



The Thytanic Navigation

Students:

Alessio Desogus
Antoine Bachmann
Adélaïde Pinel
Ramon Heeb

Map generation

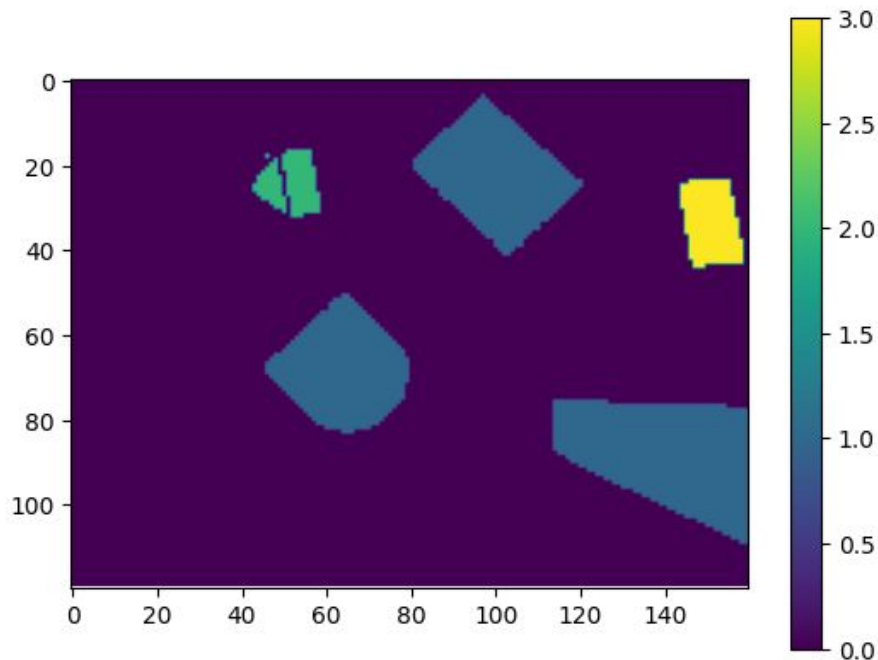
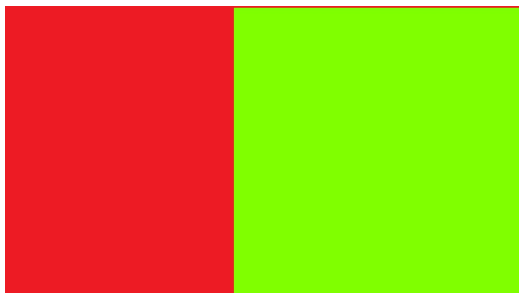
- Thresholding in HSV color space

Thymio position

- Average x, y values of all thymio pixels

Thymio orientation

- Two colors of paper on front and back of thymio
- X and y positions for front and back
- Get orientation from that



Vision post-processing

- Uneven opening to remove noise and diminish obstacle detail, while giving fine control
- Downsampling for ease of computation and navigation, due to Thymio's limited movement precision, pixels much smaller than $\sim 1\text{cm}$ are not useful

Pathfinding

- Obstacle dilation to account for robot size, mostly equivalent to accounting for size in the exploration algorithm, but much more efficient computation-wise
- A* algorithm using the euclidean distance from the goal, simple trapless trips make it much more efficient than dijkstra (up to 50x in testing)

Steps :

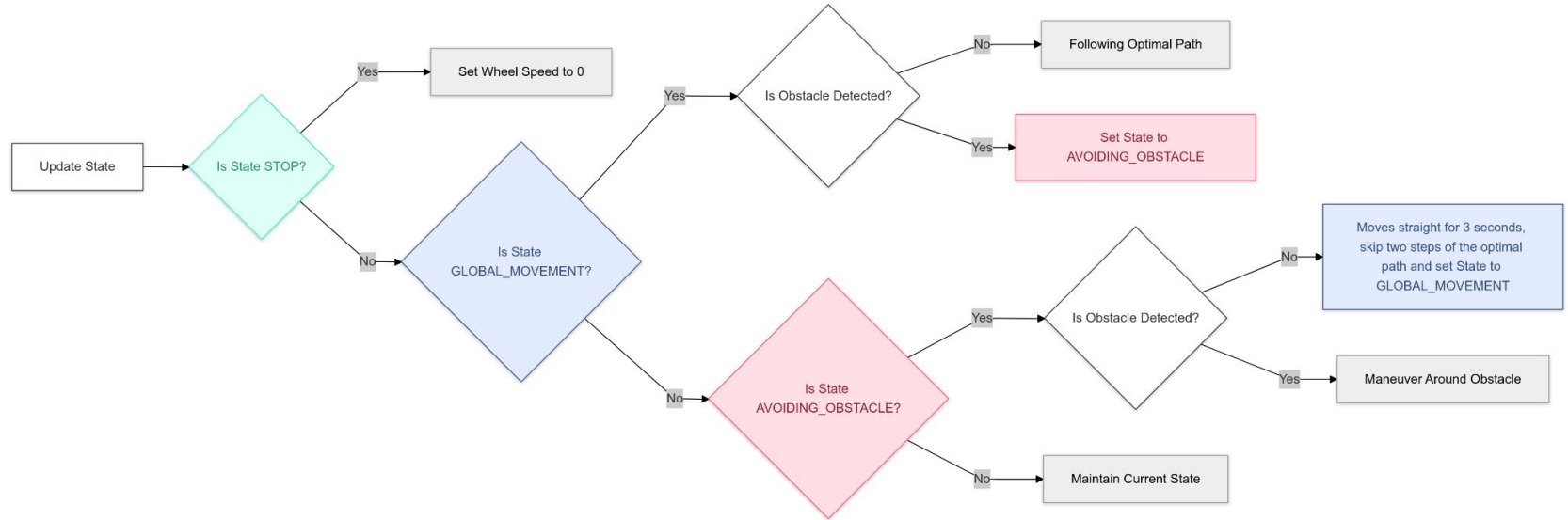
- Variances measurements of camera: taking several frames of the robot and average
- Variance measurement of velocities: setting both motors to the same velocity
- Preprocessing nonlinear outputs: calculate the velocities according to wheels velocities measurements.
- Apply a standard Kalman filter works then because:
 - nonlinear effects are small and bounded (-1 and 1)
 - the update rate is high, so nonlinear deviations are small between updates

2 situations :

- Camera is available: we can estimate the states based on all states measurements.
- Camera is covered or it could not provide one of its measurements (x, y or theta): the robot is estimating his position only with the velocity measurements, such that it doesn't do sensor fusion anymore and it gets less precise.

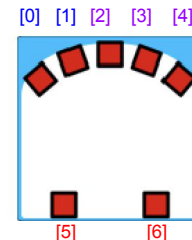
State Machine with 3 States:

- GLOBAL_MOVEMENT
- AVOIDING_OBSTACLE
- STOP



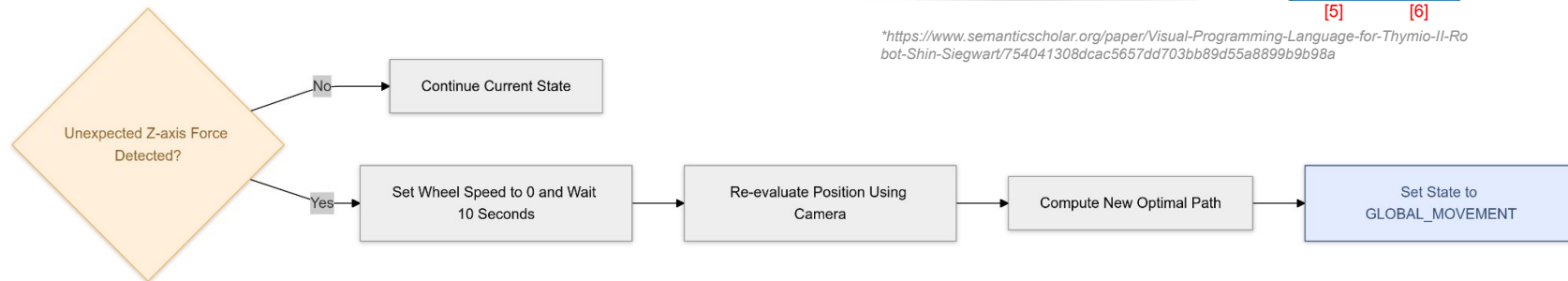
Obstacle Avoidance:

- Using the 5 forward **prox.horizontal** sensors for detecting (with a predefined threshold)
 - Turning **TRIBORD** (right) if any(value > self.detection_threshold for value in sensor_readings[0:2])
 - Turning **BABOR** (left) if any(value > self.detection_threshold for value in sensor_readings[2:5])
 - Turning until no values are above the threshold



**<https://www.semanticscholar.org/paper/Visual-Programming-Language-for-Thymio-II-Robot-Shin-Siegiwart/754041308dcac5657dd703bb89d55a8899b9b98a>*

Kidnapping Scenario:





Questions & Feedback ?

Students:
Alessio Desogus
Antoine Bachmann
Adélaïde Pinel
Ramon Heeb

