Robotics

Erwin M. Bakker| LIACS Media Lab

6-2 2023

Universiteit Leiden

Bij ons leer je de wereld kennen

Organization and Overview

Lecturer:
Dr Erwin M. Bakker (erwin@liacs.nl)
Room 126a and LIACS Media Lab (LML)

Teaching assistant:
Mor Puigventos (email)
TBA (email)

Period: February 6th - May 22nd 2023
Time: Monday 15.15 - 17.00
Place (Rooms):
a) Gorlaeus - Lecture Hall C3
b) Sylvius - 15.31
c) Van Steenis - E0.04
d) Oortgebouw - Sitterzaal

Schedule (tentative, visit regularly):

Date	Room	Subject
6-2	a	Introduction and Overview
13-2	a	Locomotion and Inverse Kinematics
20-2	b	Robotics Sensors and Image Processing
27-2	a	SLAM + SLAM Workshop
6-3	c	Mobile Robot Challenge Introduction
13-3	a	Project Proposals I (by students)
20-3	d	Project Proposals II (by students)
27-3	d	Robotics Vision (Week 13, start 15.30)
3-4	d	Robotics Reinforcement Learning&Workshop
10-4		No Class (Eastern)
17-4	d	Project Progress I (by students)
24-4	d	Project Progress II (by students)
1-5	d	Mobile Robot Challenge I
8-5	a	Mobile Robot Challenge II
15-5	d	Project Demos I
22-5	d	Project Demos II
29-5		Whit Monday
5-6		Project Deliverables

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/

Grading (6 ECTS):
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, assignments (pass/no pass) 2 workshops (0-10) (2 x 20% = 40% of grade).
- It is necessary to be at every class and to complete every workshop and assignment.

Robotics in the News: Agility Robotics

Cassie sets a Guinness World Record. (No cameras or external sensors.)
Sept. 28 2022, https://agilityrobotics.com/news/2022/cassie-sets-a-guinness-world-record
https://agilityrobotics.com/news

Universiteit Leiden. Bij ons leer je de wereld kennen

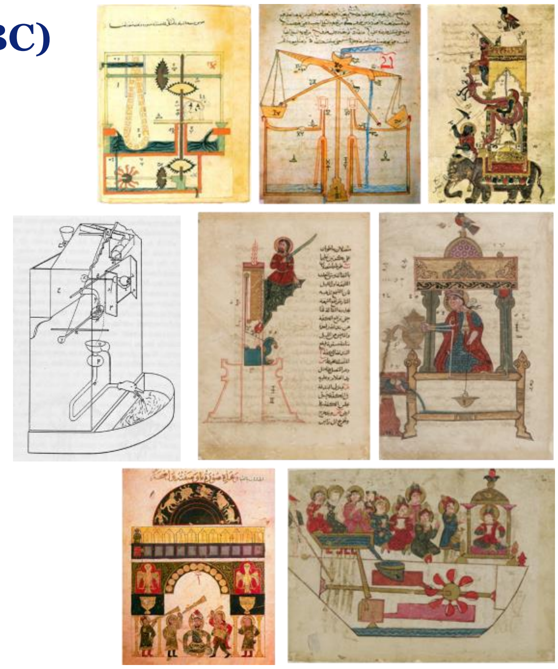
Robotics in the News: Agility Robotics

A Year of Agility Engineering.
Jan. 18 2022, https://www.youtube.com/watch?v=D8_VmWWRJgE

Universiteit Leiden. Bij ons leer je de wereld kennen

Philo of Byzantium (~280 – 220 BC)
Al-Jazari (1136 – 1206)

- Mechanisms and methods for automation
- Water-raising machines
- Clocks
- Automata
  - Drink-serving waitress
  - Hand-washing automaton with flush mechanism
  - Peacock fountain with automated servants
  - Musical robot band



Leonardo da Vinci (1452 – 1519)

- Robotic Carts
- Studies on locomotion
- Robotic Soldier
- Robotic Lion



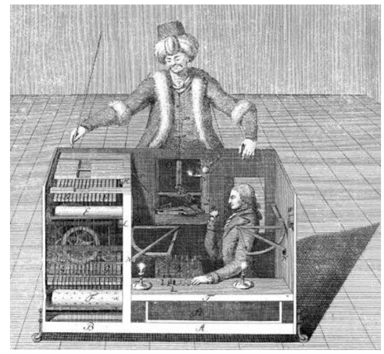


Pictures from:
http://www.leonardo3.net
http://brunelleschi.imss.fi.it

Universiteit Leiden. Bij ons leer je de wereld kennen

The Turk

Constructed and unveiled in 1770
by Wolfgang von Kempelen (1734–1804)



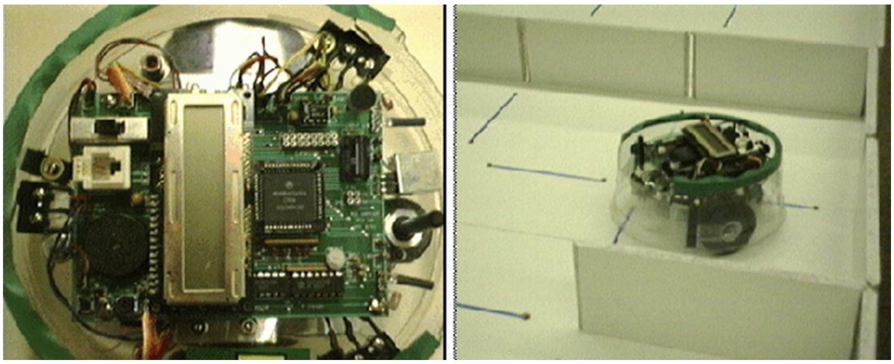
Pictures from:
http://en.wikipedia.org/wiki/The_Turk

LOCOMOTION & INVERSE KINEMATICS

EARLY ROBOTS

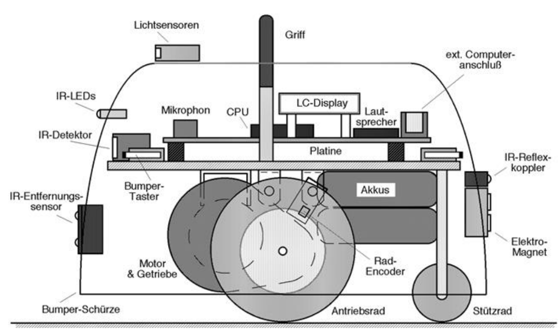
South Pointing Chariot by Ma Jun (circa 200–265)
was the first reliably documented version.

Autonomous Robots for Artificial Life (MIT, T. Braunl, Stuttgart University) 'Rug Warrior'

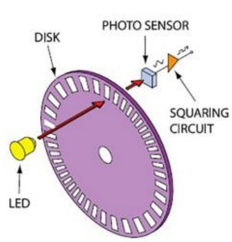


Autonomous Robots for Artificial Life

- Motorola MC68HC11 microcontroller
- Sensors
- Bumper
- Photoresistors (2)
- Infrared Obstacle Detectors w. 2 infrared LED's
- Microphone
- Two Shaft-Encoders



Tekening van: http://ag-vp-www.informatik.uni-kl.de



https://www.analogictips.com/rotary-encoders-part-1-optical-encoders/

Autonomous Robots for Artificial Life

Software (PC, Macintosh, UNIX)

Interactive C Compiler and Libraries

- **motor(0,speed), motor(1,speed)**
- **music: tone(), analog(micro)**
- **get_left_clicks(), get_right_clicks()**
- **analog(photo_left), analog(photo_right)**
- **left_ir, right_ir**
- **left_, right_, back_bumper**

- Note: Microsoft Robotics Studio 4 (2012†): development environment for different robotic platforms (Lego Mindstorm, Fischertechnik, Lynxmotion, Parallax Boe-Bot, Pioneer P3 DX, iRobot Roomba), Kinect (2014†);
- ROS (Robot Operating System) 50+ robots, etc., written in C++. Python, or LISP www.ros.org

Universiteit Leiden. Bij ons leer je de wereld kennen

Autonomous Robots for Artificial Life

Straight ahead

Universiteit Leiden. Bij ons leer je de wereld kennen

Straight Ahead

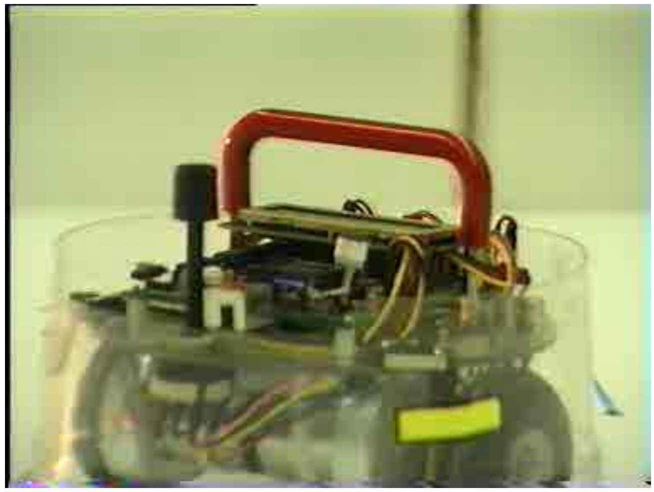
```
void main()
{         int pid_clicks, pid_fahre;
          test_number =! test_number;
          if (test_number)
          {         sleep(1.0); alert_tune();
                    pid_clicks=start_process(clicks());
                    pid_fahre=start_process(fahre_geradeaus());
                    geschwindigkeit = anfangsgeschwindigkeit;
                    while (rclicks < 500) {}
                    ... code to stop ...
                    kill_process(pid_fahre);
                    kill_process(pid_clicks);
                    printf("max. Abw.: %d",dmax);
 } else  printf("----HALT----\n");   }
```


Straight Ahead

```
void clicks()         // Continuously read out odometer clicks
{  init_velocity();
   while(TRUE)
   {
       if (rechts>0.0)
            rclicks+=get_right_clicks();
       else
            rclicks-=get_right_clicks();
       if (links>0.0)
            lclicks+=get_left_clicks();
       else
            lclicks-=get_left_clicks();
            printf("l: %d  r: %d\n",lclicks,rclicks);
   }
}
```

```
void fahre_geradeaus()  // Adapt left and right speed
{  int d;               // to get same number of
   while(TRUE)          // clicks for each wheel
  {  d=rclicks-lclicks;      // Difference
     if (abs(d)>abs(dmax))
            dmax=d;
     links = geschwindigkeit + DELTA*(float) d;
     rechts =geschwindigkeit - DELTA*(float) d;
     drive( 0, links);
     drive( 1, rechts);
     sleep(0.1);
  }
}
```

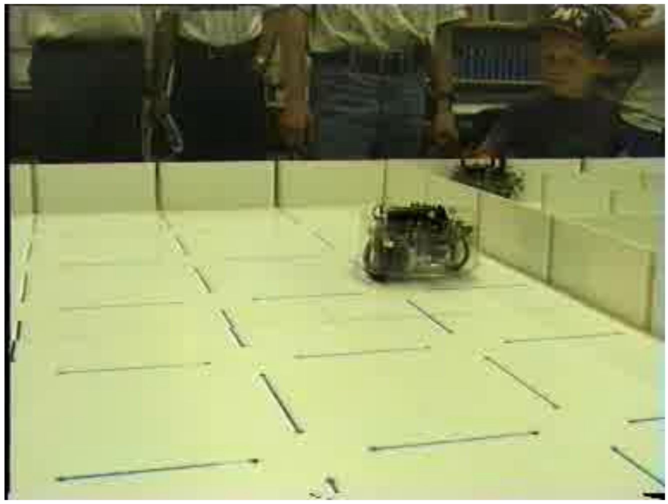
Finding the Light



Finding the Light

```
if ( analog(photo_right) < analog(photo_left) )
{
        motor(0, speed);     /* rechtsdrehen */
        motor(1, -speed);
} else
{
        motor(0, -speed);     /* linksdrehen */
        motor(1, speed);
}
clicks = 0;
while( (  (clicks += (get_left_clicks() + get_right_clicks()) / 2)) < 37 && !all_bumper )
        /* eine Umdregung machen solange kein Bumper betaetigt */
{
        printf("FIND MAX  %d   %d\n", clicks, light);
        light = get_light();              /* Lichtwert holen */
        if ( light > max_light )           /* maximum merken */
        {           max_light = light;           }
        sleep(0.2);
 }
```

Finding the Light 2



Mechanical Tortoise (1951)

British Pathé, 1951.
YouTube: https://www.youtube.com/watch?v=wQE82derooc&t=14s

Finding the Light 2

- Drive along the wall until the light source is found.
- Drive with a left curve until the IR-sensors detect an obstacle, then make a correction to the right until no sensor input is read.
- If an obstacle is found that cannot be resolved this way, then drive 1.5 seconds backwards and start over again.

Vacuum Cleaner & Lawn Mower

TurtleBot4 with ROS

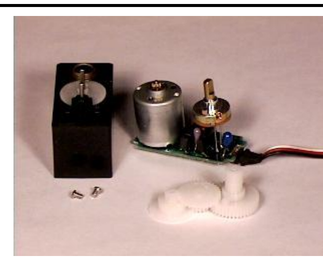
www.ros.org

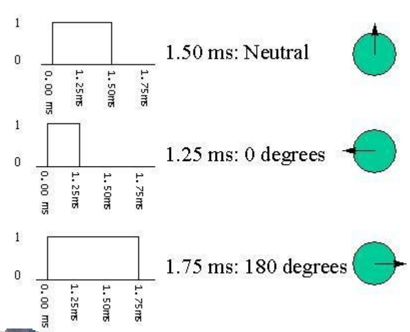
TurtleBot 4		TurtleBot 4 Lite
341 x 339 x 351 mm	Dimensions	341 x 339 x 192 mm
3.9 kg	Weight	3.3 kg
15 kg & 0.31 m/s	Max. Payload & Speed	15 kg & 0.31 m/s
2.5 - 4.0 hrs (load dependent)	Operating Time	2.5 - 4.0 hrs (load dependent)
OAK-D-PRO	Camera	OAK-D-LITE
RPLIDAR-A1	LiDAR	RPLIDAR-A1
Yes	Accessible Power & USB Ports	No
Yes	OLED Display	No
Yes	Mounting Plate	No
ROS 2	Software	ROS 2
Raspberry Pi 4B (4 GB)	Computer	Raspberry Pi 4B (4 GB)

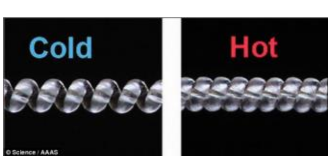
Universiteit Leiden. Bij ons leer je de wereld kennen

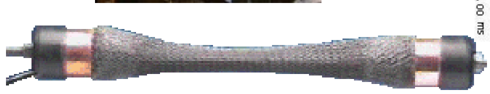
Robotics Actuators

- **Electro motors**
- **Servo's**
- **Stepper Motors**
- **Brushless motors**
- Solenoids
- Hydraulic, pneumatic actuator's
- Magnetic actuators
- Artificial Muscles
- Etc.









Universiteit Leiden. Bij ons leer je de wereld kennen

Universiteit Leiden. Bij ons leer je de wereld kennen


A robot balanced on a ball
Tohoku gakuin univ.
Robot development engineering lab.

Caltech's Leonardo

Caltech's Leonardo, Oct. 2021
https://www.youtube.com/watch?v=fh1AsW22Iks

Universiteit Leiden. Bij ons leer je de wereld kennen

Humanoid Research Platforms

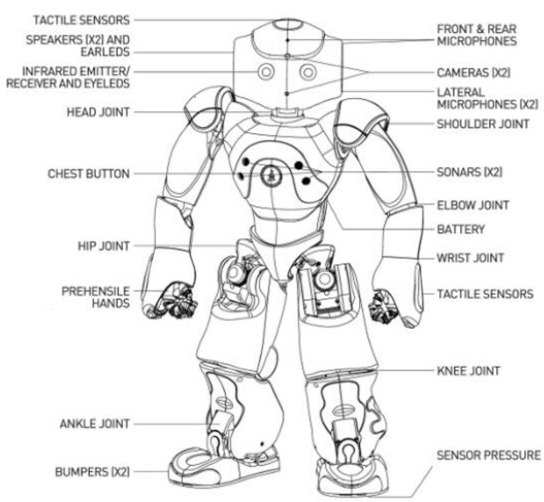
Universiteit Leiden. Bij ons leer je de wereld kennen

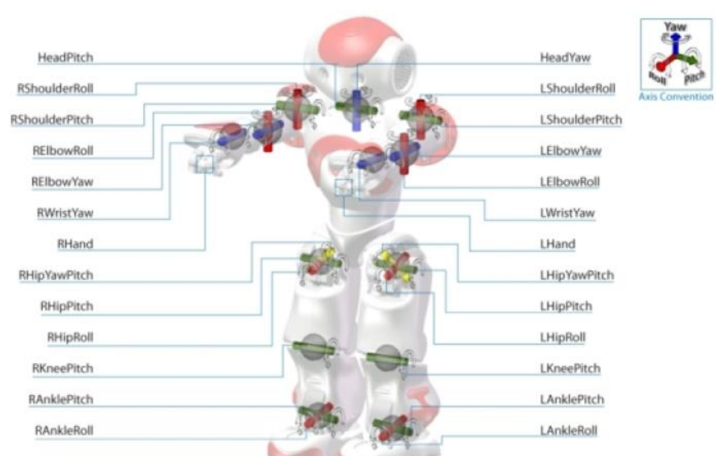
LIACS Humanoid Research Platforms

LML

LIACS
MEDIA
LAB

NAO





http://doc.aldebaran.com/2-1/family/nao_dcm/actuator_sensor_names.html

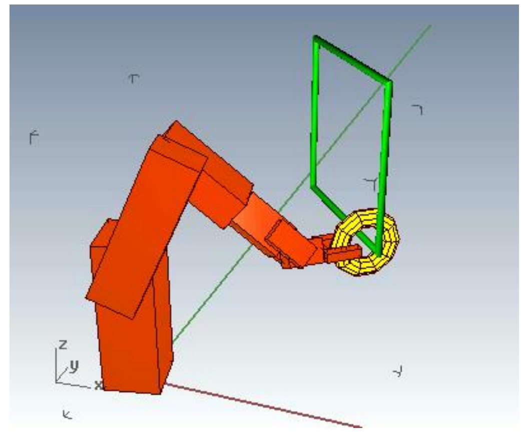
How to move to a goal?

Problem: How to move to a goal?

- Grasp, Walk, Stand, Dance, Follow, etc.

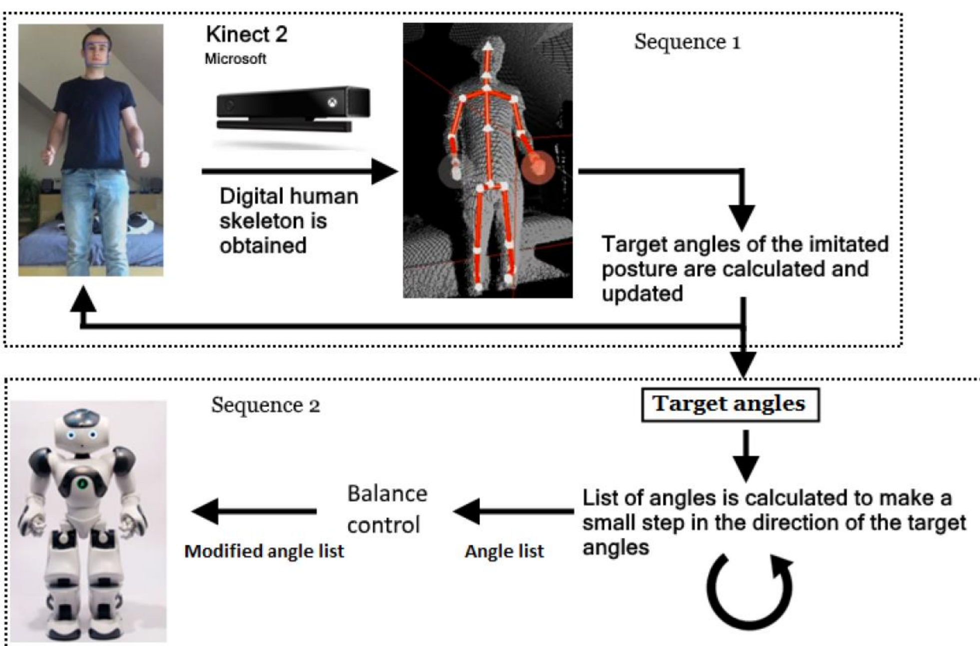
Solution:

- Program step by step.
- Inverse kinematics: take end-points and move them to designated points.
- Trace movements by specialist, human, etc.
- **Learn the right movements:**
  Reinforcement Learning, give a reward when the movement resembles the designated movement.



https://pybullet.org/wordpress/

Universiteit Leiden. Bij ons leer je de wereld kennen



From: Robin Borst, Robust self-balancing robot mimicking, Bachelor Thesis, August 2017

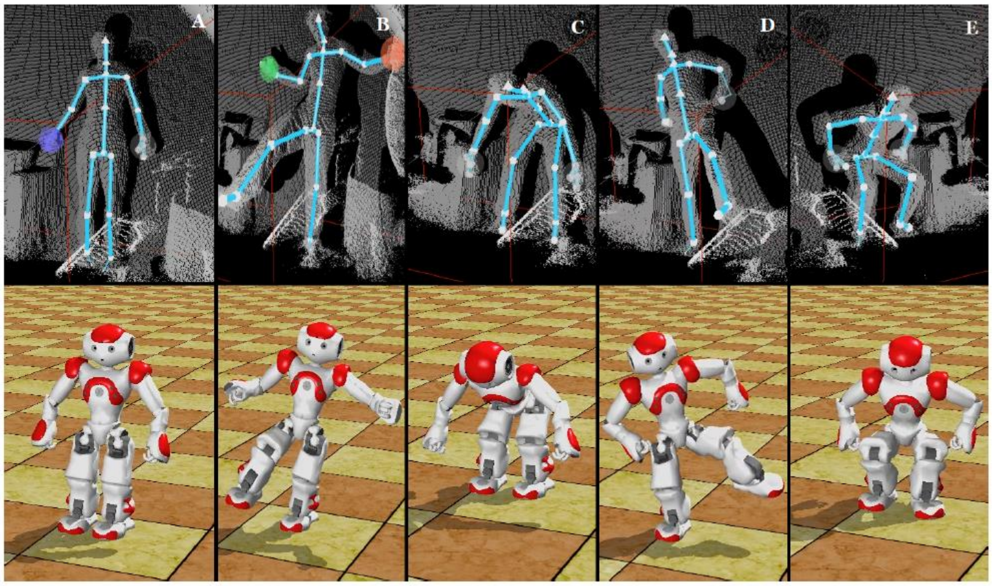


Figure 5.4: The five poses that have been selected to evaluate the effect of the balance controller.

OPNNAR

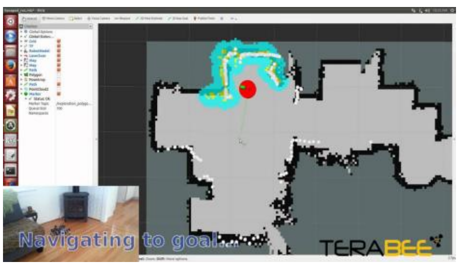
(a) *Start state* (b) *Raise Arm* (c) *Swipe*

K. Maas, Full-Body Action Recognition from Monocular RGB-Video:
A multi-stage approach using OpenPose and RNNs, BSc Thesis, 2021.

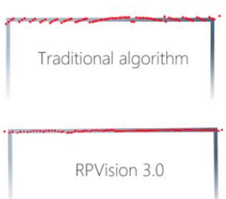
ROBOTICS SENSORS

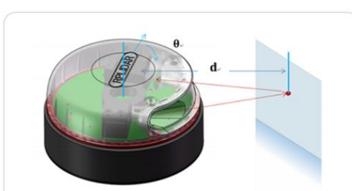
- Bumper switches
- Acceleration, Orientation, Magnetic
- IR/Visible Light
- Pressure, Force
- Ultrasonic, Lidar, Radar
- Camera's, stereo camera's
- Structured Light Camera's

	Ultrasound	Infrared Triangulation	Laser	TeraRanger Time-of-Flight
High reading frequency	✗	✗	✓	✓
Long range	✗	✗	✓	✓
Minimal weight	✓	✓	✗	✓
Small form factor	✓	✓	✗	✓
Eye safety	✓	✓	Class 1 lasers only	✓
Use with multiple sensors	✗	✗	✗	✓

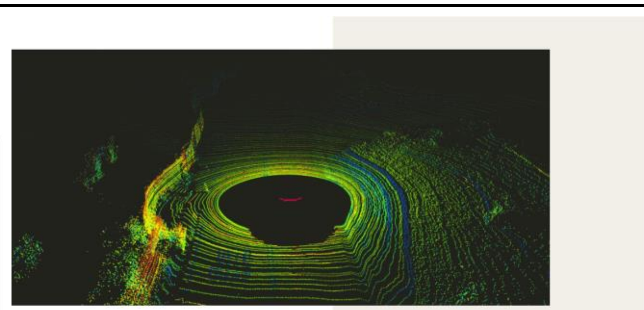


LIDAR Explanation

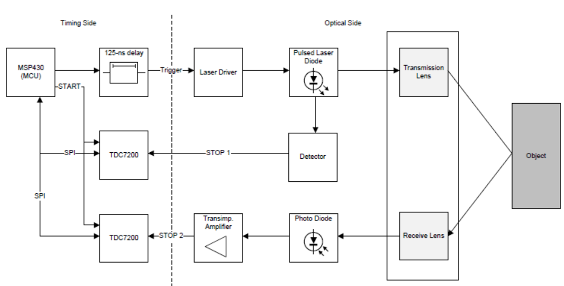




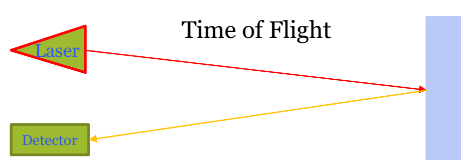
http://www.slamtec.com/en/lidar/A3



https://news.voyage.auto/an-introduction-to-lidar-the-key-self-driving-car-sensor-a7e405590cff



Texas Instruments LIDAR Pulsed Reference Design



- Speed of light ~ 3×10^8 m/s
- In 1 picosecond (= 10^{-12} sec) light travels ~ 3×10^{-4} m = 0.3 mm
- During 33 picoseconds light travels ~ 1cm

Location & Navigation

Problem:

How to locate yourself? How to navigate?

- In unknown or known environment.

With sensors:

- internal, passive, active, gps, beacons, etc.

With or without reference points.

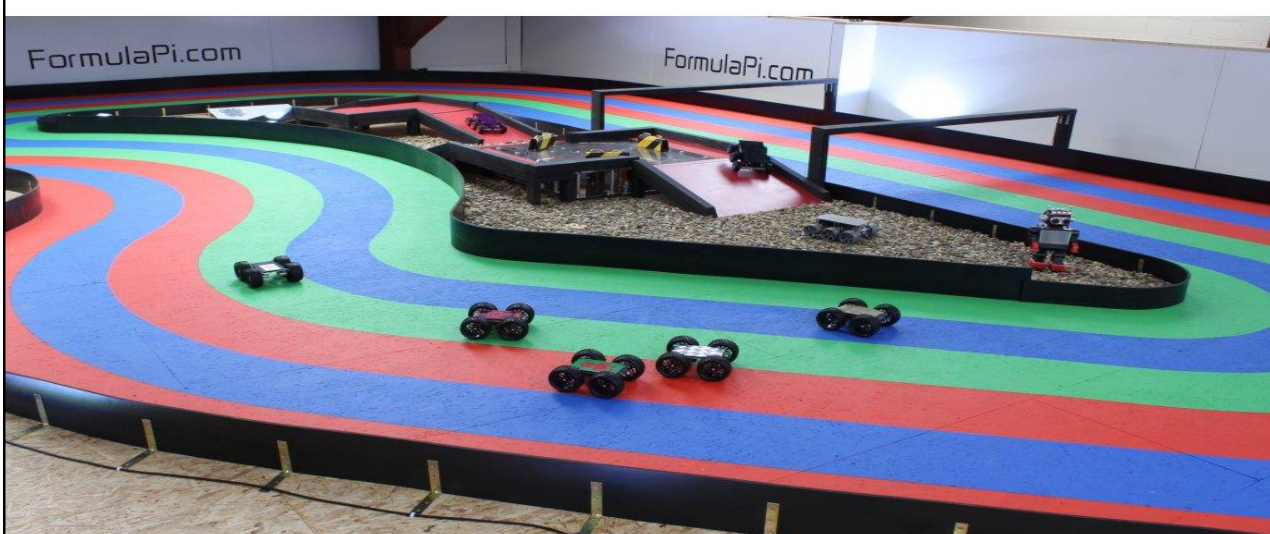
Solution:

- Collect data to determine starting position, or determine your location.
- Move around while collecting data from your environment.
- Sensor data is noisy => location and map building is a stochastic process.
- SLAM

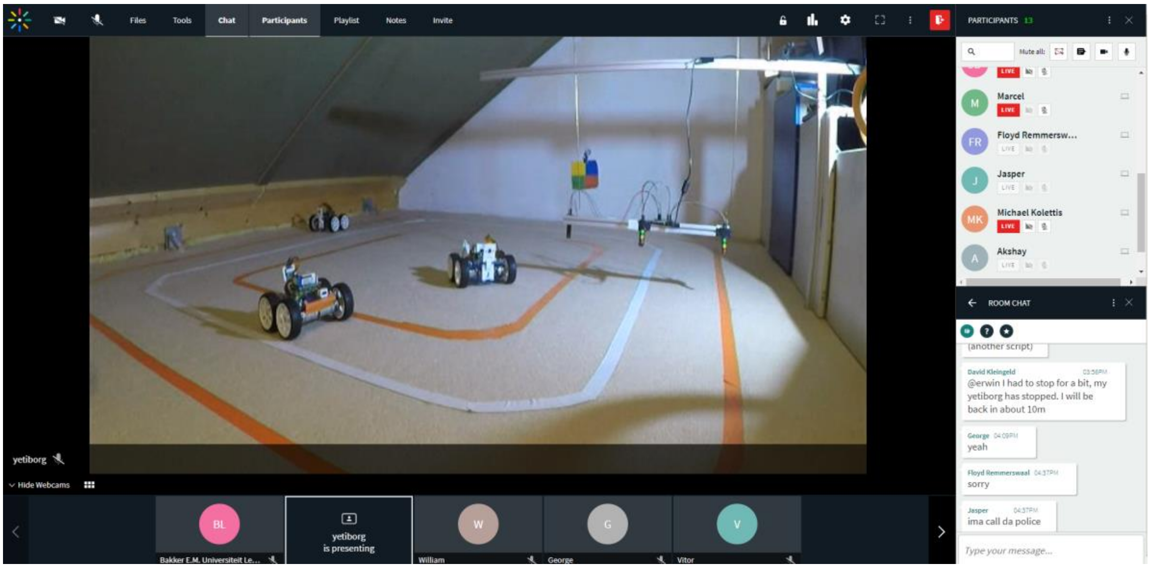
OpenCV.org

Universiteit Leiden. Bij ons leer je de wereld kennen

PiBorg: Yetiborg v2

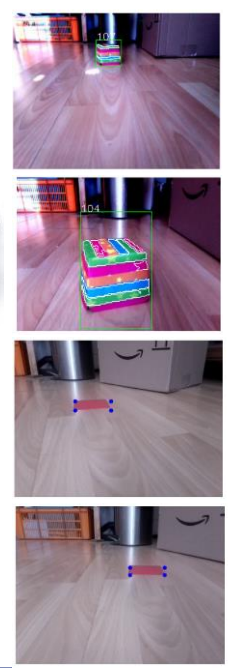


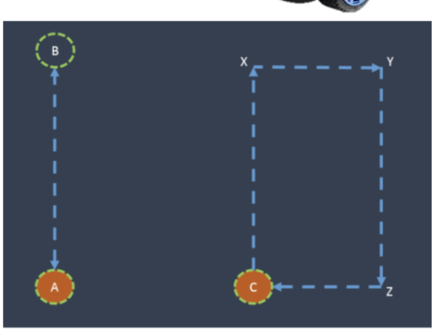
YetiBorg Challenge 2021



Universiteit Leiden. Bij ons leer je de wereld kennen

YetiBorg Challenge 2022





Universiteit Leiden. Bij ons leer je de wereld kennen

Organization and Overview

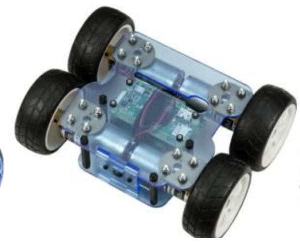
Lecturer:
Dr Erwin M. Bakker (erwin@liacs.nl)
Room 126a and LIACS Media Lab (LML)

Teaching assistant:
Mor Puigventos (email)
TBA (email)

Period: February 6th - May 22nd 2023
Time: Monday 15.15 - 17.00
Place (Rooms):
a) Gorlaeus - Lecture Hall C3
b) Sylvius - 15.31
c) Van Steenis - E0.04
d) Oortgebouw - Sitterzaal

Schedule (tentative, visit regularly):

Date	Room	Subject
6-2	a	Introduction and Overview
13-2	a	Locomotion and Inverse Kinematics
20-2	b	Robotics Sensors and Image Processing
27-2	a	SLAM + SLAM Workshop
6-3	c	Mobile Robot Challenge Introduction
13-3	a	Project Proposals I (by students)
20-3	d	Project Proposals II (by students)
27-3	d	Robotics Vision (Week 13, start 15.30)
3-4	d	Robotics Reinforcement Learning&Workshop
10-4		No Class (Eastern)
17-4	d	Project Progress I (by students)
24-4	d	Project Progress II (by students)
1-5	d	Mobile Robot Challenge I
8-5	a	Mobile Robot Challenge II
15-5	d	Project Demos I
22-5	d	Project Demos II
29-5		Whit Monday
5-6		Project Deliverables

Website: http://liacs.leidenuniv.nl/~bakkerem2/robotics/

Grading (6 ECTS):
- Presentations and Robotics Project (60% of grade).
- Class discussions, attendance, assignments (pass/no pass) 2 workshops (0-10) (2 x 20% = 40% of grade).
- It is necessary to be at every class and to complete every workshop and assignment.

References

1. L. Pinto, J. Davidson, R. Sukthankar, A. Gupta, Robust Adversarial Reinforcement Learning, arXiv:1703.02702, March 2017.
2. S. Gu, E. Holly, T. Lillicrap, S. Levine, Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates, arXiv:1610.00633v2 [cs.RO], October 2016.
3. C. Finn, S. Levine, Deep Visual Forsight for Planning Robot Motion, arXiv:1610.00696, ICRA 2017, October 2016.
4. L. Pinto, J. Davidson, A. Gupta, Supervision via Competition: Robot Adversaries for Learning Tasks, arXiv:1610.01685, ICRA 2017, October 2016.
5. K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised Pixel–Level Domain Adaptation with Generative Adversarial Networks, arXiv:1612.05424, CVPR 2017, December 2016.
6. A. Banino et al., Vector-based navigation using grid-like
7. representations in artificial agents, https://doi.org/10.1038/s41586-018-0102-6, Research Letter, Nature, 2018.
8. R. Borst, Robust self-balancing robot mimicking, Bachelor Thesis, August 2017
9. Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke, Sim-to-Real: Learning Agile Locomotion For Quadruped Robots, https://arxiv.org/pdf/1804.10332.pdf , RSS 2018.

Robotics

Universiteit Leiden

Robotics Homework I

Assignment:

Give a link to the coolest, strangest, most impressive, most novel, or technologically inspirational robot you could find. And describe in a short paragraph (< 100 words) why you selected this robot.

NB Boston Dynamics Robots are excluded this time (I know they are very cool).

Grading: Pass/No Pass

Due: Monday 13-2 2023

See BrightSpace Assignment(s) to upload your answer in a pdf file with your name and study.