OXFORD
BROOKES
UNIVERSITY
School of Engineering, Computing, and Mathematics
Module COMP7040: Autonomous Intelligent Systems, Semester 1 2024/2025

Resit Coursework: Localization and Navigation on the Anki Cozmo Robot

Objectives and Rationale

The objectives of this project are to build a working understanding of major issues in autonomous self-navigating robots including planning, localisation, and action. You will show the ability to build accurate sensor and motion models, construct usable robotic simulations, deal with uncertainty, navigate to goals using path planning, and deal with a complex task involving multiple goals in real time.

This coursework allows you to develop technical and experimental skills essential for industries related for example to autonomous cars, flying drones, or autonomous package delivery. It further introduces you to many active research areas and practical research skills important for both academic and research & development-based career paths. It also allows you to engage with an area in which robots and AI can actually save lives.

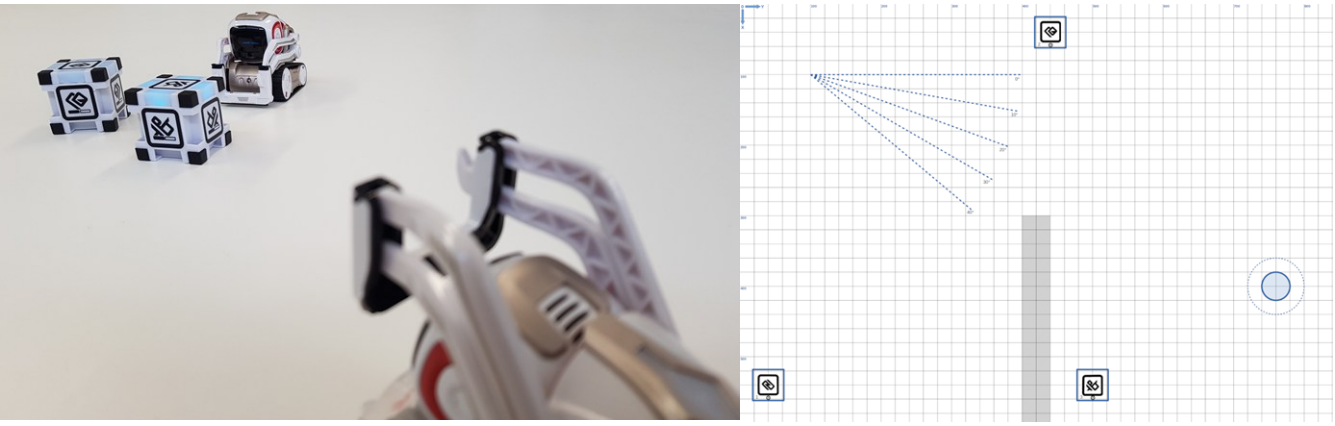
Learning Outcomes

1. Critically evaluate the requirements for the application of system intelligence in an autonomous system.
2. Collaborate in group to design and maintain hardware and software for an intelligent autonomous system.
3. Build independence, confidence and advanced professional awareness skills within the discipline including ethical, social, legal and risk considerations.
4. Synthesise and independently develop complex documentation to support autonomous systems created.

This is an *individual* coursework. The university's rules on academic misconduct including plagiarism and collusion apply to this coursework. For more information see https://www.brookes.ac.uk/students/sirt/student-conduct/academic-misconduct/

You are *allowed* to capitalize on any previous work on COMP7040 that was done by *your semester 1 coursework group*.

Introduction



This project will use the Anki Cozmo mobile robot.
This is a small, tracked vehicle with a camera able to recognise landmarks and faces, a cliff sensor to detect holes, and a lifter able to manipulate, lift, and hold objects such as the 'Light Cubes': boxes with an identifiable mark on them and lifting points for easy manipulation. You will develop against the Cozmo SDK in Python, and with an initially provided rudimentary simulator able to replicate basic physics and kinematics of the Cozmo robot and its environment.

This assignment provides a practice of representative challenges of autonomous intelligent systems. Your challenge is to navigate the robot to a known coordinate on a known map, but with unknown initial position and uncertainty of actuators and sensors on the way. In order to successfully navigate to the goal, the robot has to find out where it is and continuously update its belief from sensory data as it is moving. You will work with the basic kinematics and control of wheeled or tracked motion in order to control the robot along a desired direction. You will use the kinematics to tackle the robot's odometry, i.e. to predict the robot's movement based on observed track speeds. Visual sensing and sensor modelling will help to pinpoint the robot's location on the way. You will use the Monte-Carlo localization algorithm to integrate all of this information and make it of use for navigation towards a given target position.

Materials

The physical Cozmo robot will be available together with a physical test environment. Cozmo's SDK has in-built capabilities to detect and recognize markers on the cubes, as well as custom markers that will be placed on all obstacles. You can therefore assume that all relevant objects in the environment can be sensed, and that you have information about their position and orientation. However, sensors are never perfectly accurate, and a central challenge for your AI system is to cope with these inaccuracies and imperfections, and to model and resolve them.

You will be provided with an initial code base to develop and run your simulations. The code provides a way to specify simulation maps with obstacles (walls) and cubes to run the Cozmo simulation on, as well as basic visualization functionality. It further provides initial boilerplate code to develop sensor and motion models into, as well as room to run behavior.

Submission

The report will be submitted as single PDF file with no more than 4000 words through a dedicated submission inbox on the module's moodle page.

The submission deadline is **Monday February 24th, 5pm**.

All source code must be submitted through git/bitbucket. The code in the repository as such will not be marked, but is compulsory evidence of the achievement described in the report.

Report and assessment (100 marks, 4000 words limit)

1) Motion Model (25 marks)
   a) Implement the robot's driving motion model (the forward kinematics) based on the standard differential drive model (15 marks)
      i. This model should be implemented into the method 'track_speed_to_pose_change' in the file cozmo_interface.py in order to smoothly integrate with other parts of the code.
   b) Experimentally determine the model's wheel distance parameter. You should start off the physical distance between the tracks, approximately 50mm, and vary the value to find out which one creates the most accurate model predictions. (10 marks)
2) Probabilistic Sensor Model (25 marks)
   a) Experimentally determine a Gaussian probabilistic sensor model by estimating standard deviations in each dimension x/y/theta from suitable robot data recordings (15 marks)
   b) Implement a sensor model for the visual cube detection based on your findings that determines the probability value of sensing a cube at a given position (10 marks)
      i. This model should be implemented into the method 'cube_sensor_model' in the file cozmo_interface.py in order to smoothly integrate with other parts of the code.
3) Monte-Carlo Localization in Simulation (25 marks)
   a) Utilize the run-mcl.py template and visualization to implement the Monte-Carlo Localization algorithm based on the previously established motion and sensor models. (15 marks)
      i. Note that the file mcl_tools.py already provides implementations of suitable resampling algorithms such as 'low variance resampling'
   b) Experimentally determine suitable values for parameters of the algorithm such as the number of particles and critically evaluate the localization performance. (10 marks)
4) Robot-go-home challenge (25 marks)
   a) Design and implement a program on the basis of run-mcl.py that uses localization to drive the robot onto the designated target on the map ('home') from any initial position, while avoiding obstacles (in particular the trench in the middle). (10 marks)
   b) Experimentally assess the effectiveness and robustness of the developed solution (5 marks)
   c) Discuss the potential applications of the robotic skills developed in this coursework in society. What benefits as well as potential risks are there from a legal, ethical, environmental, and societal perspective? (10 marks)