Introduction to Robotics

Introduction

Short explanation of the two projects: Zumo and drawing robot.

Why are we here. Expose the setup. Remember we are all here to learn both teachers and students. The teacher is just a student with more experience, more mistakes made. To anchor this, ask the students to teach the teacher something (music basics for exemple). Show proof of experience from the teacher and explain that the course has been crafted to transfer that experience to students.

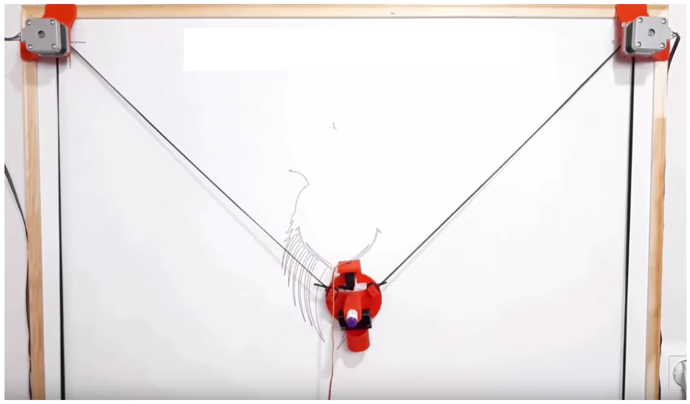
Aim of the course. Gain an understanding and remain aware of the thought process of robotics engineers and researchers, so you can use it yourself to solve new problems. What are they trying to achieve? What problems they face? What is their approach? Why and how did they came up the solutions in use today? It is people, just like you and me that made robotics the way it is. It has been constructed with some aim in mind. Understanding those aims is 50% of the job.

Open discussion & deconstruction. What is a robot? Interactive web search guided by students of existing robots and videos of what they do. Analysis of requirements to do what they do (sensors, actuators, planning, etc). How does the sensors and actuators works? Briefly get down to the basic physics of it.

Discovery of the Zumo robot. Manipulating the robot unpowered. What sensors, actuators are on the robot? How do they work, physical principle behind it? What could we imagine doing with it?

Split class in groups. Advanced students build their own robot, project based. Beginners form groups of 2-3 with one person in each group that can already coded in Arduino.

Drawing robot project



Pitch them the objectives and give them the box of components. First day objectives is to make a short presentation of the objectives and plans to the other students (must include the why of the robot, explanation of technology used (stepper motors, servomotors, arduino, python), explanation of problem to be solved (forward and inverse model)). Project in full autonomy with updates each day to me.

Links:
- https://github.com/croningp/commanduino
- https://github.com/croningp/Arduino-CommandTools

Zumo robots to learn the basics of robotics

Goal. Use the platform to expose problems in robotics, experience them, think of solutions, implement these solution naively, test if they work, challenge them, explain how it can be done better, implement, test. Discuss how this problem is still an active area of research, just more advanced and building on 40 years of people proposing solutions.

Install the tools and provide resources:
- User manual: https://www.pololu.com/docs/0J63
- Install library: https://github.com/pololu/zumo-32u4-arduino-library#installing-the-library
- Install board: https://www.pololu.com/docs/0J63/5.2
- Pololu library: https://github.com/pololu/zumo-32u4-arduino-library

Teacher resources and code: https://github.com/jgrizou/CRI-UE-Robotics

Motor control (PWM + registers)

Problem. I want my robot to move at different speeds. Show expected behavior.

Approach. Test single motor with variable power supply. Batteries have fixed voltages + change of voltage is hard and energy inefficient. Ask how to change speed anyway.

Programming. Concept of average, speed is just the time to reach a destination, we can go fast for a bit and pause for a bit, and repeat. If I need to move from A to B in 10 seconds I can pause along the way, if in 1 second I have to do shorter pauses. Test first program and progressively reduce the pause duration.

Concept to understand. PWM - Pulse Width Modulation

Hidden problem 1. How does the processor turn motor on and off? Transistors. How can we change direction? H-Bridge

Hidden problem 2. How can the robot generate PWM so precisely? We need a measure of time! How does the robot measure time? Big problem of humankind btw. It is always physics based. For Zumo: https://en.wikipedia.org/wiki/Crystal_oscillator.

Hidden problem 3. How can I generate my PWM while doing other things? Like waiting for user inputs, doing heavy math operations. We are in a loop and all our actions are taking time. Brainstorm solutions.

Explain solution. Having dedicated parallel circuits acting like “automated slaves”. Their only goal is to produce PWMs. But how do we talk to them and tell them the frequency or duty cycle? Introduce **registers.**

Demonstrate solution in class. Making music at different rhythm in class. Students are the slave. One student is the time telling device, hand clapping at a fixed rhythm. Define a non-verbal communication protocol constrained by how computers represent data (they will reinvent binary registers). The register says which instrument to use and at what frequency (every N bits of the main time telling device). Ask a musically literate student to create a simple beat using this interface, he is the main loop, they are the slaves (and one student is the time).

Expand idea. LED control of brightness using PWM.

Odometry, Encoders and Interruptions

Problem. How can a robot know where it is? How can I move the Zumo robot exactly 10 cm in a straight line? Show expected behavior.

Approach. First approach is to move the motors for a limited amount of time, which they can compute using a calibration curve. Then ask them to change the motor speed. How to generalize? What is the physical variable to take into account? Not time but the number of rotation of the wheel.

Hidden Problem 1. How can we track the rotation of the wheels? Open question for them to invent their own solution.

Explain encoders. A signal is created X times per rotation. Could be physical holes blocking a led/laser signal to get through, or a magnetic counter, or physical spike hitting a button, whatever you can think of. Draw the shape of signal when the wheel rotates.

Programming. Zumo use magnetic encoders, explain the principle of hall effect sensor. Explain simplified wiring like if there was only 1 sensor. Code a counter in the main loop, print on screen, test.

Hidden problem 2. Show that we miss events if we do something else in the main loop. How can I count event outside of main loop?

Explain solution. Similar explanation that register but for interruption. I need to draw a shape on the whiteboard while counting the number of times people behind me raise their hands. I ask a guy to tap my shoulder to interrupt my work, I look around and count the event by adding a counting bar on a dedicated area of the screen.

Programming. Test interrupt and compare with counting in the main loop.

Hidden problem 3. How can we know the direction of rotation of the wheel? Show that our counter only increment, whatever the direction of the wheel. Let students brainstorm ideas.

Explain solution. Two sensors carefully placed, in one direction sensor A follows sensor B, in the other direction it is the reverse.

Programming. Explain the wiring and the use of the XOR function. Learn how to decode and get back the original signals. Decode to get accurate positioning accounting for directionality.

Hidden problem 4. How to deal with friction, slips, non flat surfaces (use example of rolling a coin around another is rotating twice on itself, not just once)? Student brainstorm.

Explain possible solution. Free-wheel that is smaller and independent from the driving wheel. Or other systems, GPS, indoor GPS, SLAM + quick explanation of how these methods work.

Feedback Loops

Problem. I want to react to my environment. How can the robot follow a line? How can the robot avoid obstacles? How can the robot keep its heading despite disturbances? Show expected behaviors.

Approach. Students brainstorm to see they are all the same kind of problem + rough introduction to feedback loops.

Line following

Problem. How can the robot follow a line?

Approach. Inspect the robot line sensors. How they work. Logic of robot reaction wrt sensed colors.

Programming. Learn to read sensors value. Decide black/white threshold. Program the behavior.

Additional Problem. Road crossing. What happen if we try to drive on a 8-shaped circuit? How to deal with it? What additional information is needed? Open on self-driving cars.

Object Avoidance

Problem. How can the robot avoid obstacles?

Approach. Inspect the robot proximity sensors. How they work. Logic of robot reaction wrt. sensed obstacles.

Explain. Sensors specificity (binary response to specific frequency)

Hidden Problem 1. How to know if an object is close or far with a binary on/off sensor ?

Solution. Check response with varied intensity of the light source.

Programming. Learn to read sensors value at multiple range. Decide relevant threshold for the problem. Program the avoidance behavior.

Additional Problem 1. How do you follow the obstacle instead?

Additional Problem 2. How do you map all the surrounding obstacles for self-driving cars deployment in our streets? LIDAR (basically a rotating proximity sensor) + machine learning to identify object/cars/people + their predicted trajectories.

Gyroscope

Problem. How can the robot keep its heading despite disturbances?

Hidden problem 1. How do we measure the heading of the robot? Earth magnetic field. Or gyroscopic effect. Explain MEMS sensors.

Hidden problem 2. The gyroscope can only give us the rotational speed at a specific time when we read it. How do we get back the heading from it?

Programming. Learn to read sensors value. Implement the integration from speed to position. Program the behavior.

Additional Problem 1. Demonstrate drifting. How to maintain the heading for a long time? How to get absolute heading? How do plane maintain their heading ? Answer: very precise + multiple gyro + recalibration with magnetic north in stable regime + GPS triangulation -> explain sensor fusion

Additional Problem 2. How were gyroscopes implemented and their measurement fed into electronic circuits before MEMS?

Final Project

Drawing Robot. Full presentation of the robot. Demo of drawing. Self-assessment of struggling point. What improvements in the next version. What they would have done differently. Live inspection and explanation of the code.

Zumo. Sumo competition. Two robots on the field. Only one should remain. Each team code their own robot and we will compete with others for the final exams. All resources allowed. Use of the default library strongly recommended for sensors and actuators control.

Evaluation. Being able to explain the logic of their program. Being able to explain the behavior of the robot in various scenarios. Being able to explain what the code does.

Demonstration day. Morning of 12th of November.

Evaluation

The same evaluation grid will test how students understood the concepts we introduced. It will be filled by:

- Self - Students asked to assess what they think they know
- Peers - Peer student groups ask questions to group being tested and assess their work. This is meant to train students being on the other side and evaluating acquired knowledge of somebody else.
- Teacher - Teacher assessment based on discussion.

All three weighted equally unless obvious and unfair arrangements made between parties. The goal is learning not scoring.