Minutes 2019-06-03

Performed work

1. Avoidance (Bram)
    a. Full-view avoidance
    b. Issues with stopping PICO inside door
2. Avoidance (Marcel)
    a. Updated algorithm with more failsafes
3. Localization (Jeroen & Martijn)
    a. Very difficult to compute
    b. Worked, most of the time.
4. Localization (Jeroen & Bram)
    a. Range-based localization
    b. Too many matches? Perhaps multiple good fits?
    c. Requires map fitting with results.
5. Path planning (Ruben)
    a. Now works with orientation to next point
        i. More priority should be put on rotation action
    b. Map walls are blown up per-line
6. Trajectory cycling (Ruben & Bram)
    a. Cycles to next point when near.

Questions for Bob

1. How to easily create .json map?
    a. Use snap-to-grid in Inkscape with blocks, manually input to .json
2. What's a good alternative for this localization?
    a. Our methods seems fine. Door finding algorithm might also be used.
3. What localization algorithm would you recommend?
    a. Particle filters are often used, but can easily fail.

To do

1. Finish localization (Bram & Jeroen)
    a. Finetune algorithm
    b. Reposition map
2. Compare avoidance algorithms and choose most suitable (Marcel & Ruben)
3. Finetune dynamic obstacle avoidance (Marcel & Ruben)
    a. Fix stopping in doors
    b. Recognize locking situation
4. Map updating (Martijn & Marcel/Ruben)
    a. Add obstacles to obstacle map
    b. Remove obstacles that aren't present anymore
    c. Add visualization w/ different color
5. Finetune control (Marcel/Ruben)
    a. Rotation priority
6. Presentation (Bram)

    a. Assumptions for localization
    b. State machine
    c. Localization
    d. Planning
    e. Avoidance
    f. World model? (PLAATJE)
7. Update Wiki (Jeroen)
8. Recognize moving objects (Later)

PICO Test 2019-06-05

1. Localization robustness
2. Dynamic obstacle avoidance robustness
    a. Move through doors
    b. Move through hallways
    c. Avoid moving objects
3. Static obstacle avoidance
    a. Recognize issue
    b. Plan way around
4. Limits of system (later)

Agreements

1. Parameter tuning
    a. Create TUNE: label on tunable parameters