Introduction to Robotics

Jan Faigl

Department of Computer Science
Faculty of Electrical Engineering
Czech Technical University in Prague

Lecture 01

B4M36UIR – Artificial Intelligence in Robotics

Overview of the Lecture

- Part 1 – Course Organization
  - Course Goals
  - Means of Achieving the Course Goals
  - Evaluation and Exam
- Part 2 – Introduction to Robotics
  - Robots and Robotics
  - Challenges in Robotics
  - What is a Robot?
  - Locomotion

Part I

Part 1 – Course Organization

Course and Lecturers

B4M36UIR – Artificial Intelligence in Robotics

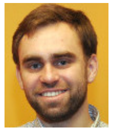
- https://cw.fel.cvut.cz/wiki/courses/b4m36uir/
- Department of Computer Science – http://cs.fel.cvut.cz
- Artificial Intelligence Center (AIC) – http://aic.fel.cvut.cz
- Lecturers

  doc. Ing. **Jan Faigl**, Ph.D.

  Ing. **Tomáš Krajník**, Ph.D.

  - Center for Robotics and Autonomous Systems (CRAS) http://robotics.fel.cvut.cz
  - Computational Robotics Laboratory (ComRob) http://comrob.fel.cvut.cz
- Lab supervisor

  Ing. **Miloš Prágr**



Course Goals

- Master (yourself) with applying AI methods in robotic tasks

  Labs, homeworks, projects, and exam

- Become familiar with the notion of intelligent robotics and autonomous systems
- Acquire knowledge of robotic data collection planning
- Acquire experience on combining approaches in autonomous robot control programs

  Integration of existing algorithms (implementation) in mission planning software and robot control program

- Experience solution of robotic problems

  Your own experience!

Course Organization and Evaluation

- B4M36UIR and BE4M36UIR – Artificial intelligence in robotics
- Extent of teaching: 2(lec)+2(lab);
- Completion: Z,ZK; Credits: 6;

  Z – ungraded assessment, ZK – exam

- Ongoing work during the semester – labs' tasks, homeworks, and semestral projects

  Be able to independently work with the computer in the lab (class room)

- Exam test

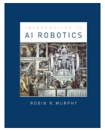
- Attendance to labs and successful evaluation of homeworks and semester projects

Resources and Literature

- **Textbooks**

  - Introduction to AI Robotics, *Robin R. Murphy* MIT Press, 2000

    First lectures for the background and context

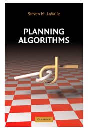


  - The Robotics Primer, *Maja J. Mataric*, MIT Press, 2007

    First lectures for the background and context

  - Planning Algorithms, *Steven M. LaValle*, Cambridge University Press, 2006

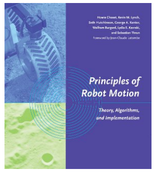
    `http://planning.cs.uiuc.edu`



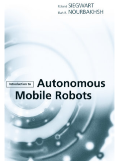
- **Lectures** – "comments" on the textbooks, slides, and **your notes**
- **Laboratory Exercises** – labs' tasks, homeworks, and projects, and projects
- **Selected research papers** – further specified during the course

Further Books 1/2

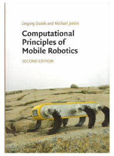
- Principles of Robot Motion: Theory, Algorithms, and Implementations, *H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun*, MIT Press, Boston, 2005



- Introduction to Autonomous Mobile Robots, 2nd Edition, *Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza*, MIT Press, 2011

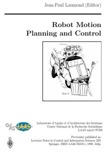


- Computational Principles of Mobile Robotics, *Gregory Dudek and Michael Jenkin*, Cambridge University Pres, 2010

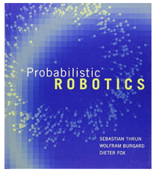


Further Books 2/2

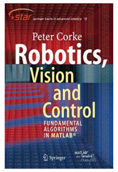
- Robot Motion Planning and Control, *Jean-Paul Laumond*, Lectures Notes in Control and Information Sciences, 2009
  http://homepages.laas.fr/jpl/book.html



- Probabilistic Robotics, *Sebastian Thrun, Wolfram Burgard, Dieter Fox*, MIT Press, 2005
  http://www.probabilistic-robotics.org/



- Robotics, Vision and Control: Fundamental Algorithms in MATLAB, *Peter Corke*, Springer, 2011
  http://www.petercorke.com/RVC1/



Lectures – Winter Semester (WS) Academic Year 2019/2020

- Schedule for the academic year 2019/2020
  http://www.fel.cvut.cz/en/education/calendar.html
- Lectures:
  - Karlovo náměstí, Room No. KN:E-126, Monday, 9:15–10:45
- 14 teaching weeks

13 lectures

Teachers

- Ing. **Miloš Prágr**
  Lab supervisor

- Ing. **Jan Bayer**
  Mobile robot exploration

- Ing. **David Milec**
  Game theory

- Ing. **Pert Váňa**
  Multi-goal planning

- Ing. **Robert Pěnička**
  Multi-goal planning

Communicating Any Issue Related to the Course

- Ask the lab teacher or the lecturer
- Use e-mail for communication
  - Use your faculty e-mail
  - Put UIR or B4M36UIR, BE4M36UIR to the subject of your message
  - Send copy (Cc) to lecturer/teacher or
    `uir-teachers at fel dot cvut dot cz`

Computers and Development Tools

- Network boot with home directories (NFS v4)

  Data transfer and file synchronizations – ownCloud, SSH, FTP, USB
- Python or/and C/C++ (**gcc** or **clang**)
- V-REP robotic simulator

  `http://www.coppeliarobotics.com/`
- Open Motion Planning Library (OMPL)

  `http://ompl.kavrakilab.org/`
- Robot Operating System (ROS)

  `http://www.ros.org/`
- Sources and libraries provided by **Computational Robotics Laboratory**, **Game Theory** group, and **Multi-Robot Systems** group.
- Any other open source libraries
- Gitlab FEL – `https://gitlab.fel.cvut.cz/`
- FEL Google Account – access to `Google Apps for Education`

  See `http://google-apps.fel.cvut.cz/`
- Information resources (IEEE Xplore, ACM, Science Direct, Springer Link)
  - *IEEE Robotics and Automation Letters (RA-L), IEEE Transactions on Robotics (T-RO), International Journal of Robotics Research (IJRR), Journal of Field Robotics (JFR), Robotics and Autonomous Robots (RAS), Autonomous Robots (AuRo), etc.*

Tasks – Labs, Homeworks, and Projects

- Several task assignments during the labs that are expected to be solved partially during the labs, but most likely as homeworks using

  BRUTE – `https://cw.felk.cvut.cz/upload`
- Mandatory homeworks (**45 pts**) organized in four thematic topics
  - **Autonomous robotic information gathering** (**14 pts**)

    Exploration – robot control, sensing, and mapping
  - **Multi-goal planning** (**10 pts**)
  - **Randomized sampling-based planning** (**6 pts**)
  - **Game theory in robotics** (**15 pts**)
- One bonus task on **Incremental Path Planning** (**5 pts**)
- Four projects can be scored (**40 pts**)

  One for each individual thematic topic
  - Focus on integration of the tasks into complete ROS (Robot Operating System) application – `https://www.ros.org/`

Tasks – Labs and Homeworks

- **Autonomous robotic information gathering** (**14** points)
  - `T1a-control` (**3 points**) – Open-loop robot motion control
  - `T1b-reactive` (**3 points**) – Reactive obstacle avoidance
  - `T1c-map` (**2 points**) – Map building (*map building of sensory perception*)
  - `T1d-plan` (**3 points**) – Grid based path planning
  - `T1e-expl` (**3 points**) – Mobile robot exploration

    robotic information gathering
- Bonus `T1-bonus` (**5 points**) – Incremental path planning (D* Lite)
- **Multi-goal path planning** (MTP) – TSP-like problem formulations (**10** points)
  - `T2a-tspn` (**5 points**) – Traveling Salesman Problem with Neighborhood (TSPN)
  - `T2b-dtspn` (**5 points**) – Curvature-constrained MTP – Dubins TSPN
- **Randomized sampling-based planning** (**6** points)
  - `T3a-sampl` (**3 points**) – Randomized sampling-based motion planning using PRM
  - `T3b-rrt` (**3 points**) – Curvature-constrained local planning in RRT
- **Game theory in robotics** (**15** points)
  - `T4a` (**3 points**) – Greedy policy in pursuit-evasion
  - `T4b` (**6 points**) – Monte Carlo Tree Search policy in pursuit-evasion
  - `T4c` (**6 points**) – Value-iteration policy in pursuit-evasion
- All tasks must be submitted to award the ungraded assessment
- Late submission will be penalized!
- The minimal scoring from homeworks is 25 points

Tasks – Projects

- `P1-expl` - **Autonomous robotic information gathering** (**15** points)
  - Implement full exploration pipeline using ROS.org and V-REP simulator
- `P2-data` - **Multi-goal path planning** (**10** points)
  - Implement full surveillance mission planning for UAV with plan execution using ROS.org and Gazebo simulator
  - Using full deployment pipeline of Multi-robot Systems (MRS) group
- `P3-motion` - **Randomized sampling-based planning** (**5** points)
  - Implement (utilize) asymptotically optimal randomized sampling-based path planning using OMPL and ROS.org
- `P4-gt` - **Game theory in robotics** (**10** points)
  - Implement complete deployment pipeline for patrolling polygonal environment using designed patrolling strategy, ROS.org, and V-REP
- Minimal required scoring from the projects is **15 points**!
  - It can be achieved by P1, but it must be perfect!
- There is a common deadline for the projects

  05.01.2020 23:59 CET

Course Evaluation

Points	Maximum Points	Required Minimum Points
Homeworks	45	25
Bonus Homework	5	0
Projects	40	15
Exam test	20	10
Total	110 points	50

- All homeworks have to be submitted
- 40 points from the semester are required for awarding ungraded assessment
- The course can be passed with **ungraded assessment** and **exam**
- All homeworks must be submitted and pass the evaluation

Grading Scale

Grade	Points	Mark	Evaluation
A	≥ 90	1	Excellent
B	80–89	1,5	Very Good
C	70–79	2	Good
D	60–69	2,5	Satisfactory
E	50–59	3	Sufficient
F	<50	4	Fail

Overview of the Lectures

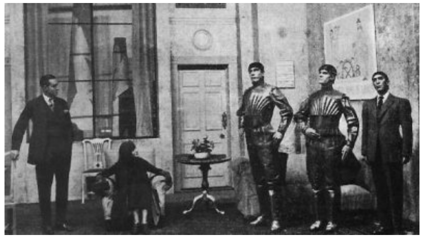
1. Course information, Introduction to (AI) robotics
2. Robotic paradigms and control architectures
3. Path and motion planning
4. Grid and graph based methods
5. Robotic information gathering - exploration of unknown environment and multi-goal planning (robotic TSP)

   Public holiday - Czech Independence Day
6. Data collection planning - TSP(N), PC-TSP(N), and OP(N)
7. Data collection planning with curvature-constrained vehicles
8. Randomized sampling-based motion planning methods
9. Game theory in robotics
10. Visibility based pursuit evaluation games (Game theory in robotics)
11. Patrolling games (Game theory in robotics)
12. Multi-robot planning
13. Long-term navigation and spatio-temporal mapping

Part II

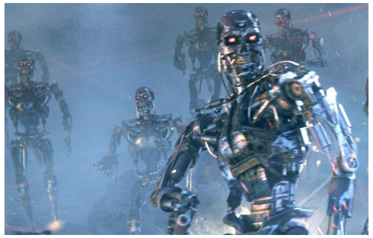
Part 2 – Introduction to Robotics

What is Understood as Robot?

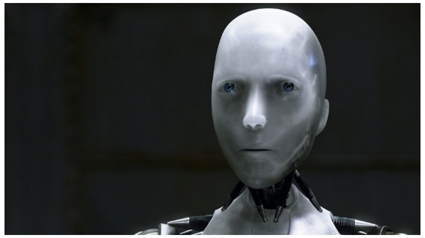


Rossum's Universal Robots (R.U.R)

Industrial robots



Cyberdyne T-800



NS-5 (Sonny)

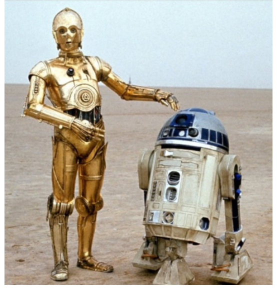
Artificial Intelligence (AI) is probably most typically understand as an intelligent robot

Intelligent Robots

- React to the environment – sensing
- Adapt to the current conditions
- Make decision and new goals

  E.g., in robotic exploration
- Even though they are autonomous systems, the behaviour is relatively well defined
- Adaptation and ability to solve complex problems are implemented as algorithms and techniques of **Artificial Intelligence**

  In addition to mechanical and electronical design, robot control, sensing, etc.



Stacionary vs Mobile Robots

- Robots can be categorized into two main groups

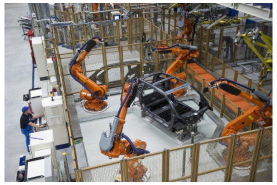
Stationary (industrial) robots

Mobile robots

- Stationary robots – defined (limited) working space
  - Even stationary robots need an efficient motion, and thus motion planning tasks can be a challenging problem
- Mobile robot – it can move, and therefore, it is necessary to address the problem of **navigation**

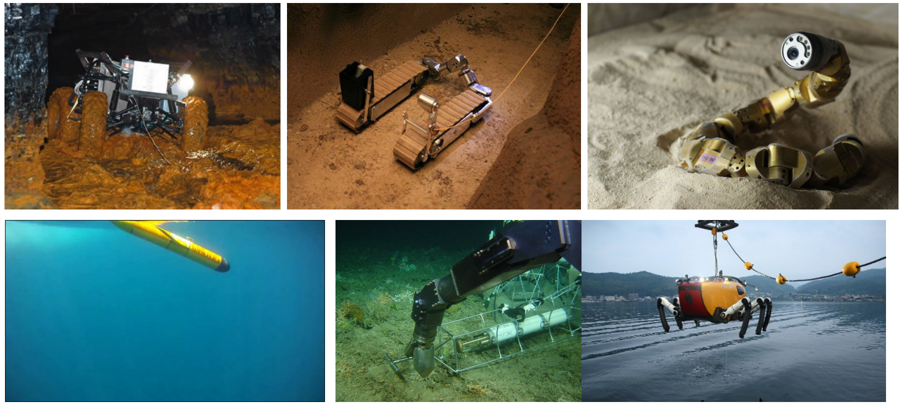
Stationary Robots

- Conventional robots needs separated and human inaccessible working space because of safety reasons
- Cooperating robots share the working space with humans



Types of Mobile Robots

- Regarding the environment: ground, underground, aerial, surface, and underwater vehicles
- Based on the locomotion: wheeled, tracked, legged, modular



Challenges in Robotics

- Autonomous vehicles – cars, delivery, etc.
- Consumable robots – toys, vacuum cleaner, lawn mover, pool cleaner
- Robotic companions
- Search and rescue missions
- Extraterrestrial exploration
- Robotic surgery
- Multi-robot coordination

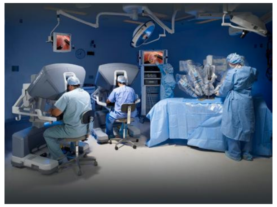
In addition to other technological challenges, new efficient AI algorithms have to be developed to address the nowadays and future challenges

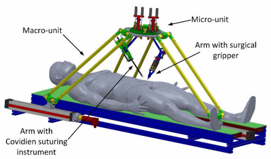
Robotic Surgery

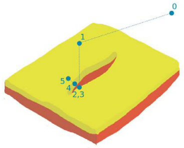
- Evolution of Laparoscopic Surgery

  Complex operations with shorter postoperative recovery
- Precise robotic manipulators and teleoperated surgical robotic systems
- Further step is automation of surgical procedures

  One of the main challenges is planning and navigation in tissue

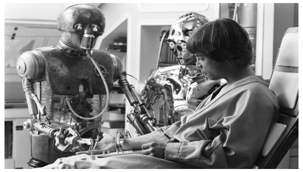



Macro-unit
Micro-unit
Arm with surgical gripper
Arm with Covidien suturing instrument



Tissue model

Robotic Arm of the Da Vinci Surgical System



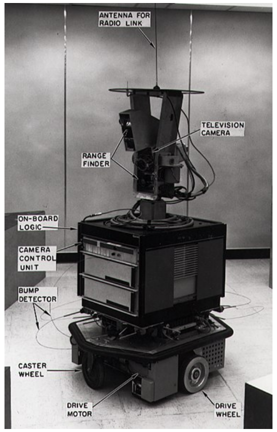
Surgical droid 2-1B

Artificial Intelligence and Robotics

- Artificial Intelligence (AI) field originates in 1956 with the summary that a intelligent machine needs:
  - Internal models of the world
  - Search through possible solutions
  - Planning and reasoning to solve problems
  - Symbolic representation of information
  - Hierarchical system organization
  - Sequential program execution

    M. Mataric, Robotic Primer
  - AI-inspired robot – Shakey

    Artificial Intelligence laboratory of Stanford Research Institute (1966–1972)



- Shakey – perception, geometrical map building, planning, and acting – early AI-inspired robot with purely deliberative control

Robotics in B4M36UIR

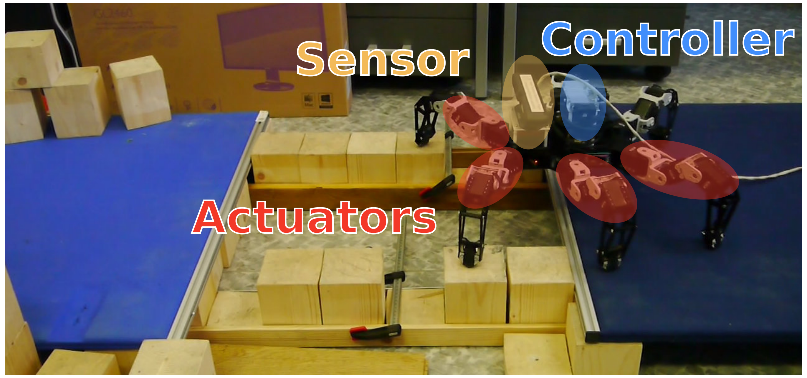
- Fundamental problems related to motion planning and mission planning with mobile robots
- The discussed motion planning methods are general and applicable also into other domains and different robotic platforms including stationary robotic arms
- **Robotics is interdisciplinary field**
  - Electrical, mechanical, control, and computer engineering
  - **Computer science** filds such as machine learning, artificial intelligence, computational intelligence, machine perception, etc.
  - Human-Robot interaction and cognitive robotics are also related to psychology, brain-robot interfaces to neuroscience, robotic surgery to medicine, etc.

In B4M36UIR, we will touch a small portion of the whole field, mostly related to motion planning and mission planning that can be "encapsulated" as ***robotic information gathering***

What is a Robot?

A robot is an autonomous system which exists in the physical world, can sense its environment, and can act on it to achieve some goals

- The robot has a physical body in the physical world – **embodiment**
- The robot has **sensors** and it can **sense**/**perceive** its environment
- A robot has effectors and actuators – it can **act** in the environment
- A robot has **controller** which enables it to be **autonomous**



Embodiment

- The robot body allows the robot to act in the physical world

  E.g., to go, to move objects, etc.
- Software agent is not a robot
- Embodied robot is under the same physical laws as other objects
  - Cannot change shape or size arbitrarily
  - It must use actuators to move
  - It needs energy
  - It takes some time to speed up and slow down
- Embodied robot has to be aware of other bodies in the world
  - Be aware of possible collisions
- The robot body influences how the robot can move

  Notice, faster robots look smarter

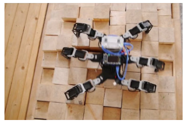
Sensing / Perception

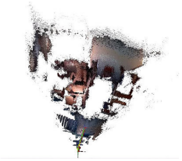
- Sensors are devices that enable a robot to perceive its physical environment to get information about itself and its surroundings
- **Exteroceptive** sensors and **proprioceptive** sensors
- Sensing allows the robot to know its **state**
- State can be **observable**, **partially observable**, or **unobservable**
  - State can be **discrete** (e.g., on/off, up/down, colors) or **continuous** (velocity)
  - **State space** consists of all possible states in which the system can be
    - **space** refers to all possible values
  - **External state** – the state of the world as robot can sense it
  - **Internal state** – the state of the robot as the robot can perceive it

    E.g., remaining battery

Sensors

- **Proprioceptive sensors** – measure internal state, e.g., encoders, inclinometer, inertial navigation systems (INS), compass, but also Global Navigation Satellite System (GNSS), e.g., GPS, GLONASS, Galileo, BeiDou
- **Exteroceptive (proximity) sensors** – measure objects relative to the robot
  - **Contact sensors** – e.g., mechanical switches, physical contact sensors that measure the interaction forces and torques, tactile sensors etc.
  - **Range sensors** – measure the distance to objects, e.g., sonars, lasers, IR, RF, time-of-flight
  - **Vision sensors** – complex sensing process that involves extraction, characterization, and information interpretation from images



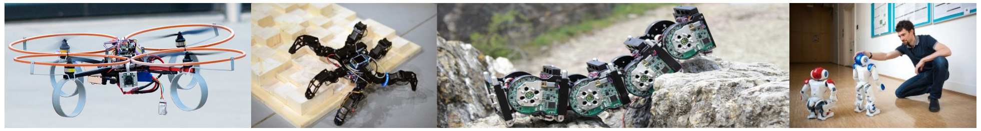


Action

- **Effectors** enable a robot to take an action
  - They use underlying mechanisms such as muscles and motors called **actuators**
- Effectors and actuators provide two main types of activities
  - **Locomotion** – moving around

    Mobile robotics – robots that move around
  - **Manipulation** – handling objects

    Robotic arms
- Locomotion mechanisms – wheels, legs, modular robots, but also propellers etc.

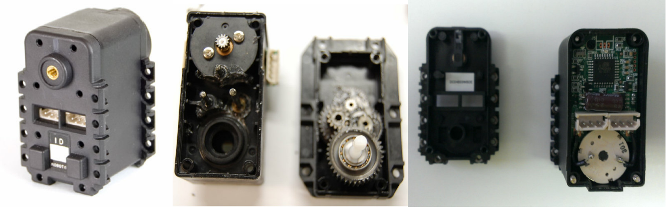


With more and more complex robots, a separation between mobile and manipulator robots is less strict and robots combine mobility and manipulation

Effectors and Actuators

- **Effector** – any device on a robot that has an effect on the environment
- **Actuator** – a mechanism that allows the effector to execute an action or movement, e.g., motors, pneumatics, chemically reactive materials, etc.
- Electric motors – Direct-Current (DC) motors, gears,
  - **Servo motors** – can turn their shaft to a specific position

    DC motor + gear reduction + position sensor + electronic circuit to control the motor

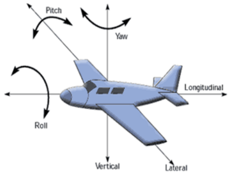


Hexapod with 3 servo motors (joints) per each leg has 18 servo motors in the total

Degrees of Freedom (DOF)

- **Degree of Freedom** (**DOF**) is the minimal required number of independent parameters to completely specify the motion of a mechanical system *It defines how the robot can move*

  In 3D space, a body has usually 6 DOF (by convention)
  - **Translational DOF** – x, y, z
  - **Rotational DOF** – *roll*, *pitch*, and *yaw*

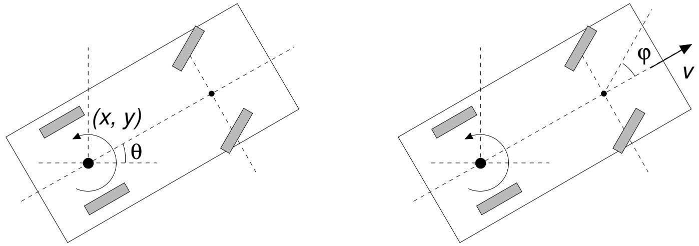


- **Controllable DOF** (**CDOF**) – the number of the DOF that are controllable, i.e., a robot has an actuator for such DOF

DOF vs CDOF

- If a vehicle moves on a surface, e.g., a car, it actually moves in 2D
- The body is at the position $(x, y) \in \mathbb{R}^2$ with an orientation $\theta \in \mathbb{S}^1$
- A car in a plane has DOF = 3, (x, y, θ) but CDOF=2, (v, φ)

Only forward/reverse direction and steering angle can be controlled



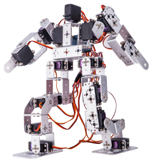
That is why a parallel parking is difficult

- A car cannot move in an arbitrary direction, but 2 CDOF can get car to any position and orientation in 2D
- To get to a position, the car follows a **continuous trajectory** (**path**), but with **discontinuous velocity**

Uncontrollable DOF makes the movement more complicated

Ratio of CDOF to the Total DOF

- The ratio of Controllable DOF (CDOF) to the Total DOF (TDOF) represents how easy is to control the robot movement
- **Holonomic** (CDOF=TDOF, the ratio is 1) – holonomic robot can control all of its DOF
- **Nonholonomic** (CDOF<TDOF, the ratio < 1) – a nonholonomic robot has more DOF that it can control *E.g., a car*
- **Redundant** (CDOF>TDOF, the ratio > 1) – a redundant robot has more ways of control



17 CDOF

6 DOF Hexapod

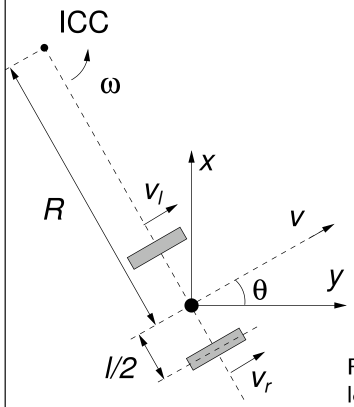
24 TDOF, 18 CDOF Hexapod walking robot

Locomotion

- **Locomotion** refers how the robot body moves from one location to another location *From the Latin Locus (place) and motion*
- The most typical effectors and actuators for ground robots are wheels and legs
- Most of the robots need to be **stable** to work properly
  - **Static stability** – a robot can stand, it can be static and stable
    Biped robots are not statically stable, more legs make it easier.
    Most of the wheeled robots are stable.
  - **Statically stable walking** – the robot is stable all the times *E.g., hexapod with tripod gait*
  - **Dynamic stability** – the body must actively balance or move to remain stable, the robots are called dynamically stable *E.g., inverse pendulum*

Locomotion – Wheel Robots

- One of the most simple wheeled robots is **differential drive** robot
  - It has two drived wheels on a common axis
  - It may use a castor wheel (or ball) for stability
  - It is nonholonomic robot *Omnidirectional robot is holonomic robot*

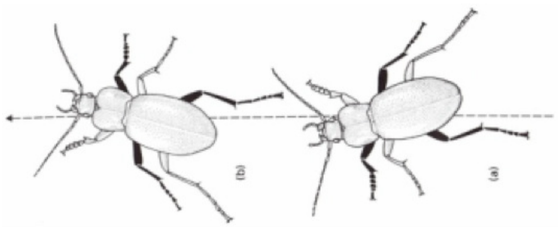


- v_l and v_r are velocities along the ground of the left and right wheels, respectively
- $\omega = \frac{v_r - v_l}{l}$, $R = \frac{l}{2}\frac{v_l + v_r}{v_r - v_l}$
- For $v_l = v_r$, the robot moves straight ahead *R is infinite*
- For $v_l = -v_r$, the robot rotates in a place *R is zero*
- Simple motion control can be realized in a turn-move like schema

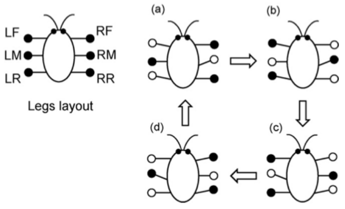
Further motion control using path following or trajectory following approaches with feedback controller based on the position of the robot to the path / trajectory

Locomotion – Legged Robots (Gaits)

- **Gait** is a way how a legged robot moves
- A gait defines the order how the individual legs lift and lower and also define how the foot tips are placed on the ground
- Properties of gaits are: stability, speed, energy efficiency, robustness (how the gait can recover from some failures), simplicity (how complex is to generate the gait)
- A typical gait for hexapod walking robot is **tripod** which is stable as at least three legs are on the ground all the times



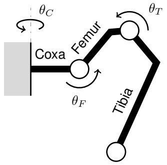
Gullan et al., The Insects: An outline of entomology, 2005

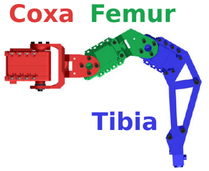


Iida et al. 2008

Locomotion of Hexapod Walking Robot

- Let have hexapod robot with six identical legs each with 3 DOF
- Each leg consists of three parts called **Coxa**, **Femur**, and **Tibia**



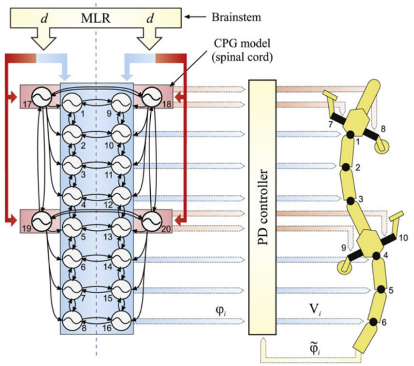


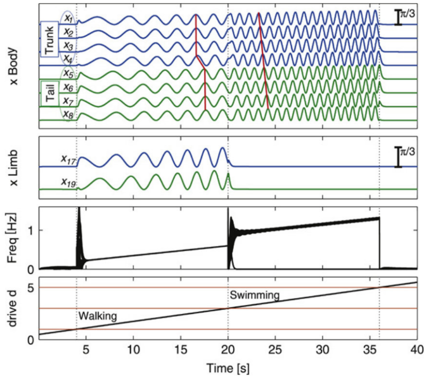
- The movement is a coordination of the **stance** and **swing** phases of the legs defined by the gait, e.g., tripod
- A **stride** is a combination of the leg movement with the foot tip on the ground (during the **stance phase**) and the leg movement in a particular direction (in the **swing phase**) within one **gait cycle**
- Various gaits can be created by different sequences of stance and swing phases
- T_{Stance}, T_{Swing}, and $T_{Stride} = T_{Stance} + T_{Swing}$ defines the **duty factor** $\beta = T_{Stance}/T_{Stride}$

Triod $\beta = 0.5$

Central Pattern Generator (CPG)

- **Central Pattern Generators** (**CPGs**) – are neural circuits to produce rhythmic patterns for various activities, i.e., locomotor rhythms to control a periodic movement of particular body parts
- Salamander CPG with 20 amplitude-controlled phase oscillators





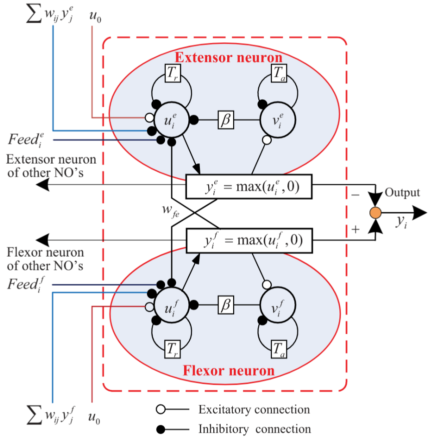
Auke Jan Ijspeert, Neural Networks, 2008

Example of Rhythmic Pattern Oscillator

- Matsuoka oscillator model based on biological concepts of the extensor and flexor muscles
- Van der Pol oscillator

$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x = 0$$

- The rhythmic patterns define the trajectory of the leg end point (foot tip)
- Joint angles can be computed from the foot tip coordinates using the **Inverse Kinematics**



Matsuoka, K. (1985). Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biological Cybernetics 52, 367––376

An example of simple CPG to control hexapod walking robot will be shown during the labs

Control Architectures

- A single control rule may provide simple robot behaviour

  Notice, controller can be feed-forward (open-loop) or feedback controller with vision based sensing
- Robots should do more than just avoiding obstacles
- The question is "How to combine multiple controllers together?"
- **Control architecture** is a set of guiding principles and constraints for organizing the robot control system
  - Guidelines to develop the robotic system to behave as desired

    It is not necessary to know control architectures for simple robotic demos and tasks. But it is highly desirable to be aware of architectures for complex robots

Summary of the Lecture

Topics Discussed

- Information about the Course
- Overview of robots, robotics, and challenges
  - Robot – Embodied software agent
  - Sensor, Controller, Actuators
  - Degrees of Freedom (DOF) and Controllable DOF
  - Mobile Robot Locomotion
  - Locomotion Gaits for Legged Robots
  - Central Pattern Generator
- Next: Robotic Paradigms and Control Architectures