Discrete Filters - Exercise: w/ Orientation

What if grid-robot | has orientation
available controls change to → MOVE_FORWARD, MOVE_BACKWARD, ROTATE_LEFT, ROTATE_RIGHT

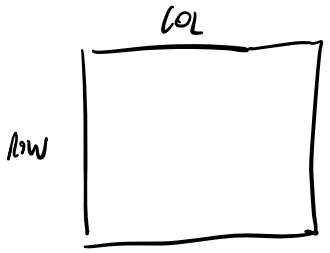
- How does the state change?
- What about the observation and transition model?

RBS - Resolution

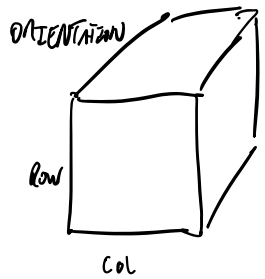
Scenario: GRID-ROBOT

Map → | plain
M x N matrix
1: occupied
0: free
⇓
MAPS / MAP.TXT

Originally: grid-robot lived in 2D grid world ⟹ Now: 3D grid world {North, South, East, West}



STATE = {ROW, COL}
|||
POSITION

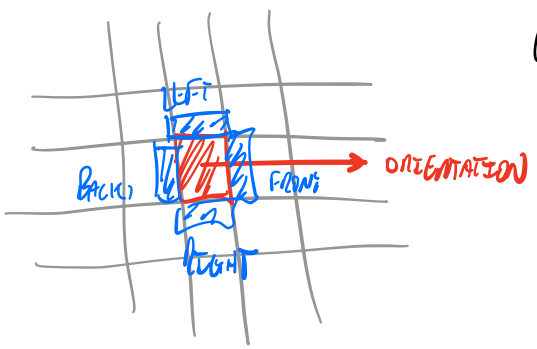


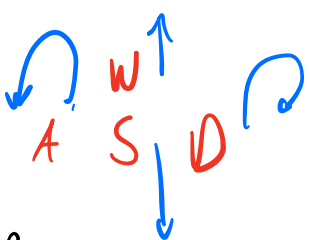
STATE = {ROW, COL, ORIENTATION}

(still, discrete one) 1 … M | 1 … N | N S E W

4 commands: MOVE_FWD, MOVE_BWD, ROT_LEFT, ROT_RIGHT

Seems with 4 bumpers

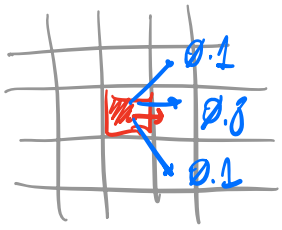


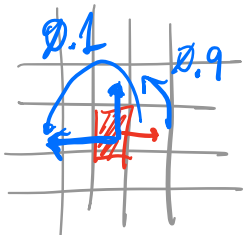


3D Transition Model: transition_probability_tensor = transitionModel(map_, row_from_, col_from_, theta_from_, control_input_)

- constraints
  - move only adjacent cells → 0 probability if not
  - $|\Delta theta| \leq 90°$
  - invalid motions: if obstacle encountered / outside of the map ⟹ translation = 0 ↷ p() = 1.0; ELSE ↷ p(\) = 0.0

- model





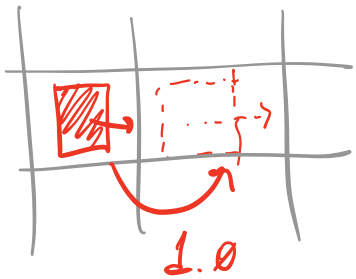
MOVE_FWD / MOVE_BWD

↳ but maintain orientation ...?

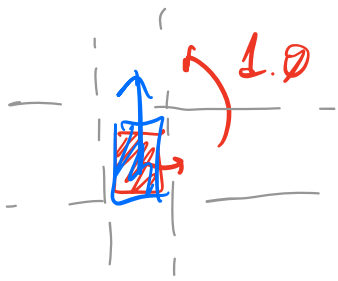
ROT_LEFT / ROT_RIGHT

↳ but maintain position ...?

↳ no / noise

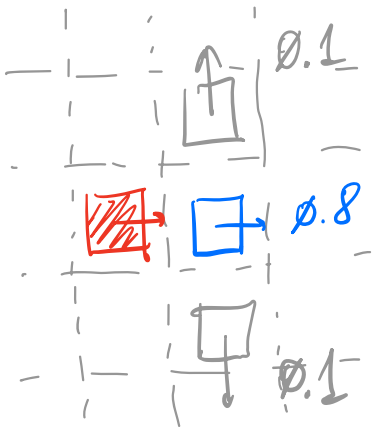


MOVE_FWD

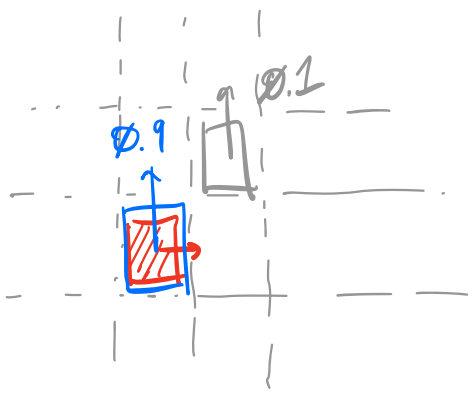


ROT_LEFT

↳ with noise ?...



MOVE_FWD



ROT_LEFT

TRANSITION_PROBABILITY_TENSOR = TRANSITION_MODEL(MAP, RAW_FROM, TOL_FROM, THETA_FROM, CONTROL)

map_rows = rows(map)

map_cols = cols(map)

transition_probability_tensor = zeros(map_rows, map_cols, 4) # possible orientations $(\in -\pi, -\pi/2, 0, \pi/2?)$

FOR row_i = 1 .. map_rows

For col_o = 1 .. map_cols

For theta_idx = 1 .. 4

IF (translational_rows ≥ 1 || translational_cols ≥ 1)

└ CONTINUE → ∅ probability

IF (|delta_orientation| > π/2)

└ CONTINUE → ∅ probability

switch (control_input)

MOVE_FORWARD:

switch (theta_from):

∅: target_col ++;

π/2: target_row --;

-π: target_col --;

-π/2: target_row ++;

MOVE_BACKWARD:

└ ...

ROTATE_LEFT:

theta_target = (theta_from + π/2)

wrapToPi

....

// Each possible motion

switch (control_input):

MOVE_FWD:

switch (theta_from):

∅: if (translation_rows == ∅) && (translation_cols == 1) , ∅.8

...

- - -

4) **OBSERVATION** : observation_probability = OBSERVATIONMODEL(map_, row_, col_, theta_, observation_)

now the complete ←

cell occupied ⇒ observation

also depends on current theta_

4) **STATE BELIEF** : now also depend on theta

AHHH → they assumed the noise-free model!

↓

that is why when probability become 1.0, it never decrease / uniform / increase uncertainty ....

(+)

marginalize out theta to obtain 3D matrix → visualize probability of being in a cell independently of the orientation

get Next State → USE CUMULATIVE PROBABILITY to get random samples from probability distribution