# Robotics

Erwin M. Bakker| LIACS Media Lab 15-2 2019

Universiteit Leiden

Bij ons leer je de wereld kennen

# Organization and Overview

**Period:** February 15th - May 10th 2019
**Time:** Friday 09.00 – 10.45
**Place:** LIACS, Room 401 (Workshops Room 303)
**Lecturer:** Dr Erwin M. Bakker ( erwin@liacs.nl )
**Assistant:** Andrius Bernatavicius

NB E-mail your name and student number to erwin@liacs.nl

**Schedule:**

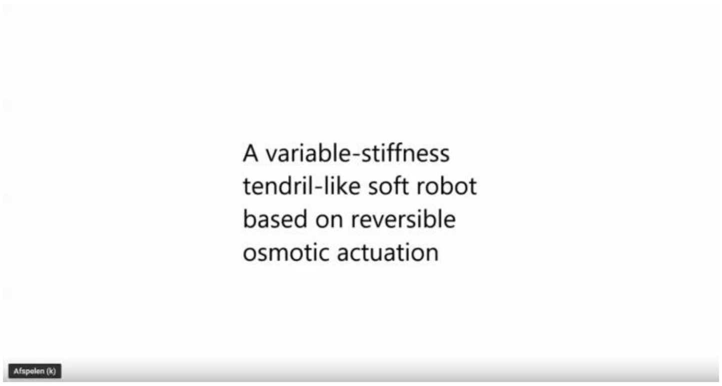
15-2 Introduction and Overview
22-2 Locomotion and Inverse Kinematics
1-3 Sensors and Algorithms
8-3 *SLAM Workshop I and Yetiborg Introduction*
15-3 *Project Proposals (presentation by students)*
22-3 *ROS Workshop II* and Yetiborg Qualification
29-3 Robotics Image Processing
5-4 *Yetiborg Race and/or Nao Workshop III*
12-4 Robotics Image Processing and Understanding
19-4 No Class
26-4 Robotics Reinforcement Learning.
3-5 Robotics Reinforcement Learning Workshop IV
10-5 ***Project Demos** (by students)*

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

Grading (6 ECTS): Presentations and Robotics Project (60% of grade). Class discussions, attendance, workshops and assignments (40% of grade). It is necessary to be at every class and to complete every workshop.

Universiteit Leiden. Bij ons leer je de wereld kennen

# Robotics in the News





Indrek Must, Edoardo Sinibaldi, Barbara Mazzolai. **A variable-stiffness tendril-like soft robot based on reversible osmotic actuation**. *Nature Communications*, 2019; 10 (1) DOI: 10.1038/s41467-018-08173-y



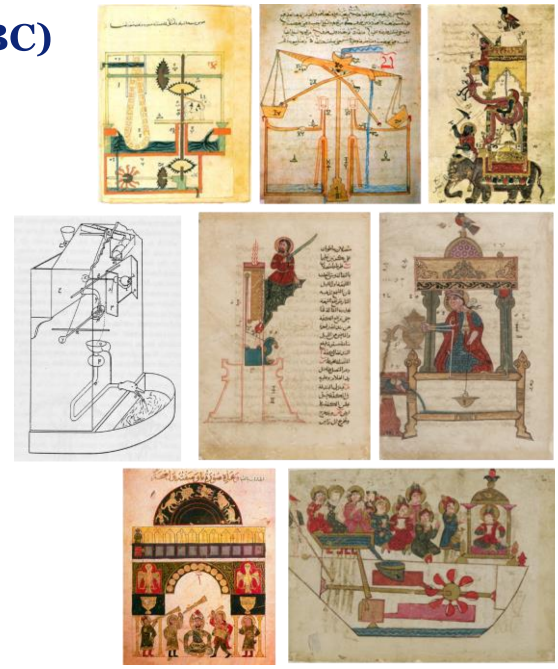
# Robotics in the News


University of California at Berkeley, Feb. 2011


https://foldimate.com/ , CES 2019

## 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



Universiteit Leiden. Bij ons leer je de wereld kennen

## Leonardo da Vinci (1452 – 1519)

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

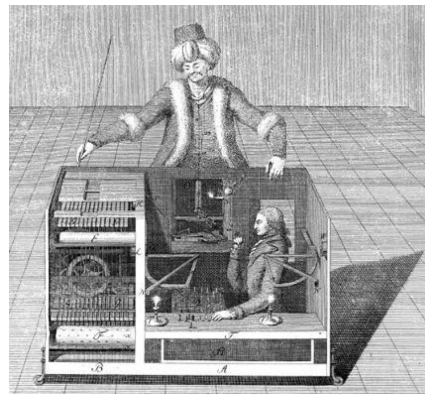




Pictures from:
http://www.leonardo3.net
http://brunelleschi.imss.fi.it

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



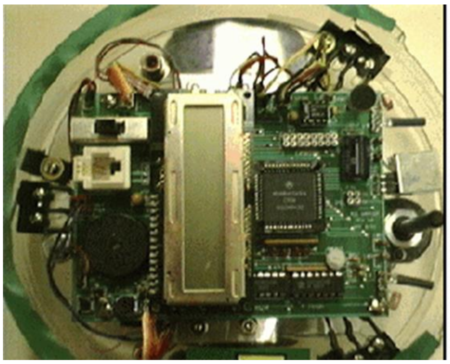
# LOCOMOTION & INVERSE KINEMATICS

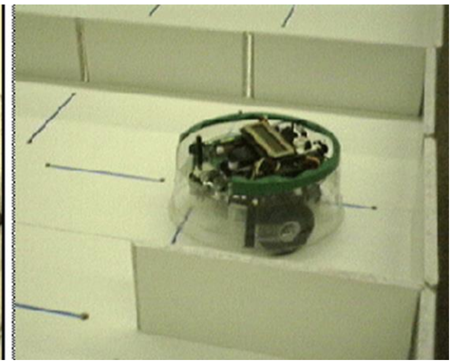
# EARLY ROBOTS

South Pointing Chariot
by Ma Jun ( c. 200–265 )

Universiteit Leiden. Bij ons leer je de wereld kennen

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

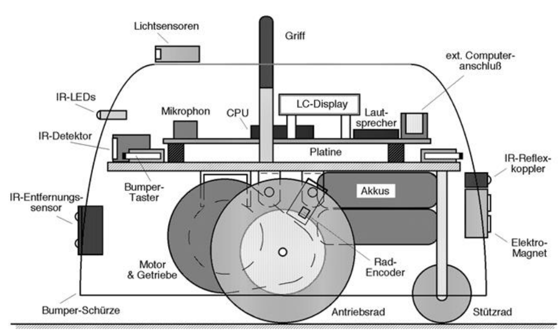






# Autonomous Robots for Artificial Life

- 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

# 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: 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.

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");   }
```

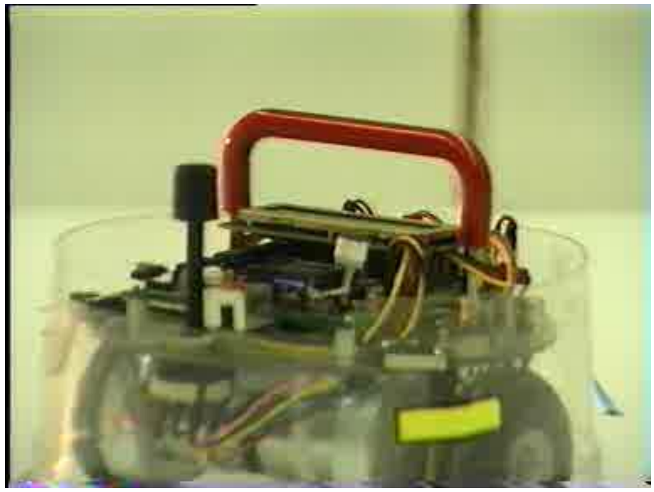


# Straight Ahead

```
void clicks()        // Continuously read out odometer
{ 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()
{ int d;
  while(TRUE)
  { 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

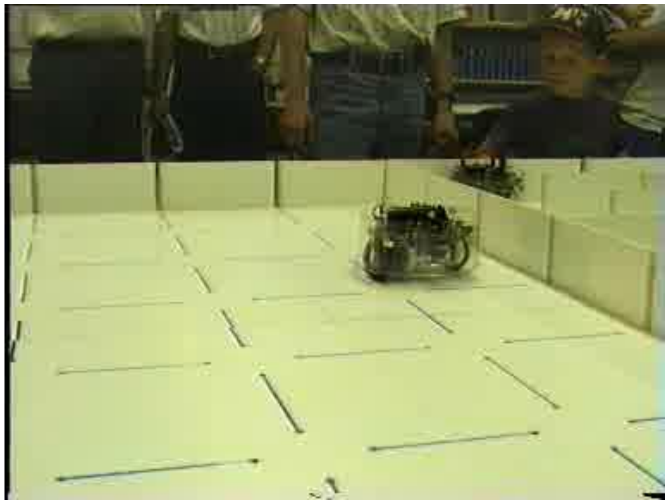




# Finding the Light

```
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





# Finding the Light 2

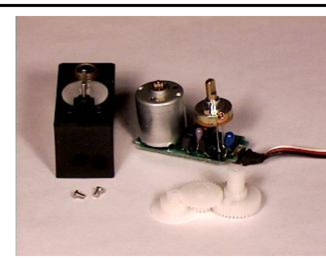
- Drive along the wall until the light source is found.
- Drive with a left curve until the IR-sensors detect an obstacle, then make a correction to the right until no sensor input is read.
- If an obstacle is found that cannot be resolved this way, then drive 1.5 seconds backwards and start over again.

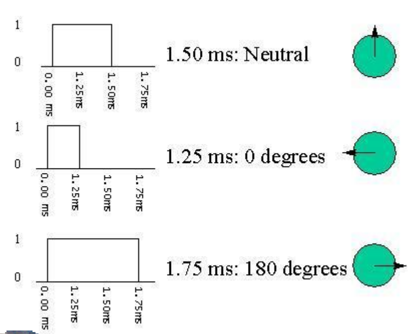
# Vacuum Cleaner & Lawn Mower

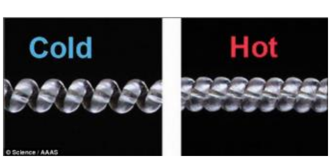


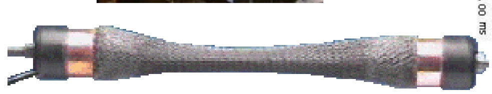
# Robotics Actuators

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





Universiteit Leiden. Bij ons leer je de wereld kennen

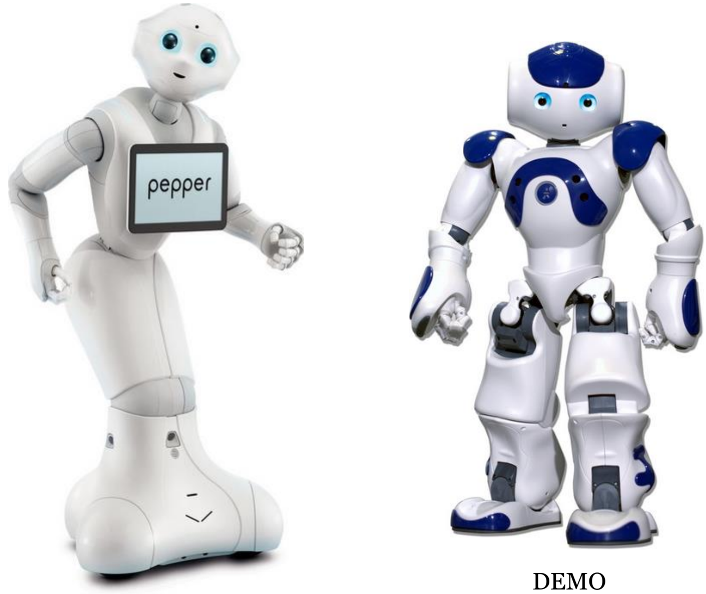
Universiteit Leiden. Bij ons leer je de wereld kennen




# Humanoid Research Platforms


Universiteit Leiden. Bij ons leer je de wereld kennen

# LIACS Humanoid Research Platforms



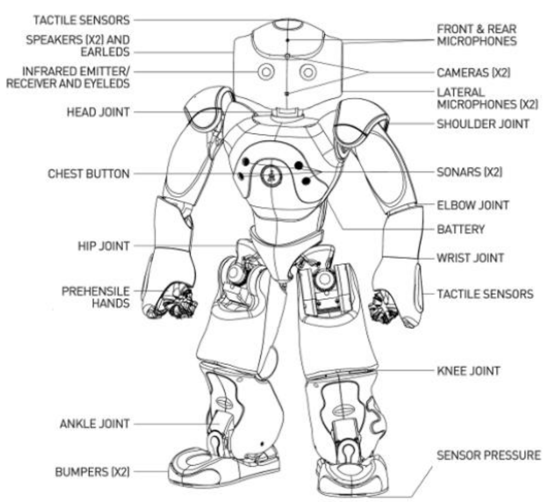

**LML**

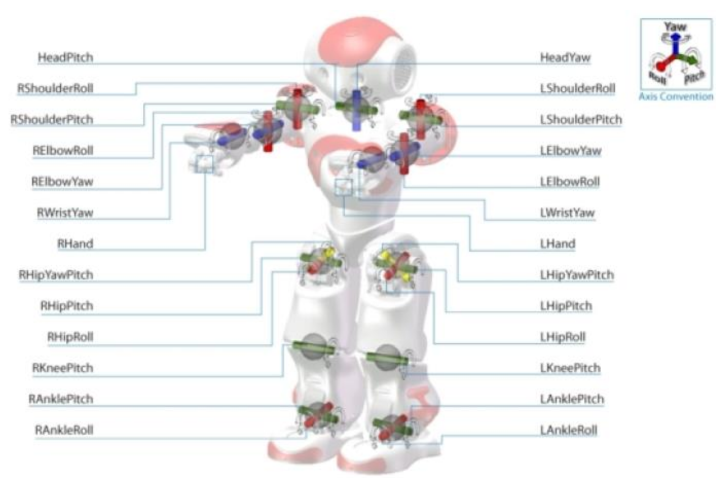
LIACS
MEDIA
LAB

DEMO

Universiteit Leiden. Bij ons leer je de wereld kennen

# NAO




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

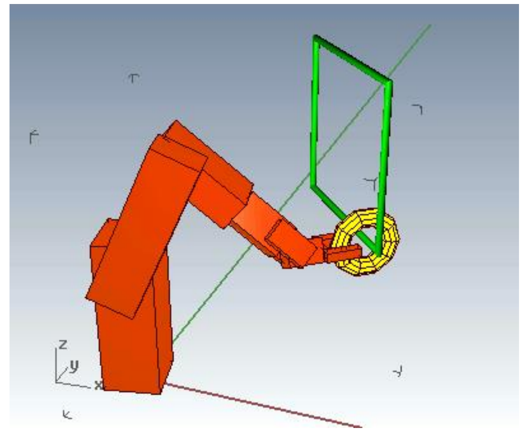
# How to move to a goal?

**Problem: How to move to a goal?**

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

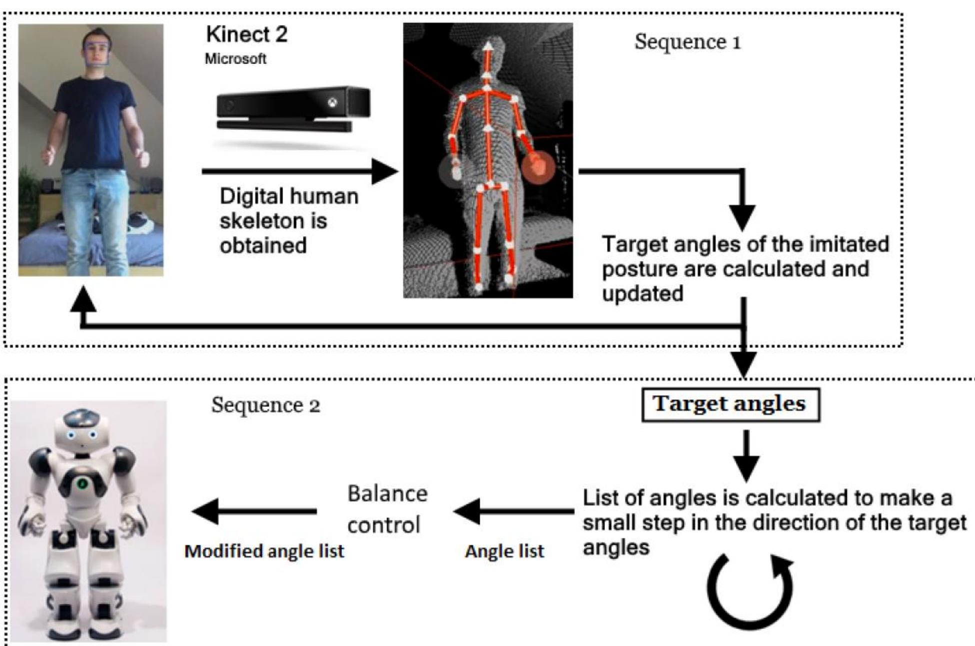
**Solution:**

- 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/






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

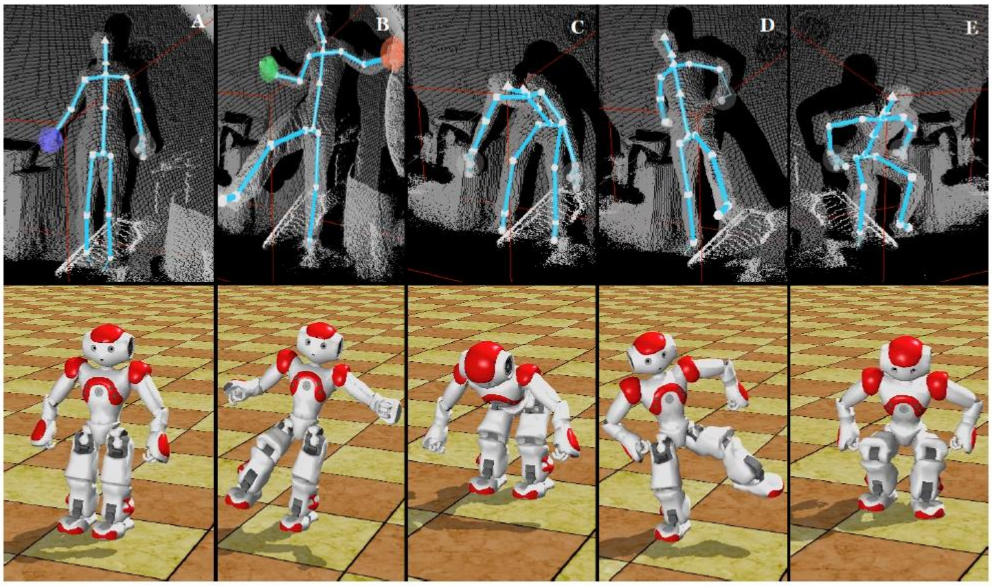
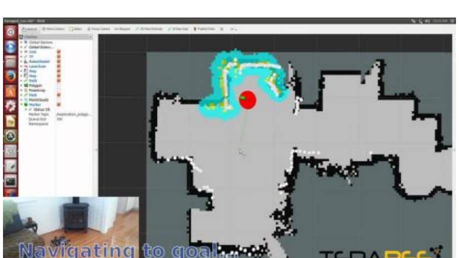


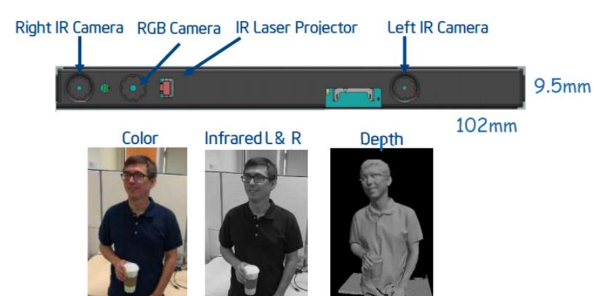

Figure 5.4: The five poses that have been selected to evaluate the effect of the balance controller.

# ROBOTICS SENSORS

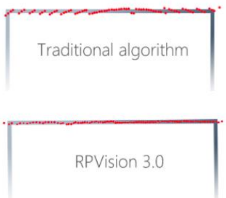
- Bumper switches
- Acceleration, Orientation, Magnetic
- IR/Visible Light
- Pressure, Force
- Ultrasonic, Lidar, Radar
- Camera's, stereo camera's
- Structured Light Camera's

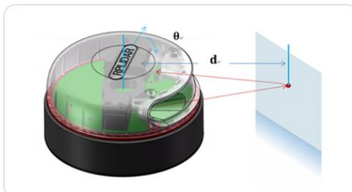

| | Ultrasound | Infrared Triangulation | Laser | TeraRanger Time-of-Flight |
|---|---|---|---|---|
| High reading frequency | ✗ | ✗ | ✓ | ✓ |
| Long range | ✗ | ✗ | ✓ | ✓ |
| Minimal weight | ✓ | ✓ | ✗ | ✓ |
| Small form factor | ✓ | ✓ | ✗ | ✓ |
| Eye safety | ✓ | ✓ | Class 1 lasers only | ✓ |
| Use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



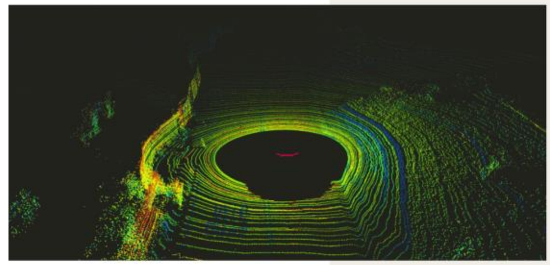



# LIDAR Explanation

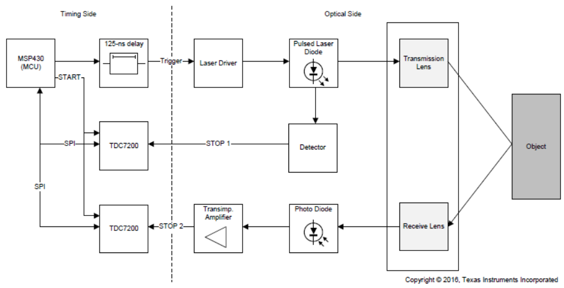


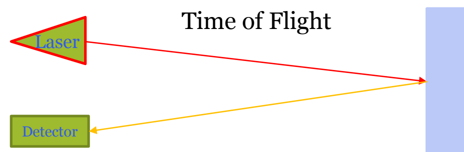

http://www.slamtec.com/en/lidar/A3



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

Universiteit Leiden. Bij ons leer je de wereld kennen

# Location & Navigation

**Problem:**

How to locate yourself? How to navigate?

- In unknown or known environment.

With sensors:

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

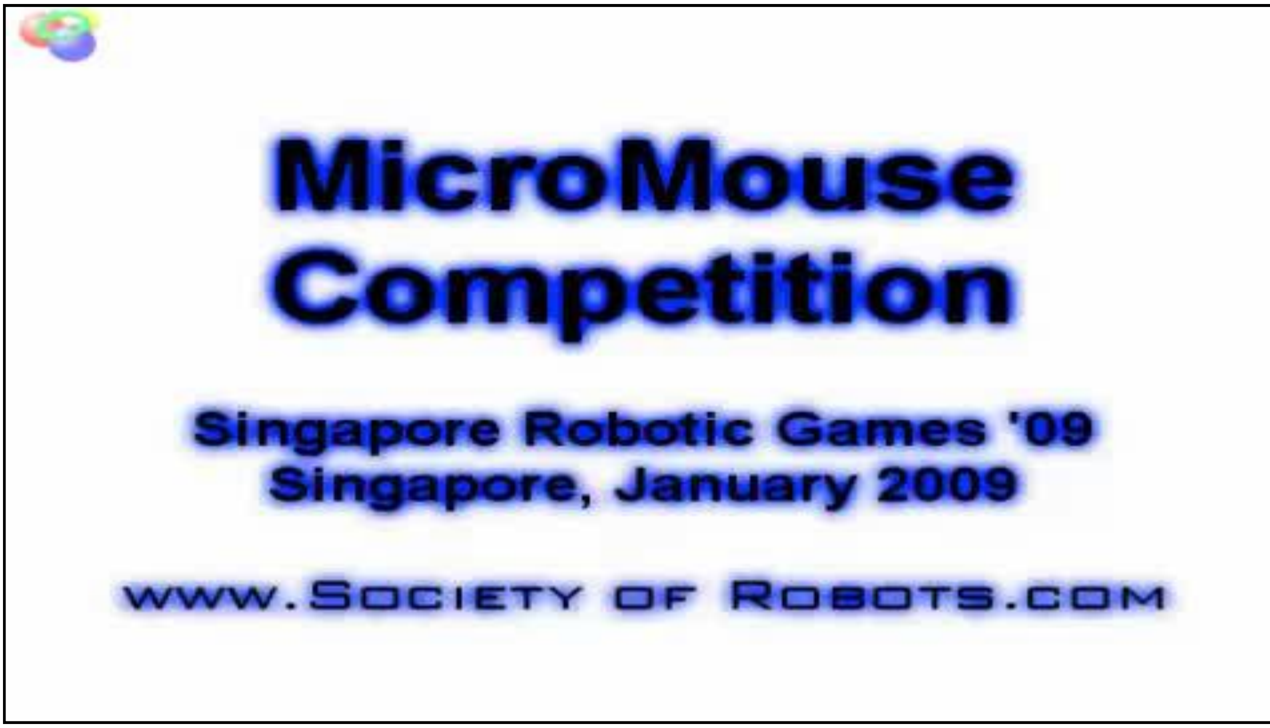
With or without reference points.


**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

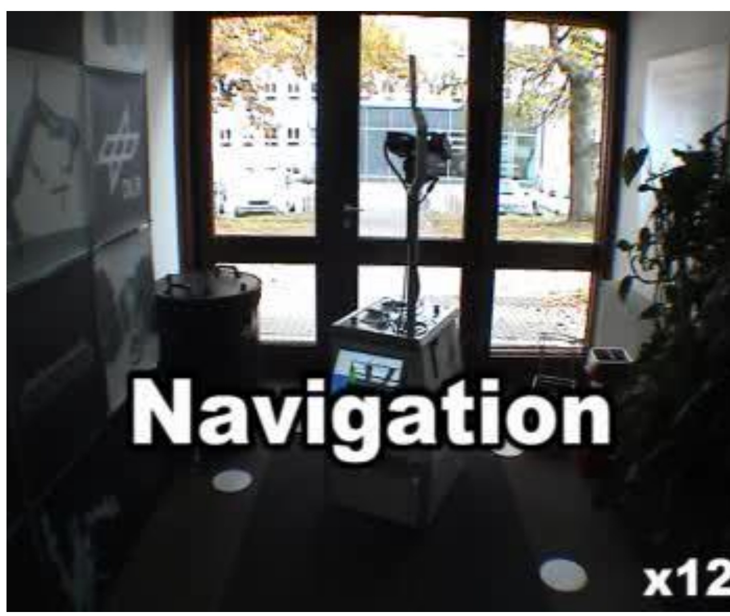
Universiteit Leiden. Bij ons leer je de wereld kennen




## Landmark Navigation

**Only proprio-perception**
- dead reckoning
- no-dead reckoning (no known starting position)
- drift errors

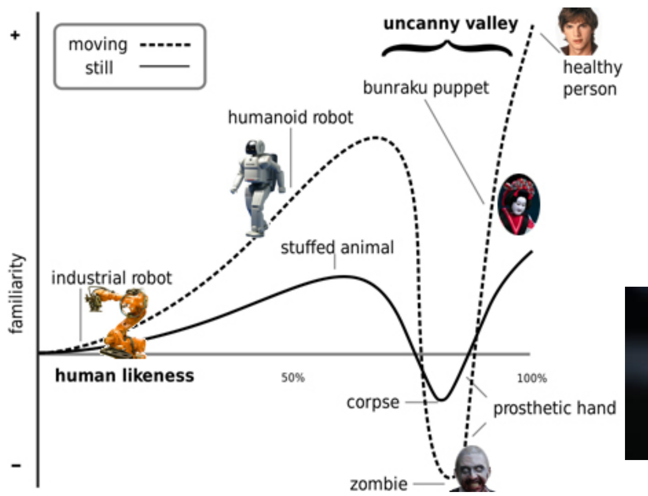
**With extero-ception**
- global odometry
- corrections by sighting of known environmental features, etc.




Universiteit Leiden. Bij ons leer je de wereld kennen

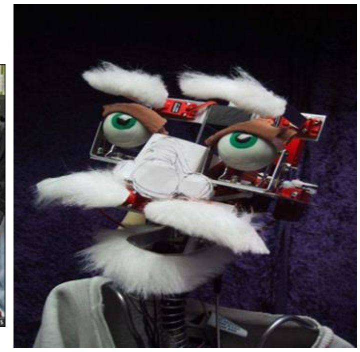
# PiBorg: Yetiborg v2


# Human and Robot Interaction




From: spectrum.ieee.org/automaton/robotics/humanoids/040210-who-is-afraid-of-the-uncanny-valley

Hiroshi Ishiguro



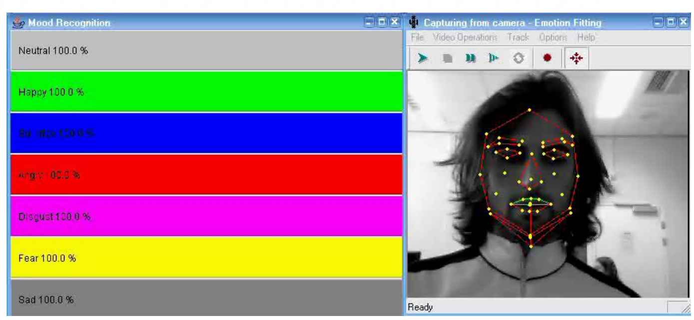
Robot Sophia

Universiteit Leiden. Bij ons leer je de wereld kennen

# IMAGE PROCESSING & UNDERSTANDING

**Problem: How to understand your environment?**

- What are obstacles?
- What are the objects?
- Can you locate the objects you have to manipulate?
- Humans, voice, speech, and emotion recognition?




**Solution:**

- Basic image processing.
- Content based image understanding, classification.
- Speech recognition, speech synthesis.
- Multimodal data processing.
- Natural Language Processing, etc.

Universiteit Leiden. Bij ons leer je de wereld kennen

# Robotics at LIACS Media Lab

**Content Based Retrieval**

**Robotics Research**

- Vision, audio, and passive sensor based
- Environment and object recognition and understanding
- Paradigms: DNN, RL and ARL
- …

**Education**

- Multimedia Systems
- Multimedia Information Retrieval
- **Robotics**
- **Robotic Projects** and many more

# Robotics Challenges

DARPA Challenges

- Autonomous vehicles
- Search and Rescue
- SHRIMP (https://www.darpa.mil/news-events/2018-07-17 ), Etc.


JULY 22, 2018 WEBLOG

DARPA has competition plans for insect-scale robots

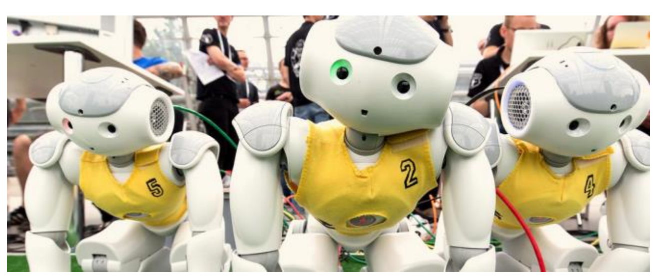
by Nancy Owano , Tech Xplore

Credit: DARPA

Earlier this month, DARPA announced it is launching a new SHort-Range Independent Microrobotic Platforms (SHRIMP) program. SHRIMP will develop and demonstrate micro-to-milli robotic platforms for scenarios brought on by natural and critical disasters.

Robocup

- Goal: A team of robots defeating the human world champion in 2050.
- Yearly event
- Different Leagues:
- Simulation league
- Small-size, Mid-size
- Aibo-, Nao-, Humanoid League



Robocup 2018 Montreal

Universiteit Leiden. Bij ons leer je de wereld kennen

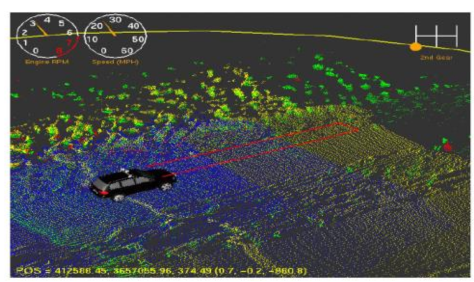



# DARPA Challenge 2005

**$2 million Prize awarded to**

**the Stanford Racing Team**

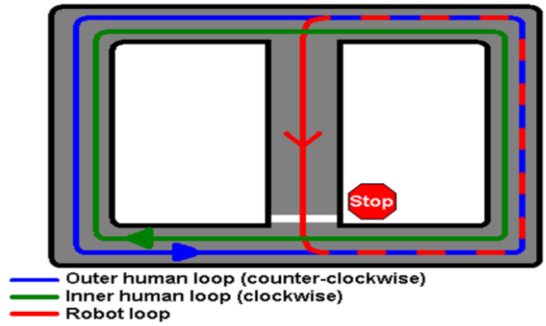
Five teams completed the Grand Challenge; four of them under the 10 hour limit. The Stanford Racing Team took the prize with a winning time of 6 hours, 53 minutes.



Universiteit Leiden. Bij ons leer je de wereld kennen

# URBAN Challenge 2007

A new DARPA challenge, which took place at the George Air Force Base on a 96 km urban area course, to be completed in less than 6 hours, obeying all traffic regulations while negotiating with other traffic




**Results**

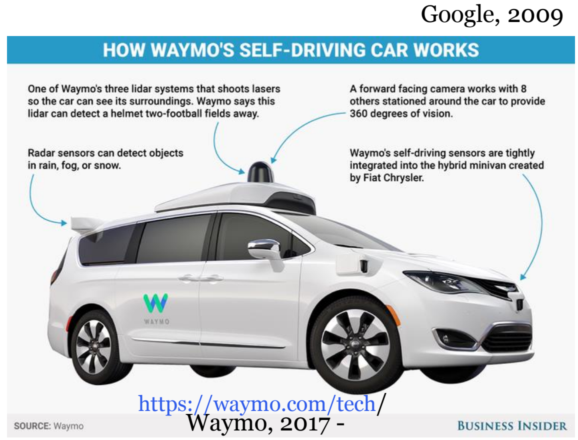
1. Tartan Racing ($2 million), a collaborative effort by Carnegie Mellon University and General Motors Corporation, with vehicle "Boss", a Chevy Tahoe (average 14 miles/hour)
2. Stanford Racing Team ($1 million) with vehicle "Junior", a 2006 Volkswagen Passat (average 13 miles/hour).
3. Victor Tango from Virginia Tech ($500,000) with their 2005 Ford Escape hybrid, "Odin“ (average 13 miles/hour)




# Robotics Self Driving Cars

Google Firefly, 2015


Google, 2009


https://waymo.com/tech/
Waymo, 2017 -

# Complex Tasks

**Problem: How to train on complex tasks?**

- Grasping different objects, clothes.
- Interact with people in natural environments, drive a car, etc.

**Solution:**

- Training in the real world.

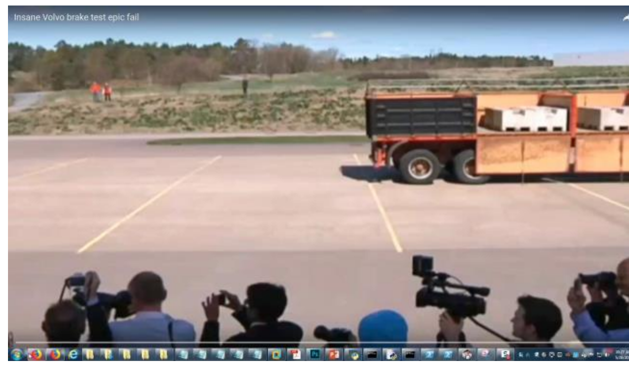
**Draw backs:** Too costly, too little iterations, etc.

**Deep neural networks success in the field of Reinforcement Learning:**

- Fast computations
- Fast Simulations
- Improved networks

**But, most RL-based approaches fail to generalize, because:**

1. gap between simulation and real world
2. policy learning in real world is hampered by data scarcity


Universiteit Leiden. Bij ons leer je de wereld kennen

# REINFORCEMENT LEARNING

## In the Real World: use more robots

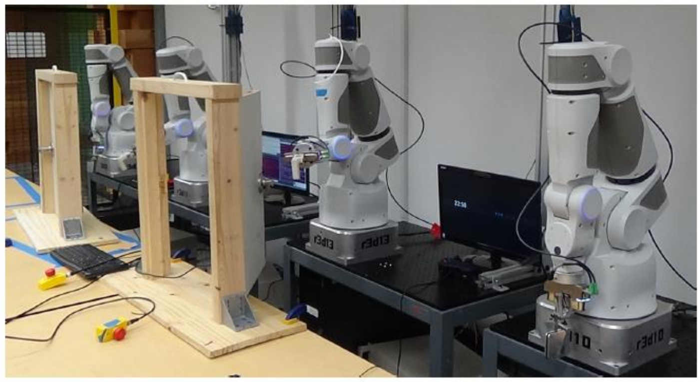


Fig. 1: Two robots learning a door opening task. We present a method that allows multiple robots to cooperatively learn a single policy with deep reinforcement learning.

From [2] Gu et al. , Nov. 2016.

## Robust Adversarial Reinforcement Learning (RARL) to close the Reality Gap

Training of an agent in the presence of a **destabilizing adversary**

- Adversary can employ disturbances to the system
- Adversary is trained at the same time as the agent
- Adversary is reinforced: it learns an optimal destabilization policy.

Here policy learning can be formulated as a zero-sum, minimax objective function.

# Adversary with Domain Knowledge

https://gym.openai.com/

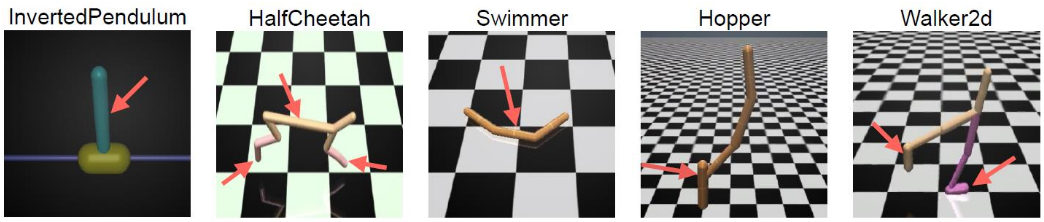



*Figure 1.* We evaluate RARL on a variety of OpenAI gym problems. The adversary learns to apply destabilizing forces on specific points (denoted by red arrows) on the system, encouraging the protagonist to learn a robust control policy. These policies also transfer better to new test environments, with different environmental conditions and where the adversary may or may not be present.

Figure from [1].

# Actions of Adversary

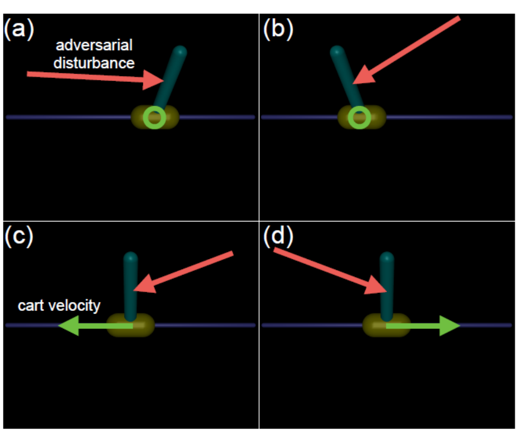



*Figure 8.* Visualization of forces applied by the adversary on InvertedPendulum. In (a) and (b) the cart is stationary, while in (c) and (d) the cart is moving with a vertical pendulum.

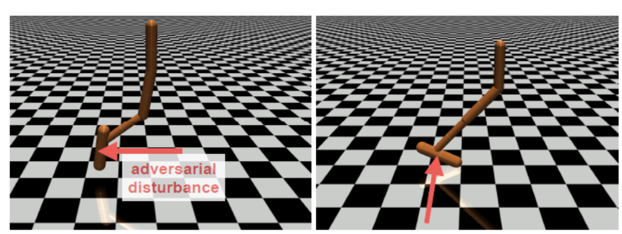



*Figure 9.* Visualization of forces applied by the adversary on Hop per. On the left, the Hopper's foot is in the air while on the righ the foot is interacting with the ground.

# Organization and Overview

**Period:** February 15th - May 10th 2019
**Time:** Friday 09.00 – 10.45
**Place:** LIACS, Room 401 (Workshops Room 303)
**Lecturer:** Dr Erwin M. Bakker ( erwin@liacs.nl )
**Assistant:** Andrius Bernatavicius

NB E-mail your name and student number to erwin@liacs.nl

**Schedule:**

| | |
|---|---|
| 15-2 | Introduction and Overview |
| 22-2 | Locomotion and Inverse Kinematics |
| 1-3 | Sensors and Algorithms |
| *8-3* | *SLAM Workshop I and Yetiborg Introduction* |
| *15-3* | *Project Proposals (presentation by students)* |
| 22-3 | *ROS Workshop II* and Yetiborg Qualification |
| 29-3 | Robotics Image Processing |
| 5-4 | *Yetiborg Race and/or Nao Workshop III* |
| 12-4 | Robotics Image Processing and Understanding |
| 19-4 | No Class |
| 26-4 | Robotics Reinforcement Learning. |
| 3-5 | Robotics Reinforcement Learning Workshop IV |
| 10-5 | ***Project Demos** (by students)* |

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

Grading (6 ECTS): Presentations and Robotics Project (60% of grade). Class discussions, attendance, workshops and assignments (40% of grade). It is necessary to be at every class and to complete every workshop.

Universiteit Leiden. Bij ons leer je de wereld kennen

# Robotics Homework I

## Assignment:

Give a link to the coolest, strangest, most impressive, most novel, or technologically inspirational robot you could find.

**Due:** Thursday 21-2 at 14.00 PM.

Email your link to erwin@liacs.nl with subject ‘Robotics’.

Universiteit Leiden. Bij ons leer je de wereld kennen

# 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