Note: Research Thesis

This is a thesis topic that is designed as an opportunity for excellent students who are interested in getting a first dive into research.

For this topic, there is a very high risk of failure!!!

Please note that this only make sense if

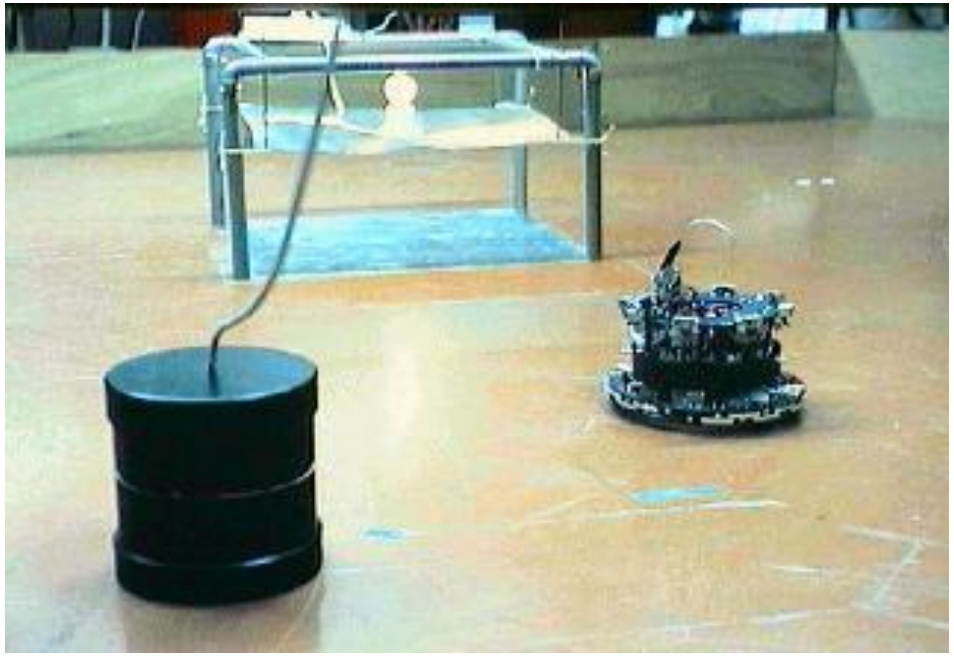
a) you understand the topic presented in the slides,
b) you are willing to work yourself into the topic and to read some background material,
c) you have excellent theoretical skills, and
d) you are willing and capable to work independently on a challenging topic.

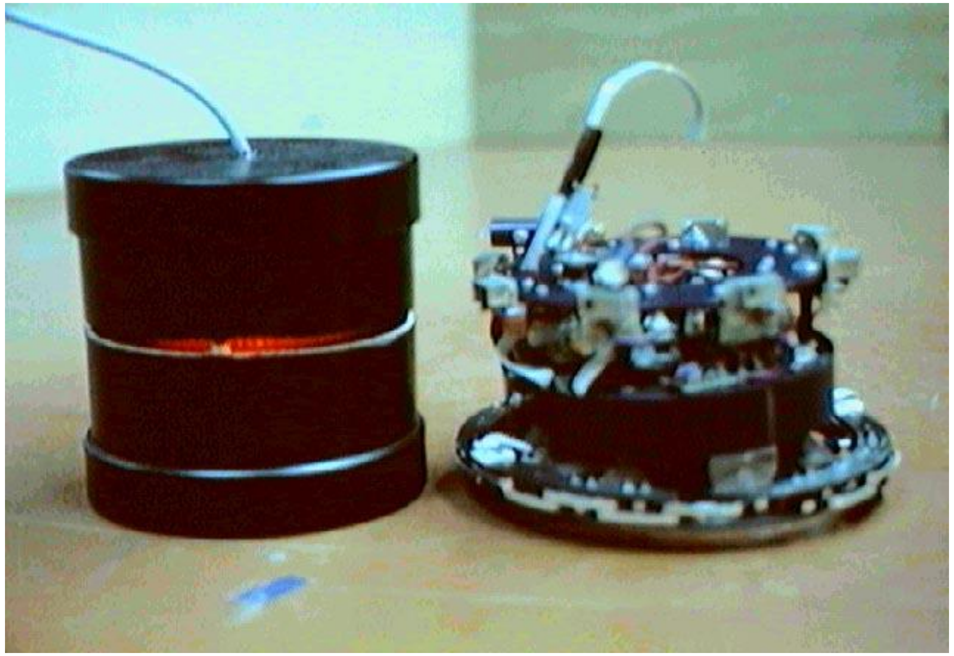
As a **reward**, there is a **high likelihood** that a **scientific publication** is the outcome.

B-Scheduling on Ubuntu Linux

Context: B-Scheduling is an efficient way to control robots with behavior processes
http://robotics.jacobs-university.de/TMP/BScTheses/data/B-Schedule/NPDL-RAS02.pdf
(more related papers will be provided)

Example for Behavior-Based Control: robots “living” in an artificial ecosystem

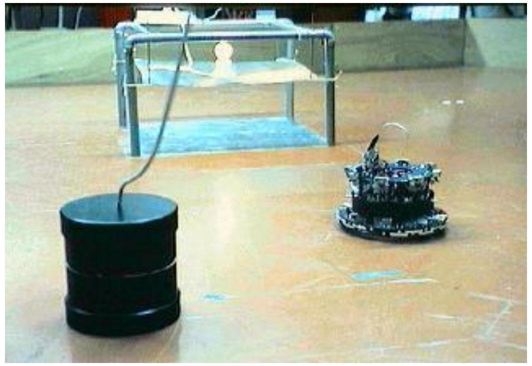


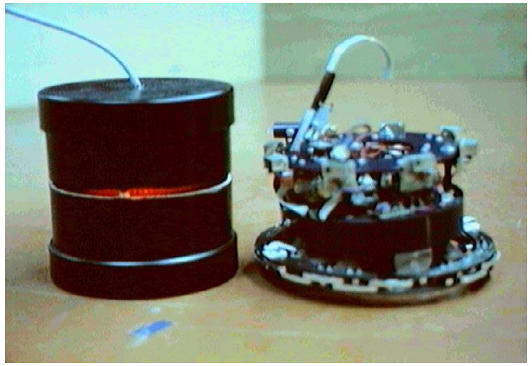


B-Scheduling on Ubuntu Linux

ecosystem

- "simple" mobile robots
  - with chargeable batteries
- charging-station
  - with white light on top to indicate the location
- competitors
  - boxes with red lamps
  - connected to same energy source as charging station
  - they hence "eat away" energy the robots need
  - but robots can (temporarily) knock them out
  - competitors hence establish a working task for the robots

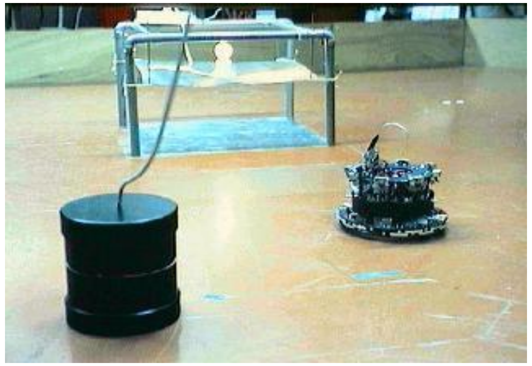


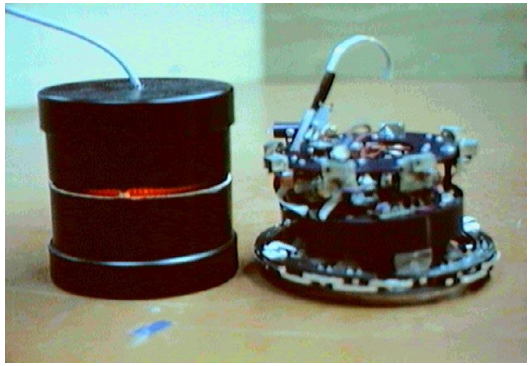


B-Scheduling on Ubuntu Linux

robot behaviors in the ecosystem

- touch-based obstacle avoidance
- active IR obstacle avoidance
  - active IR is a distance sensor
- photo-taxis to the charging-station
  - taxis to a beacon (white light)
- charging
  - stop when current is flowing
  - move when the battery is charged
- attraction to competitors
  - 2nd taxis to a beacon (red light)

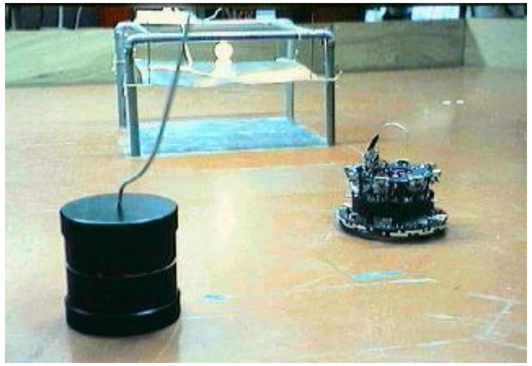


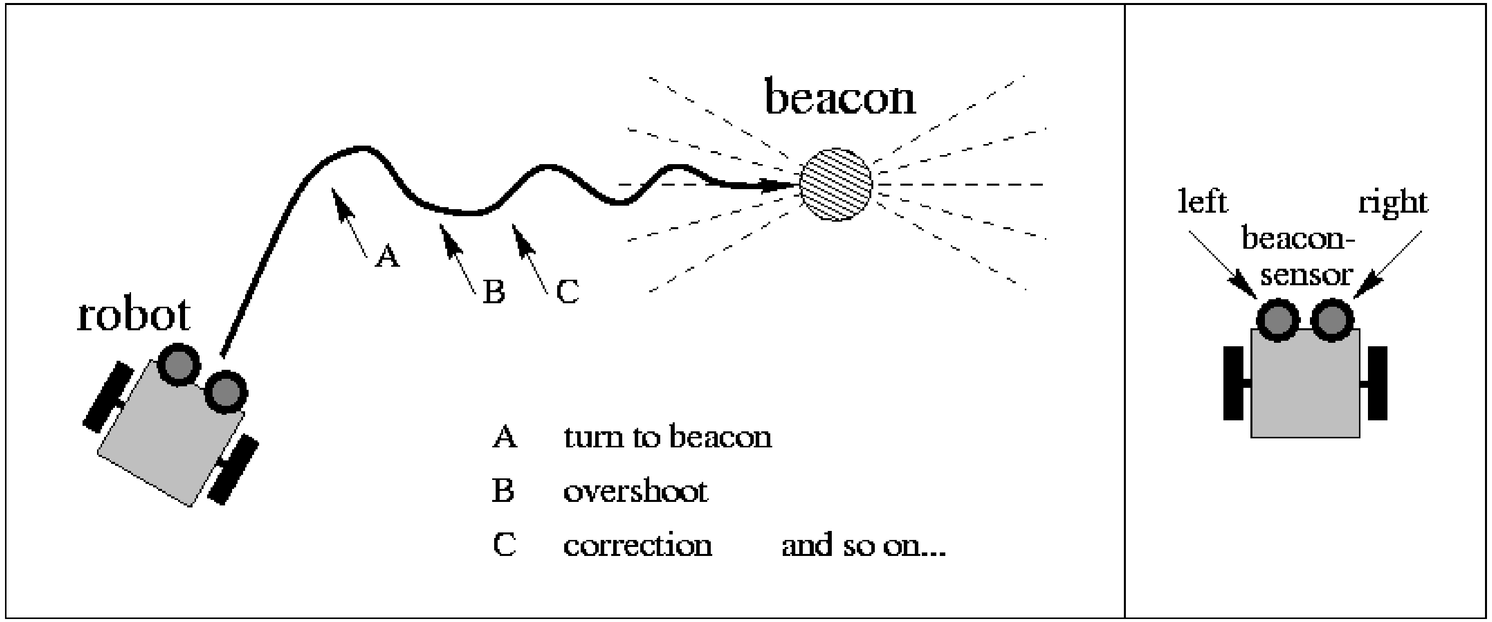


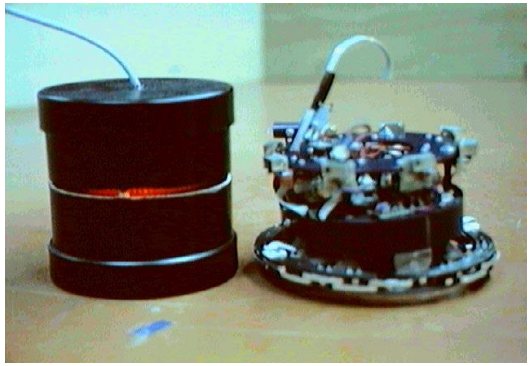
B-Scheduling on Ubuntu Linux

robot behaviors in the ecosystem are simple processes,

- e.g., photo-taxis to a light source (beacon)
- two sensors: rotate towards the beacon plus move forward

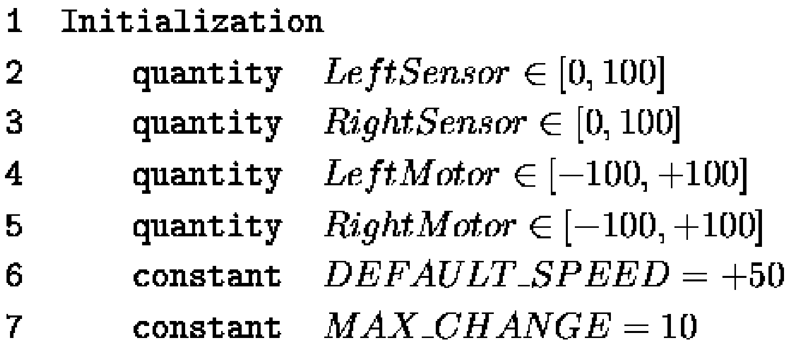




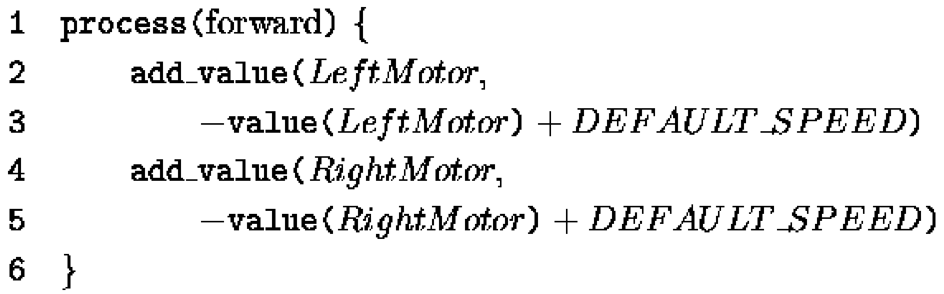


B-Scheduling on Ubuntu Linux

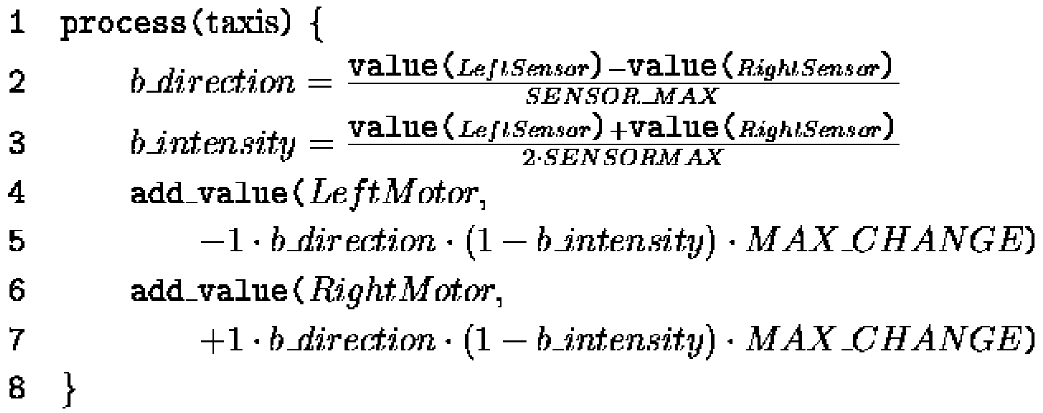
e.g., photo-taxis to a beacon in PDL (see also AI lecture)



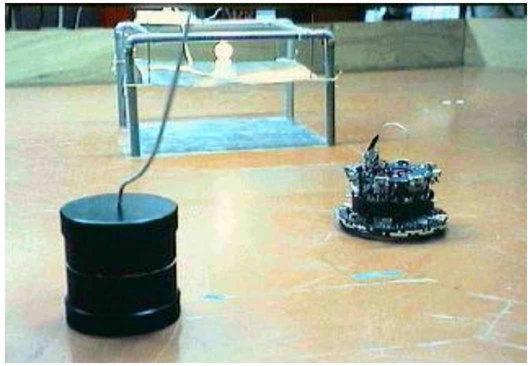
```
Initialization
    quantity  $LeftSensor \in [0, 100]$
    quantity  $RightSensor \in [0, 100]$
    quantity  $LeftMotor \in [-100, +100]$
    quantity  $RightMotor \in [-100, +100]$
    constant  $DEFAULT\_SPEED = +50$
    constant  $MAX\_CHANGE = 10$
```

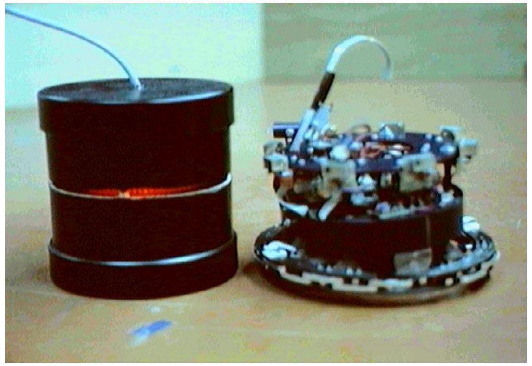


```
process(forward) {
    add_value($LeftMotor$,
        $-$value($LeftMotor$) $+ DEFAULT\_SPEED$)
    add_value($RightMotor$,
        $-$value($RightMotor$) $+ DEFAULT\_SPEED$)
}
```



```
process(taxis) {
    $b\_direction = \frac{\text{value}(LeftSensor) - \text{value}(RightSensor)}{SENSOR\_MAX}$
    $b\_intensity = \frac{\text{value}(LeftSensor) + \text{value}(RightSensor)}{2 \cdot SENSORMAX}$
    add_value($LeftMotor$,
        $-1 \cdot b\_direction \cdot (1 - b\_intensity) \cdot MAX\_CHANGE$)
    add_value($RightMotor$,
        $+1 \cdot b\_direction \cdot (1 - b\_intensity) \cdot MAX\_CHANGE$)
}
```





B-Scheduling on Ubuntu Linux

Implementation

- run a process at a fixed frequency (e.g., 100 Hz) under Ubuntu Linux (ideally directly in ROS)
- embed B-scheduling in this process (demo C code for B-scheduling is available)
- simulate the robot ecosystem (e.g., in Gazebo)
- implement the robot behavior processes with PDL (demo code is available)