4SC020 Embedded Motion Control

Minutes of data gathered on May 8th 2019, during the group presentations

To do

1. Cleanup

There's a big mess in Git, in script naming, and in variable naming. This should be altered for easier use and accessibility. A .gitignore file should be created for easier management.

Homework assignment (Bram)

Do script and Git cleanup by May 8th, as well as the .gitignore file, work on a naming structure for variable cleanup.

2. Splitting

Split functionality into main (world model), perception, monitoring, planning, and control.

Homework assignment (Bram)

First, determine what high-level functions perform what functionality, then split accordingly.

3. Finite state machine

Theory

The theoretical aspect of the FSM should be determined, preferably visual (i.e. flowchart w/ arrows). Make sure this covers all possible states, no dead- or livelocks, and switch events.

A single-variable FSM is advised, as multivariable state machines quickly become very complicated. For now, it should remain manageable, but we might

Implementation

Unsure how best to implement this into a program. The example program from the course contains a working FSM, I believe.

Homework assignment (Marcel)

First, work on a theoretical model of the finite state machine for the escape room challenge. Review this with the group often. Once done, implementation can be attempted. If time is left, the FSM can be extended for the hospital challenge.

4. Perception

Blind corners

Martijn's method of determining an inner corner based on instant value shifts should be further developed and added to the door detection script.

Exit hallway scanning

A method should be found of scanning the exit hallway and finding the final points. Martijn already had the idea of scanning inward from both sides and finding range discrepancies.

Homework assignment (Martijn & Ruben)

Do above.

5. Exiting

New state

A new state should be defined that will be used when exiting the room.

Find exit

From the perception script, the exit corners of the hallways shall be input, from which an exit location slightly outside the hallway shall be calculated, and movement started.

Homework assignment (Jeroen)

Do above, the robot should be able to fully exit the room.

6. Location updating

It might be preferable to use sensor data to verify the robot's current position against the encoder-driven algorithms. A new trajectory can then be generated or the current position updated.

7. Safety

Collision

Collisions should always be avoided. Avoidance is preferred in the end, but for now a stop command is fine.

Unexpected objects

It might occur that from one frame to the next, a near object is found. Actions should be taken in this instance.

Nearing walls

PICO should be able to near walls, but if possible steering action away from the walls should be taken. Perhaps simply checking range left/right and steering away if too close, by adding or removing speed from the trajectory.

Homework assignment (Bram)

Start on basic STOP functionality during unsafe situation.

8. Data storage

It will probably be useful to store data such that the world map is updated, and information on what is behind the robot is available for use. This is not necessary for the escape room challenge, but it could easily become relevant during the hospital challenge.