J.Craig

Introduction

Robotics

What is robotics?

“ The branch of technology that deals with the design, construction, operation, and application of **robots****”*

Ok, so what is a robot?

…it is a bit more complicated to define

*Oxford Languages dictionary

Robot

Multiple definitions online:

- “A machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer.” (Oxford dictionary)

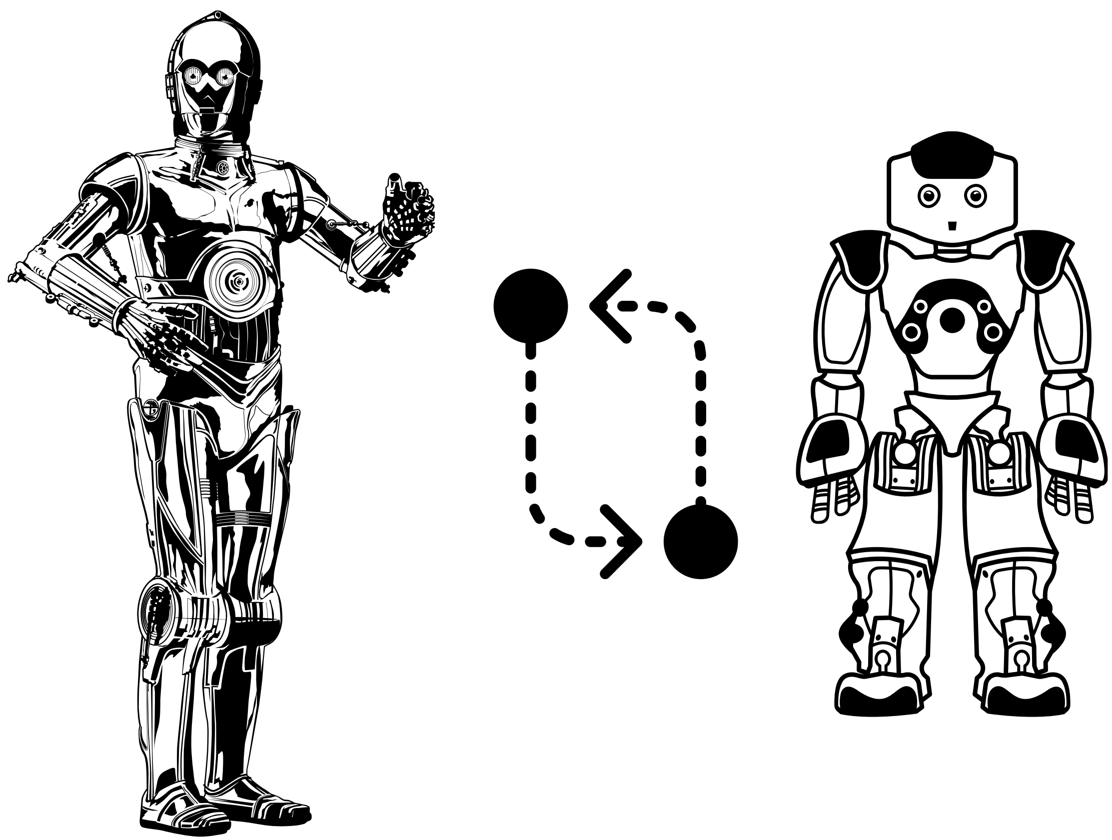
- “A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of task.” (Robot Institute of America)

- “A reprogrammable manipulator device” (British Department of Industry)

In essence they all talk about “automatic” execution of tasks…

Robots, technology and fiction

"Feedback loop" between fiction and the evolution of technology in robots



Images by Simon Child and Mahmure Alp from https://thenounproject.com

Robots(?) in fiction (before 20th century)

- Ancient Greece's **Myth of Talos** (700 B.C.), an automated bronze giant who protected the island of Crete*

- Mary Shelley's **Frankenstein** (1818), the creation of a sapient creature through a science experiment**

*Short and interesting read: https://news.stanford.edu/2019/02/28/ancient-myths-reveal-early-fantasies-artificial-life/

**Some regard this as the first science fiction novel

Automata: “acting of one’s own will”

- Machines designed to follow a predetermined sequence of operations
- Early examples of machines to execute “a task” on their own
- Mostly for entertainment, not necessarily “useful”

XV Century
Da Vinci’s “Humanoid”

XVIII Century
Letter writing automaton

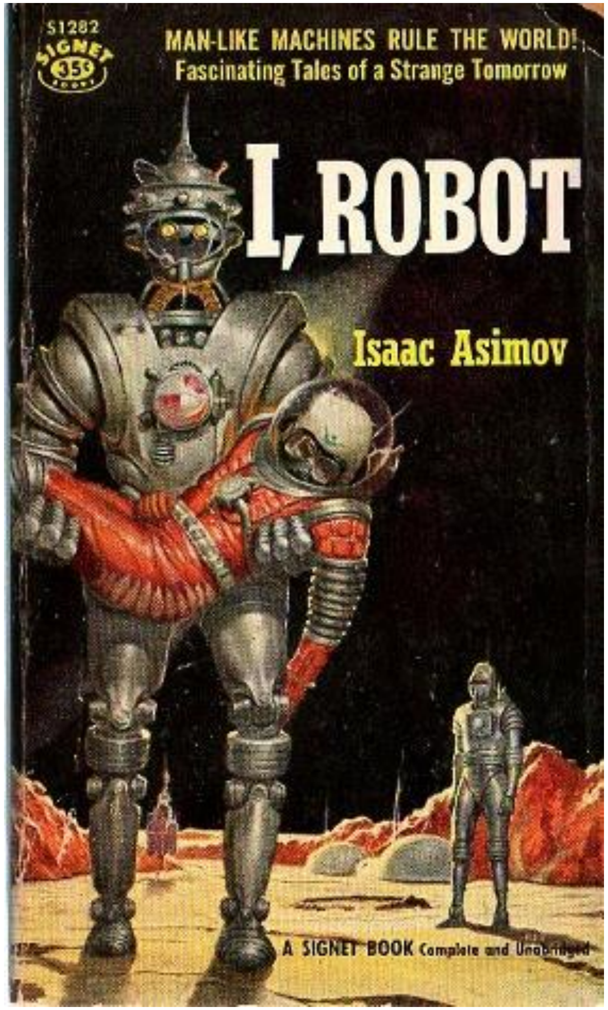
XIX Century
Karakuri puppets

Robots in fiction (20th century)

- Karel Čapek's "*R.U.R. Rossum's Universal Robots*" (1920), first time that the term "**robota**" is used (forced laborer in Czech).

- Isaac Asimov's "*I, Robot*" (1942), first use of the word "**robotics**" and the famous *Three Laws of Robotics*

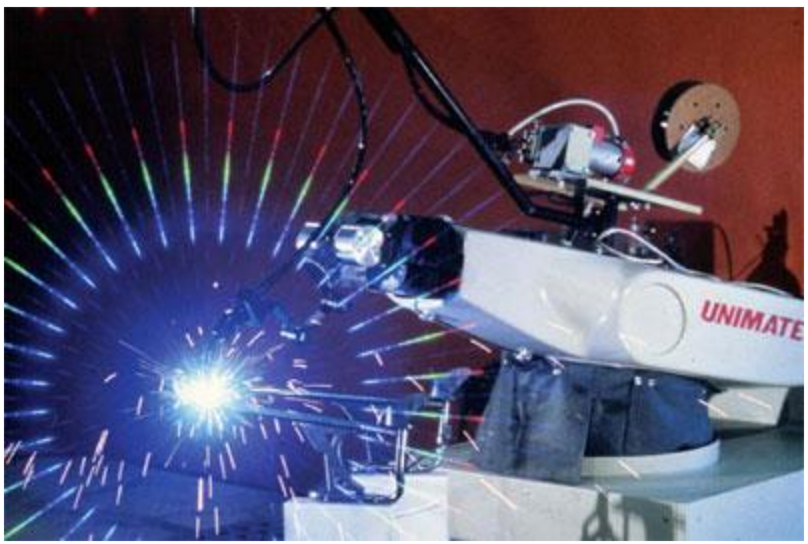
- And many movies and shows:
  - *Metropolis* (1927)
  - *Star Trek* (1965)
  - *2001: A Space Odyssey* (1968)
  - *Star Wars* (1977)
  - *Robotech* (1985)



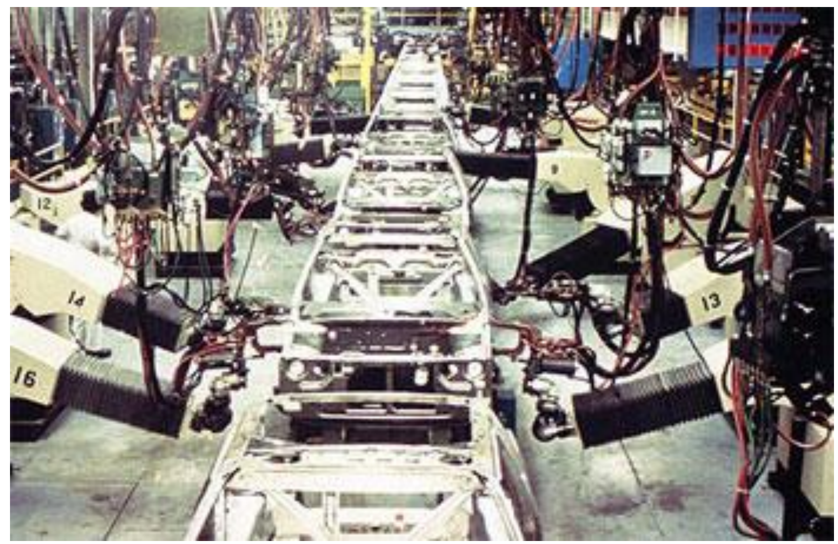
Nice recommendations on literature related to robotics: https://lithub.com/a-brief-literary-history-of-robots/

20^{th} and 21^{st} Century Robotics

- First step towards “useful” applications in industrial setups
- Early applications in General Motors car assembly lines for welding



1961, **Unimate** the first mass produced robot



General Motors assembly line (circa 1969)

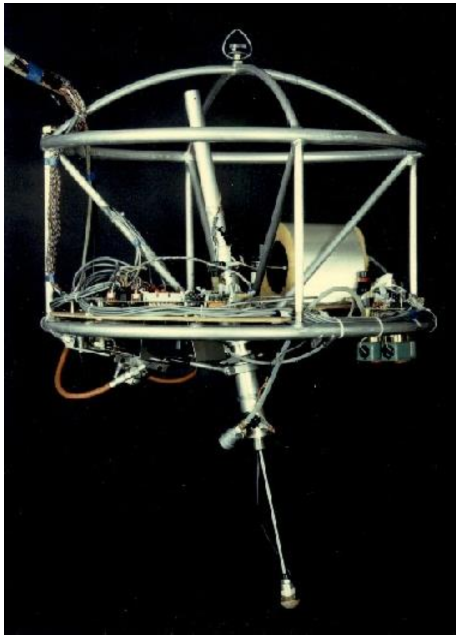
20th and 21st Century Robotics

- **Robot manipulators** have become a standard in factory assembly lines
- They are good at tasks that require both precision and speed

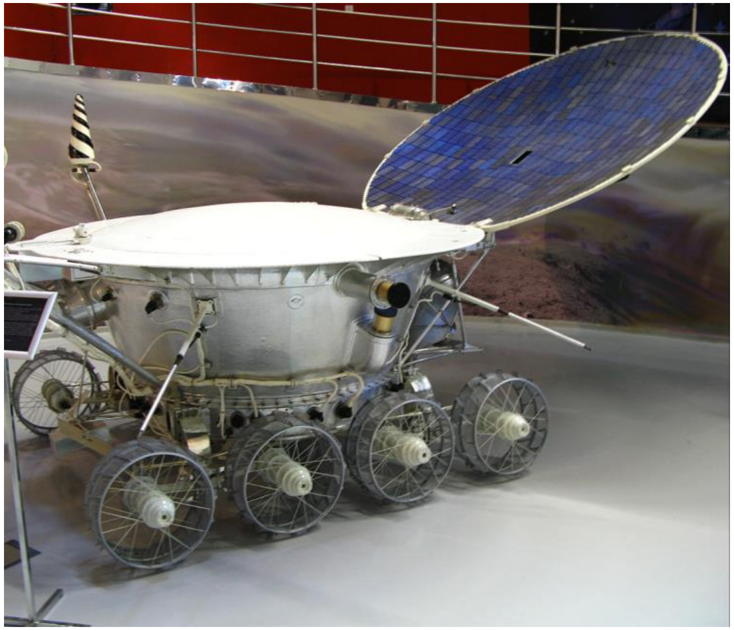
BMW car assembly line (2019, 4x)

20^{th} and 21^{st} Century Robotics

- Marc Raibert's (Boston Dynamics founder) CMU Leg Laboratory was founded in 1980
- Key steps in space exploration made by space rovers in the 70's



1983, CMU's Leg Lab 3D hopper



1970, Lunokhod lunar rover. First rover to land on a celestial body

20th and 21st Century Robotics

Legged Machines from the MIT/CMU Leg Laboratory

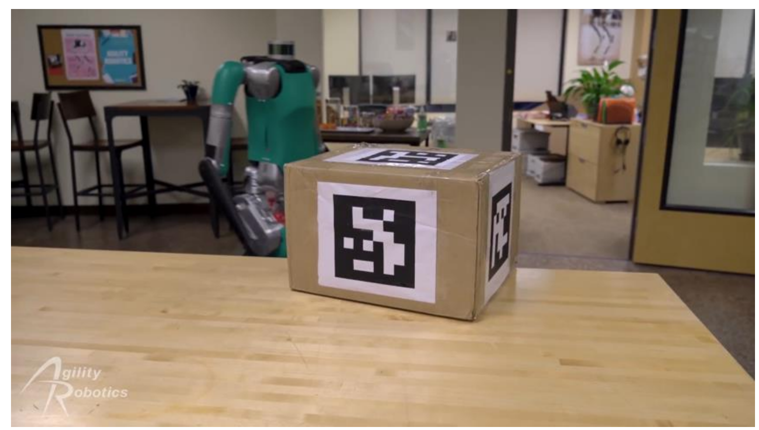
20th and 21st Century Robotics

- Legged robots are starting to be sold to the general public and industries

Boston Dynamics (U.S.A), 2020

ANYbotics (Switzerland), 2020

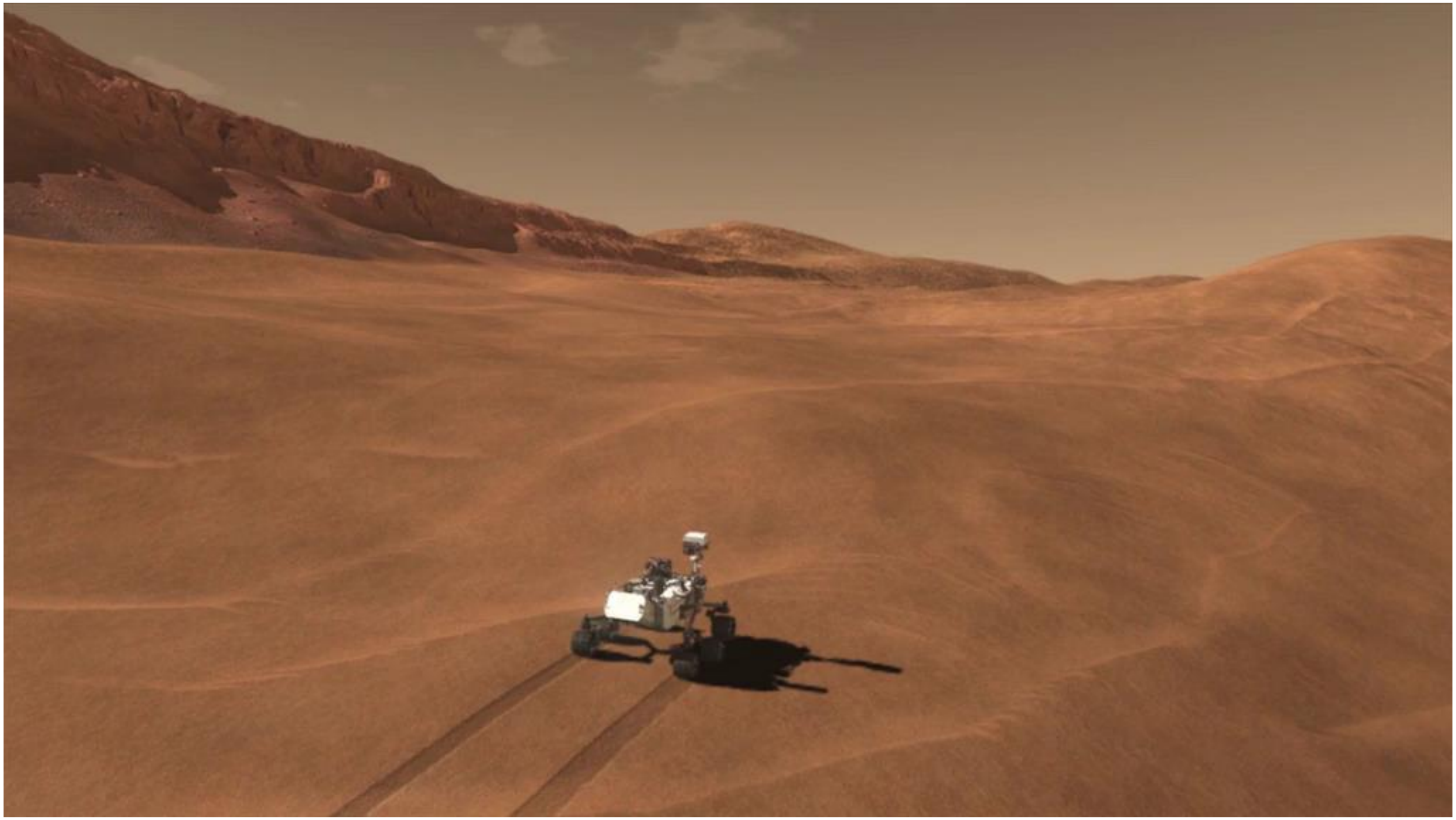
Unitree (China), 2021



Agility Robotics (U.S.A), 2019

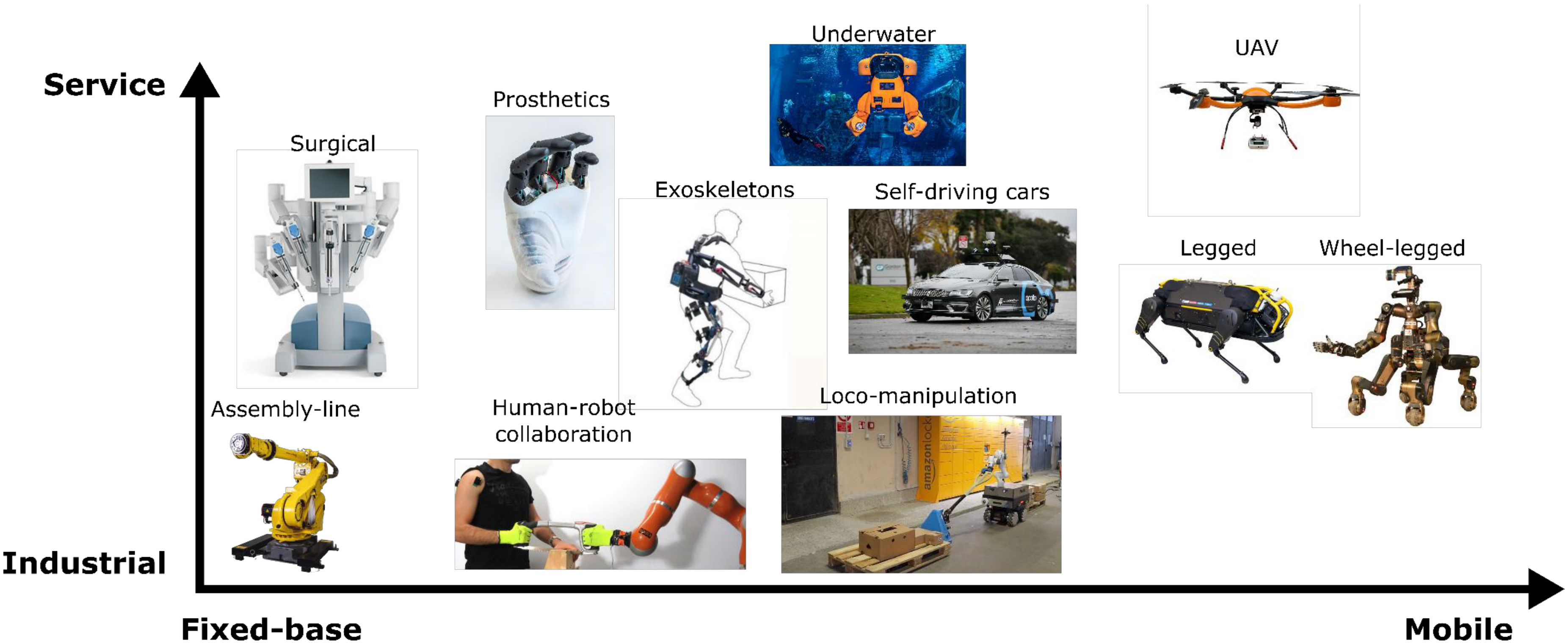
20th and 21st Century Robotics

On February 18th, 2021, the Perseverance rover landed on Mars



Animation of Perseverance soil sample return (JPL, 2020)

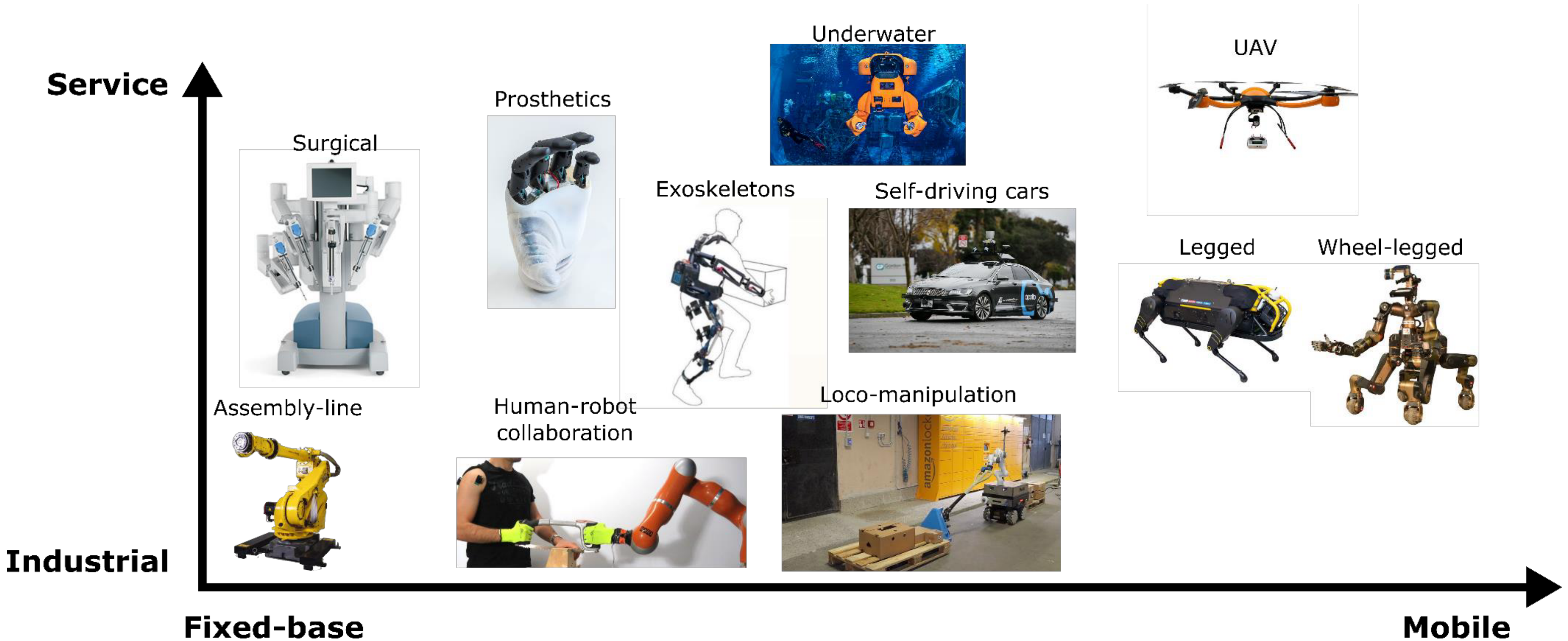
Modern Robotics: a wide spectrum



What do they all have in common?

They all work based on concepts and principles discussed in this course

Modern Robotics: a wide spectrum

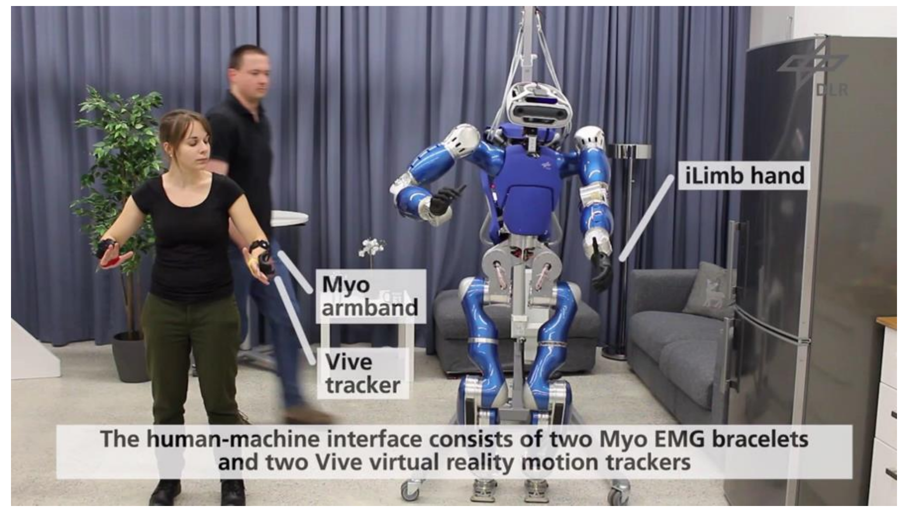


In this course we will focus on the use case of **robot manipulators**
However, the tools that you will learn here can be easily applied to other domains

State of the art - manipulation

- Shared autonomy and bimanual manipulation

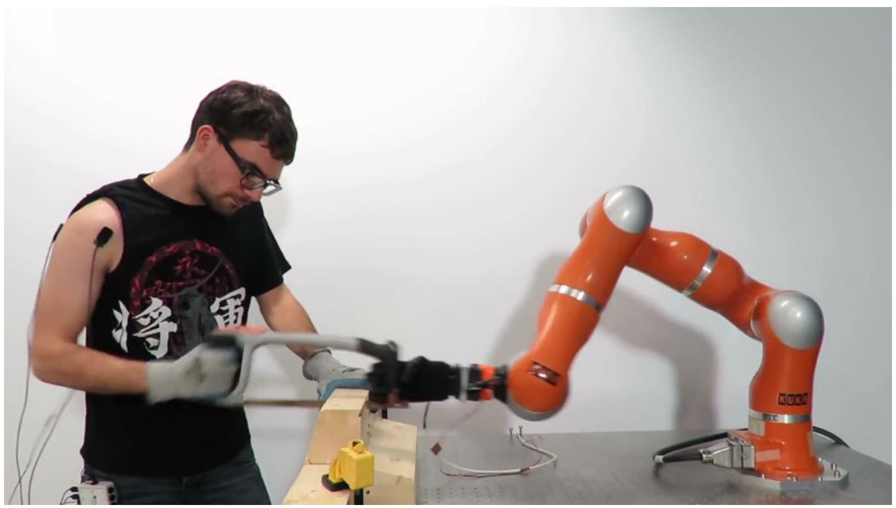
IIT, U. di Pisa, 2018



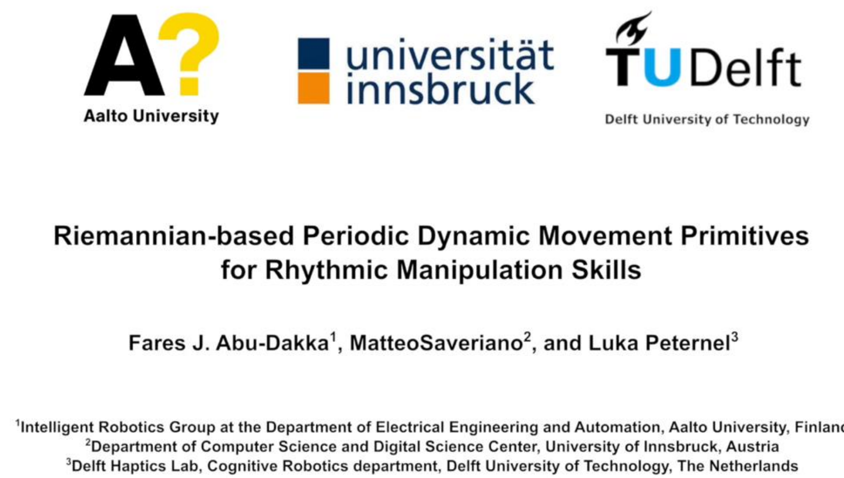
DLR, 2019

State of the art - manipulation

Human-robot collaboration



IIT, 2018



Riemannian-based Periodic Dynamic Movement Primitives for Rhythmic Manipulation Skills

Fares J. Abu-Dakka[1], MatteoSaveriano[2], and Luka Peternel[3]

[1]Intelligent Robotics Group at the Department of Electrical Engineering and Automation, Aalto University, Finland
[2]Department of Computer Science and Digital Science Center, University of Innsbruck, Austria
[3]Delft Haptics Lab, Cognitive Robotics department, Delft University of Technology, The Netherlands

Aalto, UIBK, TUD, 2021

Course notation

SLIDE NOTATION

$\oplus$ Advantage / pros

$\ominus$ Disadvantage / cons

$\triangleq$ Definition

[PRATT 18] local reference to paper

$\odot$ optional slide, for in-depth study, normally hidden during class, not part of the exam

MATH NOTATION

- $I_{3\times3}$: identity matrix 3×3
- $[a]_\times$: skew symmetric matrix (3×3) associated with the cross product $a\times$
- Frame : origin + orientation frame
- ${}_wP$: coordinate of point p wrt frame w
- ${}_wP_{A,B}$: position of point B wrt A expressed in W
- ${}_wI_i$: inertia of body i expressed in the frame W
- wrench : generic 6D force

Functional Units of a Robot

Outline

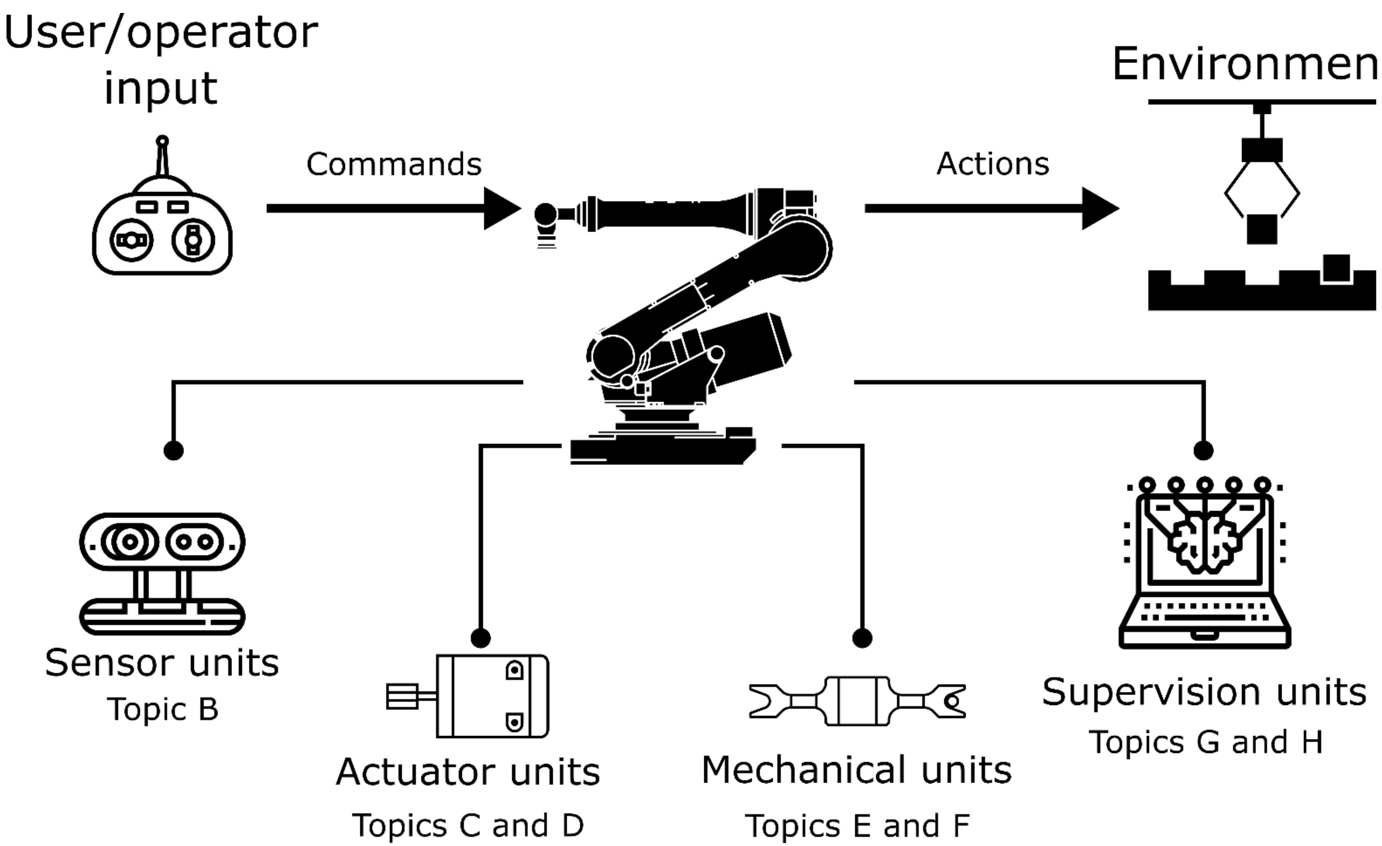
Overview of functional units of a robot

Classification of robots

Overview of the main topics in robotics

Robot as a system

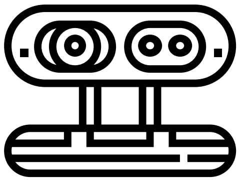
- Robotics is an interdisciplinary engineering field in between mechanics, control theory, electronics and computer science



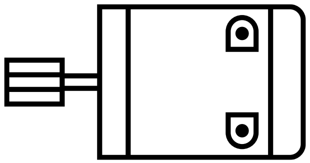
Images by Lluisa Iborra, Bakunetsu Kaito, Eucalyp, Pham Thanh Loc, Dompbel Studio, Rob Armes and Konkrit Noenpoempisut from https://thenounproject.com

Overview of functional units of a robot:

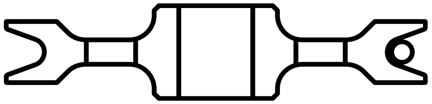
- Sensor units:
  - Proprioceptive: Internal robot state (joint position and velocity, acceleration, orientation)
  - Exteroceptive: interaction with the world around (force, vision, proximity, haptic)



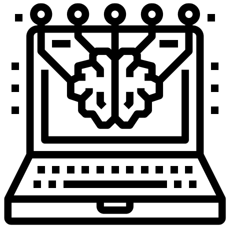
- Actuator units
  - Motors (electric, hydraulic, pneumatic)
  - Transmissions (gears, drives)
  - Actuator control (low-level)



- Mechanical units
  - Joints of kinematic chain
  - End-effector
  - Supporting structure (rigid links)



- Supervision units
  - Planning and (high-level) control
  - Artificial intelligence and cognition



Overview of functional units of a robot:

How does the robot solve its tasks?

- What to do? **Control Objective** and **feedback control laws**
- How to do it? Software **implementation** (Computers, µC, embedded)

How does the robot interact with the environment?

- **Autonomy** is achieved by a combination of **perception** and **actions**
- Actions are strictly determined by the control law and performed via **actuators**
- Perception is gathered from the surrounding environment using **sensors**

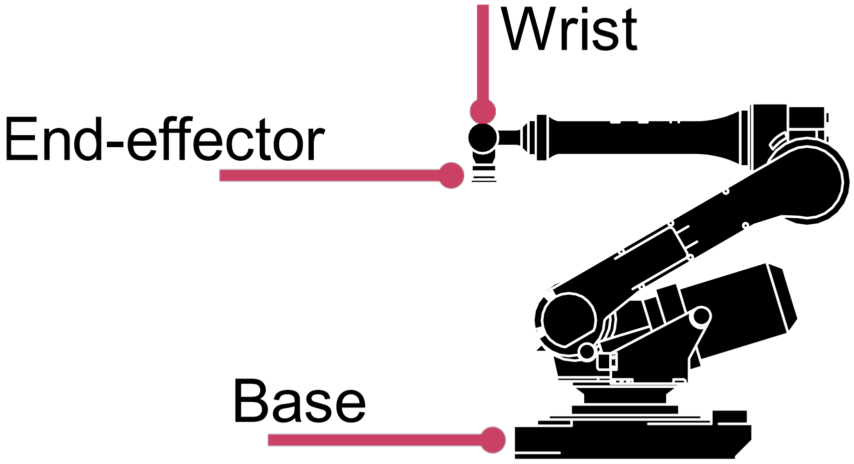
Mechanical Structure

- The mechanical structure of a manipulator consists of a sequence of rigid bodies (links) connected by means of either a **prismatic** or **revolute joints** that form a kinematic chain

- At one end of the chain we have the **base**, usually fixed to floor

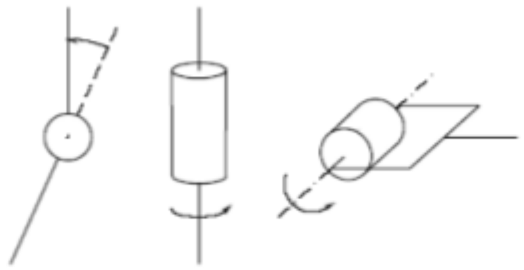
- The other end is the **end-effector** where a gripper or tool is mounted

- Degrees of freedom (**DoFs**) are distributed along the mechanical structure in a sufficient number to perform a task:
    - Three for positioning in 3D
    - Three (**wrist**) for orientation in 3D

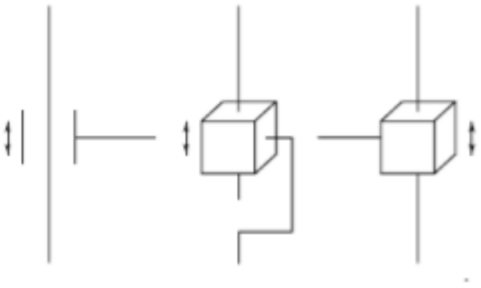


The joints

- Each joint allows for one (and only one) degree of freedom between two links
- We call joint variable the coordinate associated to such degree of freedom
- Symbols to represent joints in schematic diagrams:



ROTATIONAL JOINTS

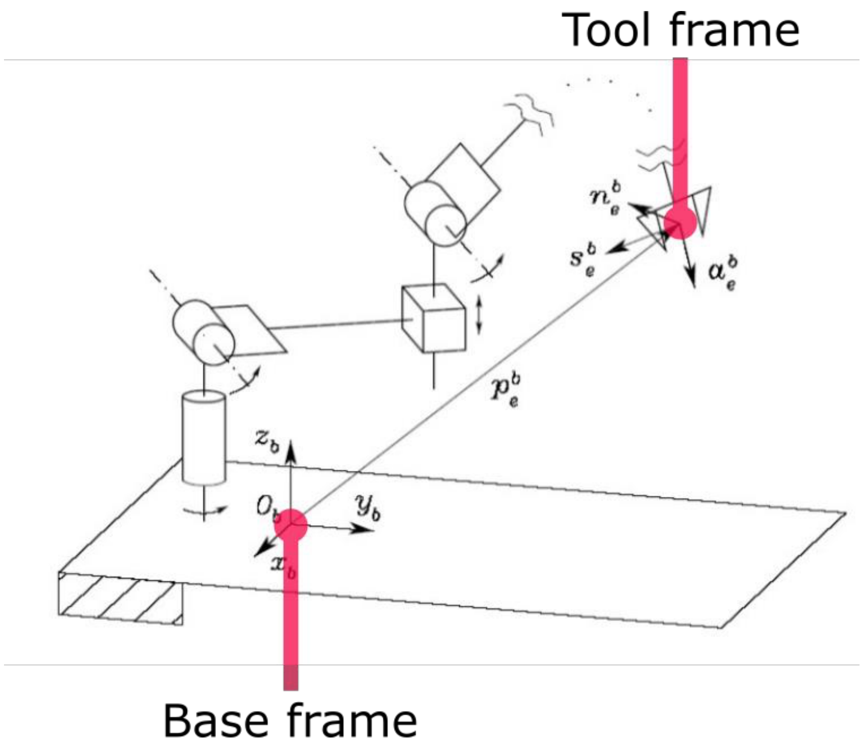


PRISMATIC JOINTS

Base frame and end-effector (tool) frame

- The **base frame** is attached to the first link of the kinematic chain (base)

- The **tool (end-effector) frame** is located at the central point of the gripper



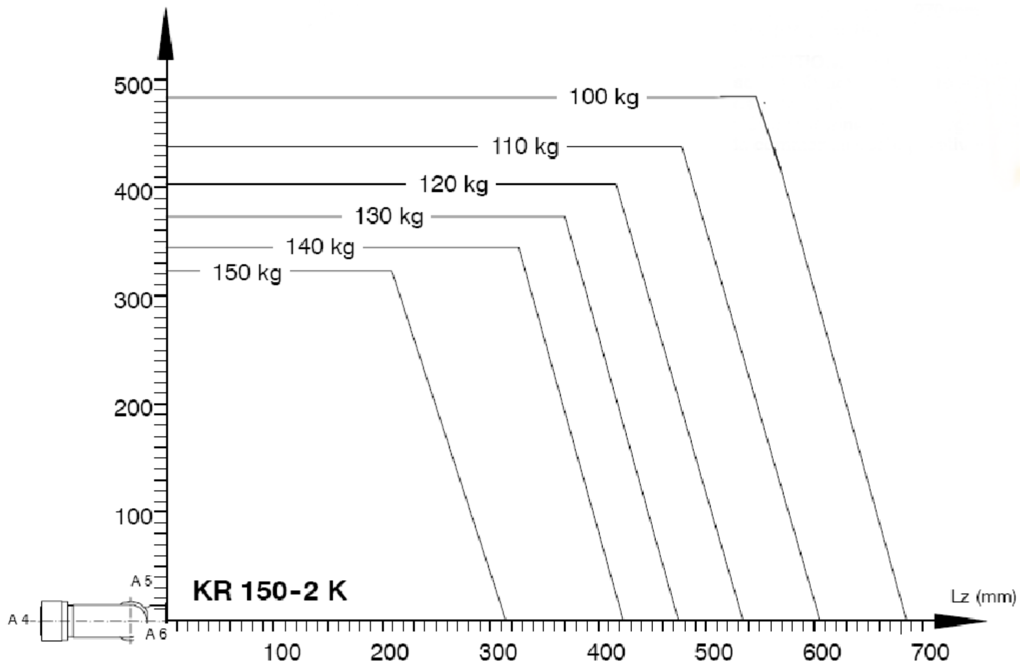
End-effector

- Mechanical part that effectively performs a task
- Specified according to the task that the robot has to execute
  - Material handling: **gripper**
  - Machining tasks: **tool** (e.g., mill, drill, welder, torch)

Payload

- The load that the robot can transport in a precise way

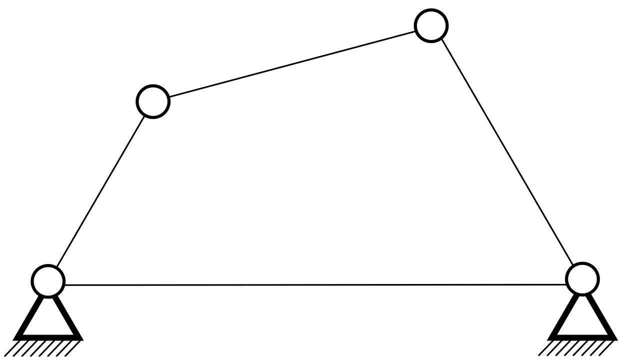
- Payload example graph for the KUKA KR 150.2K:



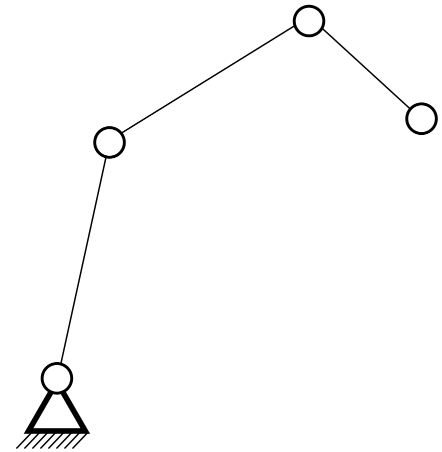
- Maximum payload of 150 kg at 310 mm

Basic kinematic structures

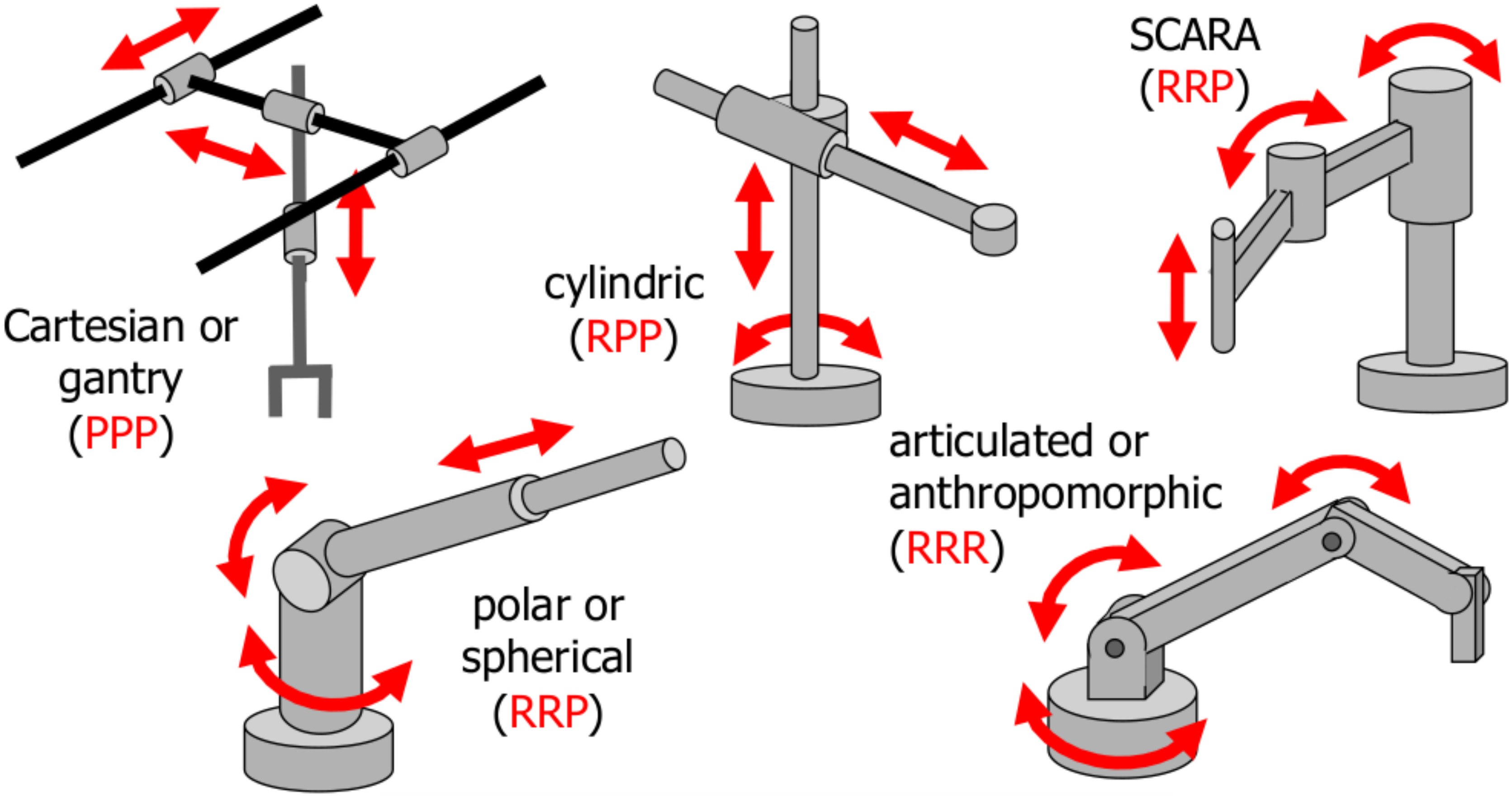
Closed kinematic chain:
the sequence of links forms a loop



Open (serial) kinematic chain:
the sequence of links are open at the end



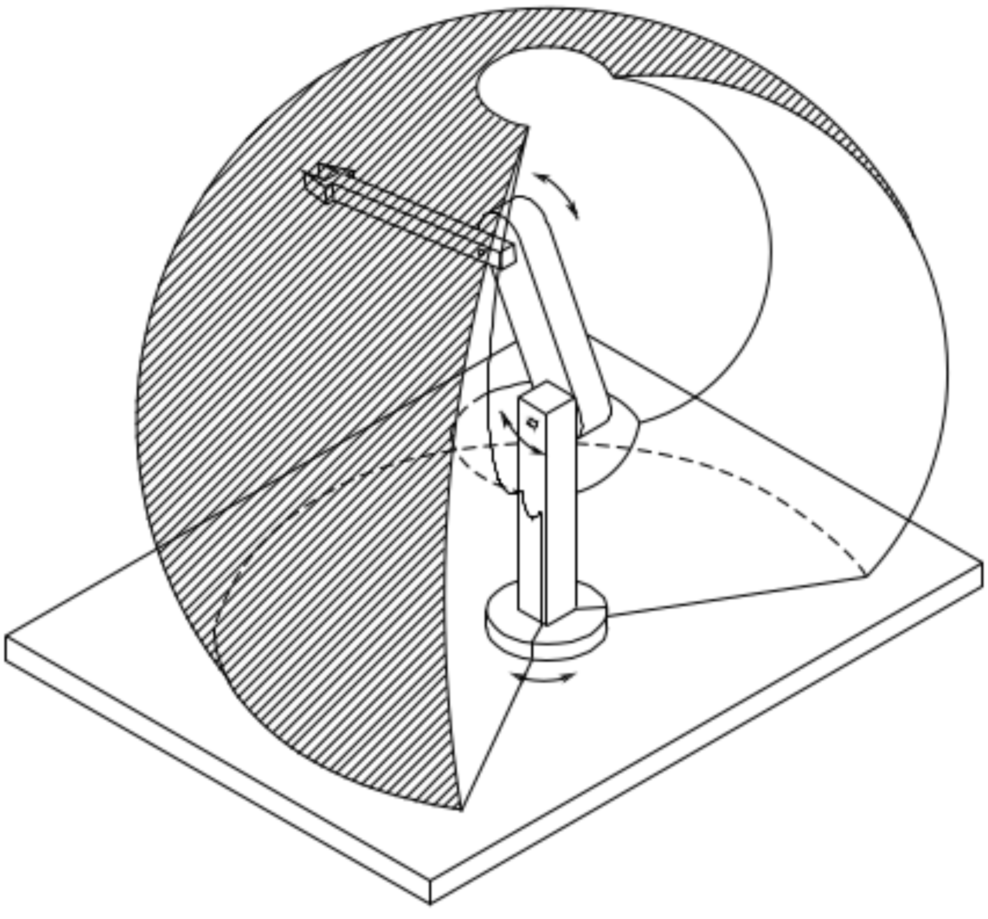
Classification by kinematic type (first 3 DoFs only)



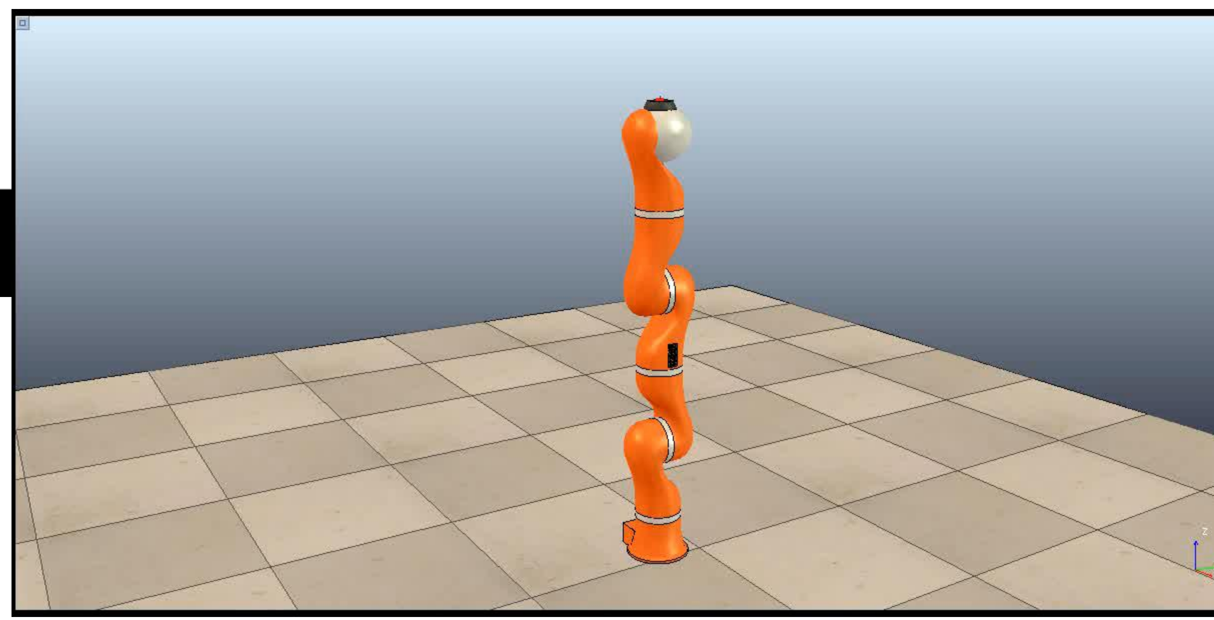
R = 1-dof rotational (revolute) joint
P = 1-dof translational (prismatic) joint

Workspace

- Portion of the environment that the robot has access to
- Its shape and volume depend on the manipulator structure and the presence of mechanical joint limits



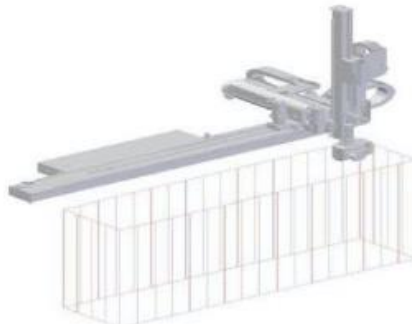
Workspace



Examples of manipulator types and their workspaces

Cartesian or gantry (PPP)

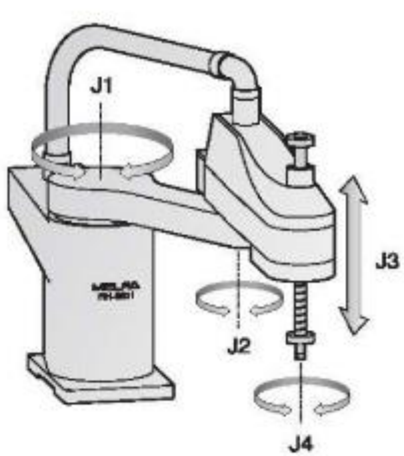
Cartesian Robot

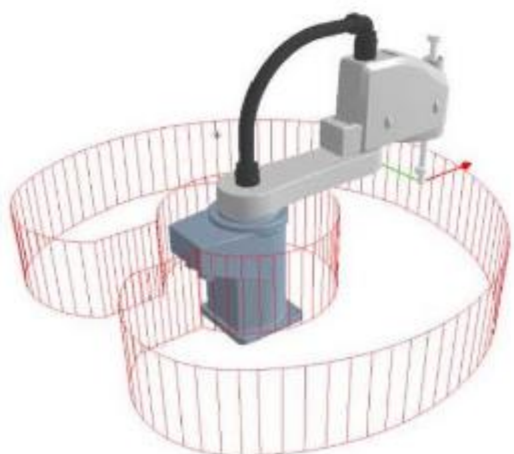


Accuracy: constant
Good stiffness

SCARA(RRP)

SCARA Robot



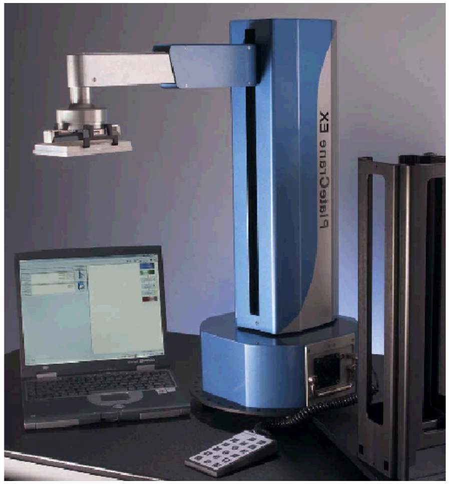


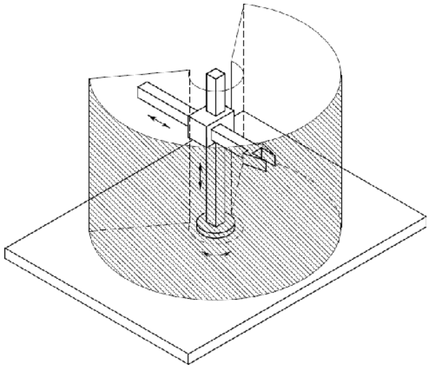
All joint axes are parallel

- Workspace: Cylindrical
- Accuracy: decreases with stroke
- Stiffness: high for vertical loads

Examples of manipulator types and their workspaces

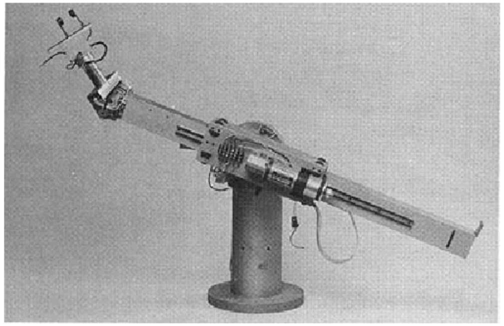
Cylindrical (RPP)

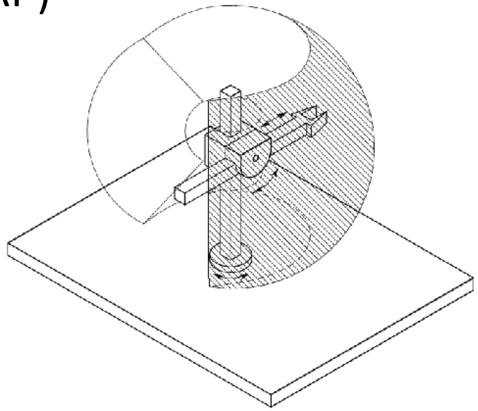




- Workspace: Cylinder

Spherical or Stanford arm (RRP)

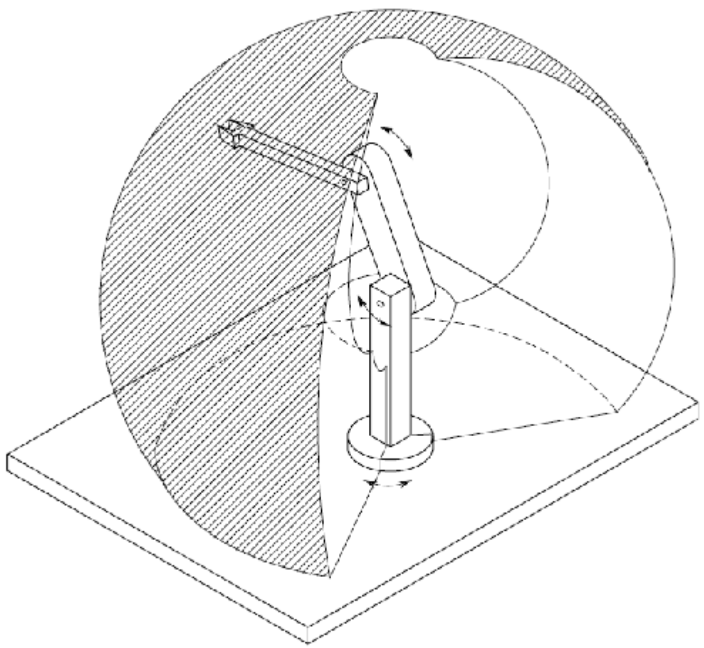




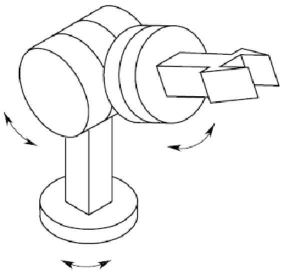
- Workspace: Sphere
- Accuracy: decreases with stroke
- Lower stiffness

Examples of manipulator types and their workspaces

- Anthropomorphic (RRR): KUKA KR-60



- Spherical wrist (axis intersecting at a single point) is chosen to ensure highest dexterity



60% of installed robot manipulators have anthropomorphic structure

- Workspace: large portion of Sphere
- Accuracy: varies inside workspace
- Lower stiffness

Example of industrial robots

ABB

DAIHEN

EPSON

FANUC

KUKA

NAICHI

Overview of the main robotics topics: control, perception, estimation, planning

Sensors (position, force, tactile, cameras, IMU, kinect, etc...)

Hundreds of times a second

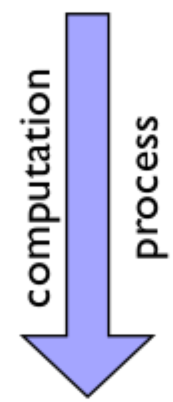
Actuation (electric motors, hydraulic or pneumatic)

Estimation (e.g. what is the state of the robot)

Perception (e.g. vision to understand the environment)

Control (e.g. how to move each limb, how to apply forces)

Planning (e.g. how to climb up a mountain)



Estimation

- Not all the states of a robot can be directly measured through sensors
- State estimation refers to the branch of control theory that deals with the problem of providing estimates of the internal states of a system based on measured variables and the system inputs (e.g., using observers)
- State estimation is key for a robotic system since it provides the values for the states used for control

Perception

- Ability to perceive, comprehend, and reason about the surrounding environment, handling the information coming through sensors
- It can involve both proprioceptive (e.g., joint position and velocity) and exteroceptive (e.g., vision, laser) sensors
- A challenge related to floating-base robots is the so-called "Simultaneous Localization and Mapping" (SLAM) problem

Agile Flight in Dynamic Environments, UZH, 2018

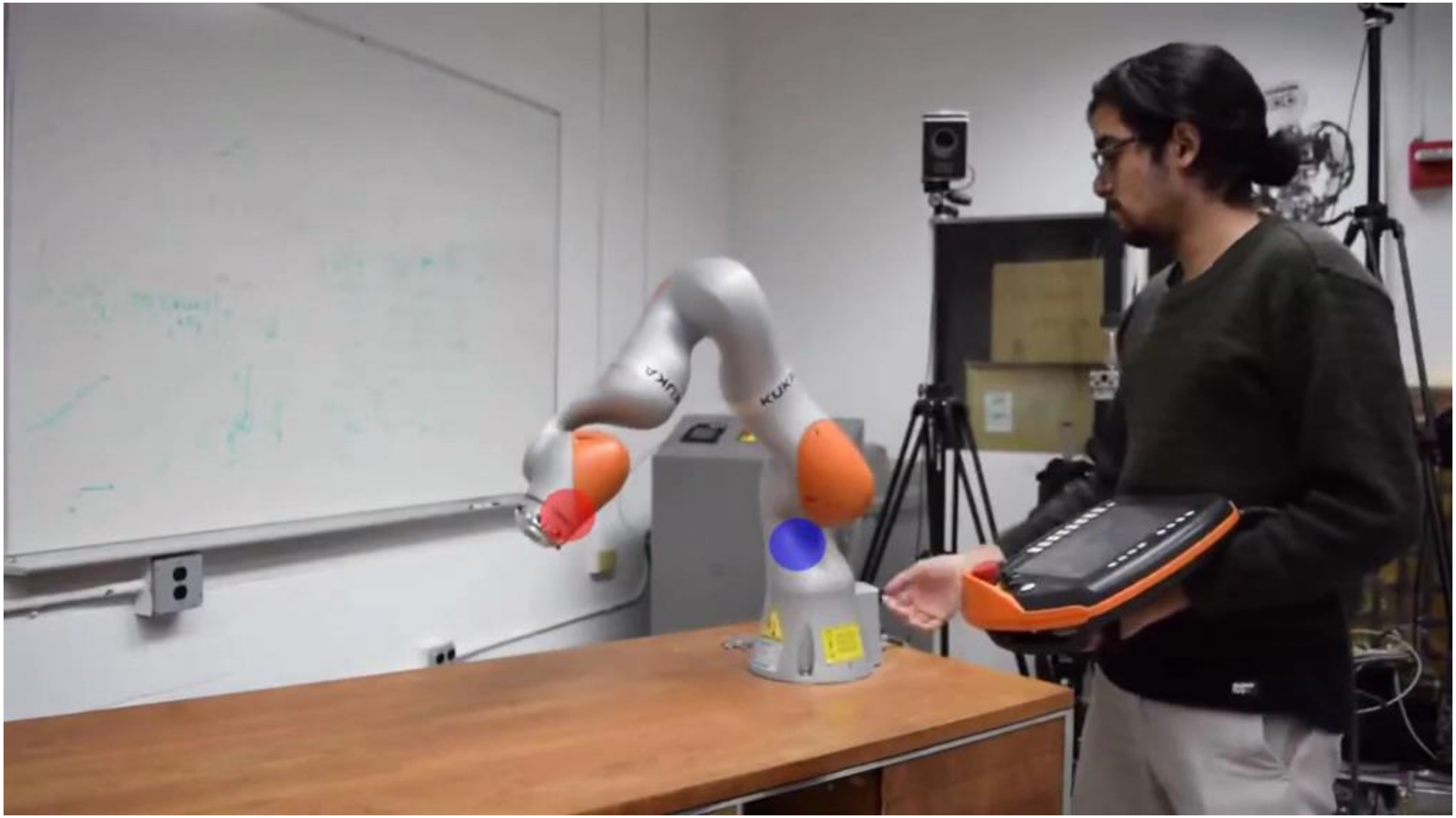
Control

- To execute a desired motion (e.g., change a joint’s position), we need to design a **controller**

- A controller is a mathematical law (generally based on the concept of **feedback**) that aims to drive a system to a desired set of states

- We can enumerate three main goals for a controller
    1. (Asymptotic) stability: The state variables of the system do not diverge exponentially under disturbances or changes in the system.
    2. Reference tracking: by computing the deviation between a desired behavior and the measured output of the system, a feedback control system can follow said behavior up to a specific **accuracy**
    3. Robustness: A system is said to be robust if it can remain stable even under presence of uncertainty (model discrepancies, sensor noise, motion coupling between joints

Control



Control of a manipulator using MPC, NYU,2020

Planning

- Planning refers to the problem of finding a sequence of valid actions to execute a specific task

- Depending on the task this general definition poses different challenges, e.g.:

  - In material handling tasks it is sufficient to assign pick-up and release locations of an object (point-to-point motion)

  - In machining tasks, the end-effector has to follow a desired trajectory, also requiring timing laws for the relevant variables (joints or end-effector indifferently)

  - Remotely operated robots using a physical or software interface (such as joystick or a GUI)

  - reactive motion planning, deals with both planning a trajectory and adjusting it in case of changes in the environment such as moving objects.

Reactive planning

Real time perception and reactive motion planning, MPI,2020

https://youtu.be/R9gZF6ihPSk

Robot model

Often we need a (mathematical) model to predict the robot's behaviour, based on the laws that govern the motion of the mechanical structure

- All of the previous topics require the use of **models**

    - A model to design **state estimation** algorithms
    - Sensor models to do **sensor data processing** and develop **perception** algorithms
    - Robot models (e.g., kinematics and dynamics) to design **control laws** and **planning algorithms**

- A model is used to predict the system's behaviour when performing a task

- Nowadays, a commonly used format to describe robots is the so-called Universal Robot Description Format (URDF)

- The URDF is a domain specific file format based on XML that describes:
    - The body layout of the robot (kinematic and collisions surfaces)
    - Visual appearance
    - Information about joint position and velocity limits
    - It can also include sensor models through (e.g. cameras, LiDar) plugins for specific simulators (e.g. Gazebo)

Software used in robotics

There are different types of software used during development in robotics. Some of the most common ones and some examples of them are:

- Middleware : Software designed to manage multiple technologies, provides a common programming abstraction across a distributed system, communication with messages. Some examples are:
    - Robot Operating System (ROS), widely used and accepted
    - Yet Another Robot Platform (YARP), developed by IIT
    - OROCOS: real-time compliant, more focused on control

- Simulators: Software designed to recreate the physical behavior of robot and its interaction with the environment
    - Gazebo, the main simulator used by ROS
    - PyBullet, widely used in Machine Learning and based on the Bullet physics engine
    - NVIDIA Omniverse, GPU powered physics simulation with photorealistic quality

- Rigid body kinematics and dynamics libraries: Software designed to compute the algorithms that provide the kinematic and dynamic states of a robot with rigid links
    - Rigid Body Dynamics Library (RBDL), most used in the past
    - Robotics Code Generator (RobCoGen), optimized code based on auto-diff
    - Pinocchio, very recent and becoming widely adopted