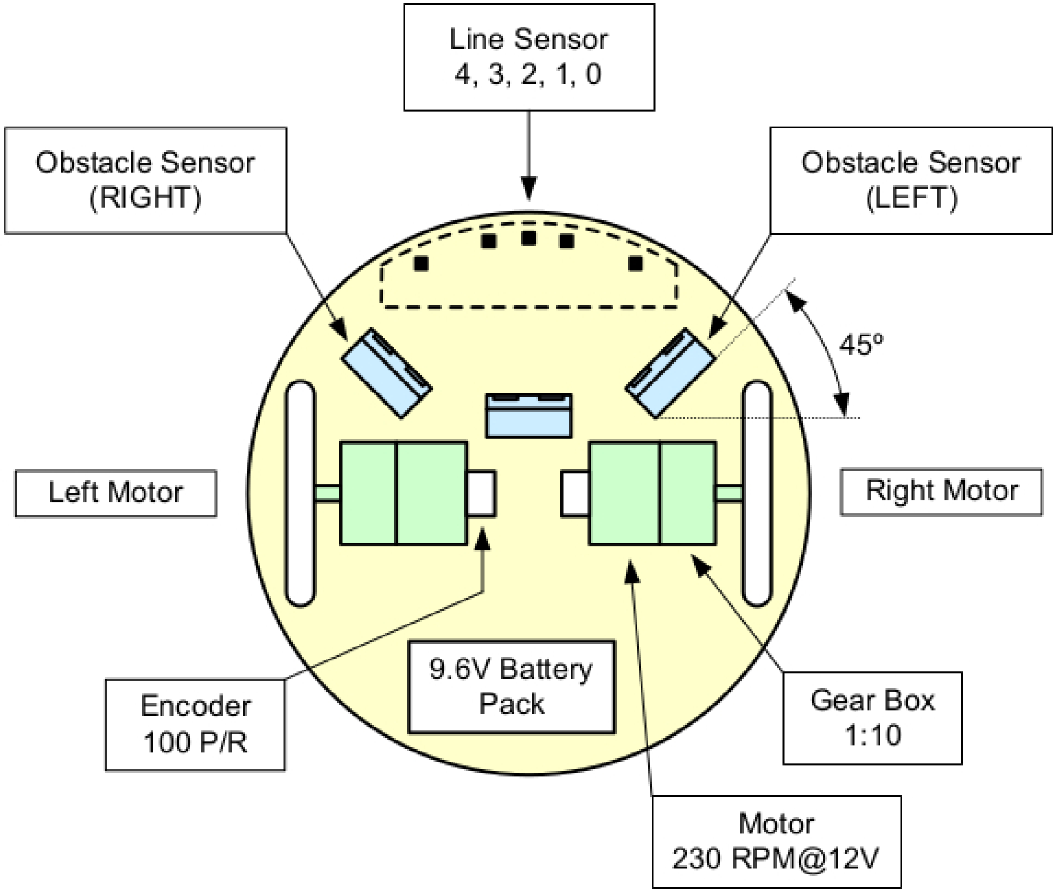
Development of an agent for a robot that solves labyrinths

Intelligent and Mobile Robotics - 2nd project

Vedran Semenski, Aveiro, November 2014.

Intro

- C
- Basic text editor
- Robot
- pcompile



Basic concept

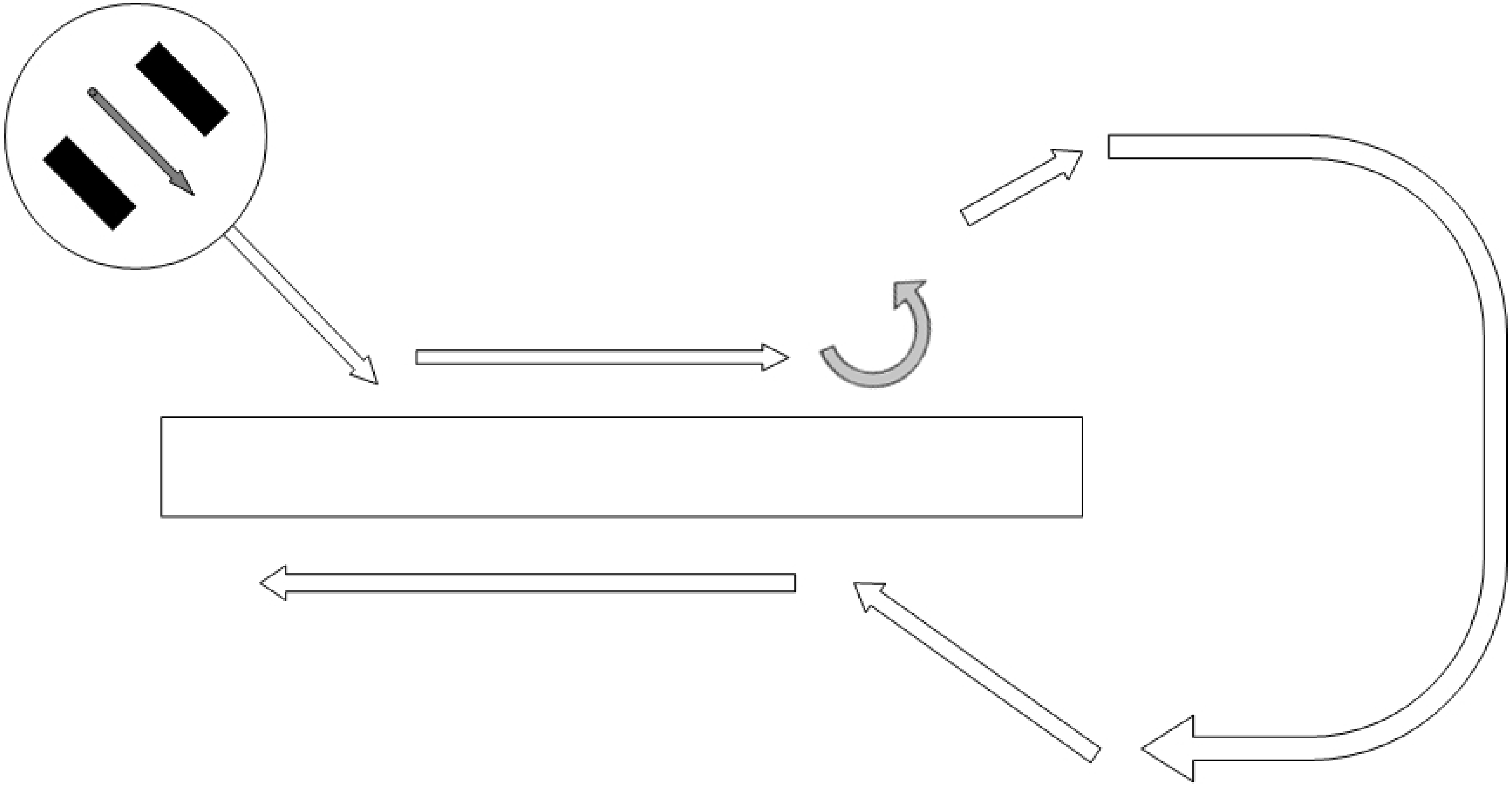
- Abstracting the robot
  1. Movements
  2. Sensors
- Subsumption Architecture
  - Prioritized list of behaviors (dinamic)
- Behaviors

Behaviors

1. Stop at beacon
2. Avoid collision
3. Follow the beacon
4. Follow the wall
5. Wander
6. Stop at starting position
7. Return to starting position (avoid loops)

- Beacon sensor servo

Behaviors – follow the wall

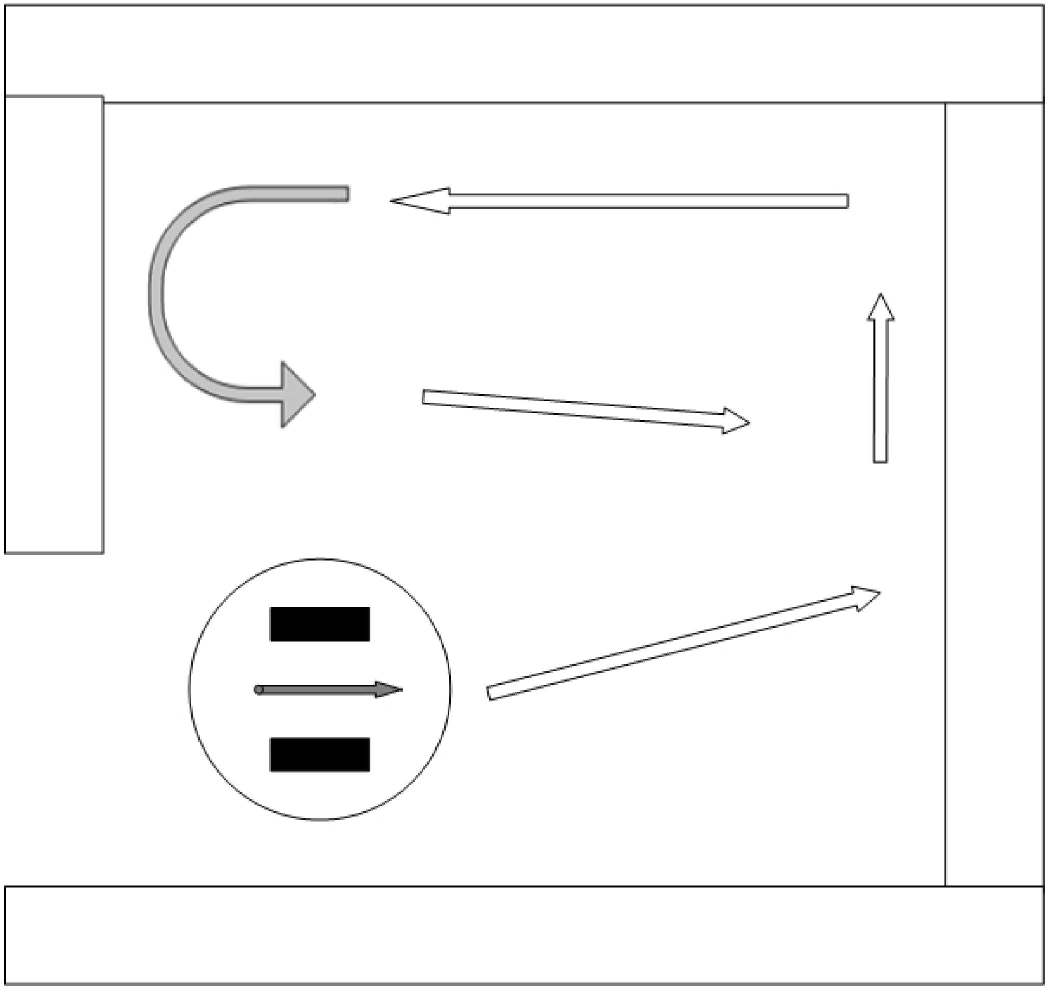


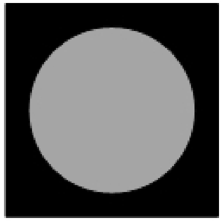
Dinamic proprity list 1

Default (starting) list

1. Stop at beacon
2. Avoid colision
3. Follow the beacon
4. Follow the wall
5. Wander

Problem



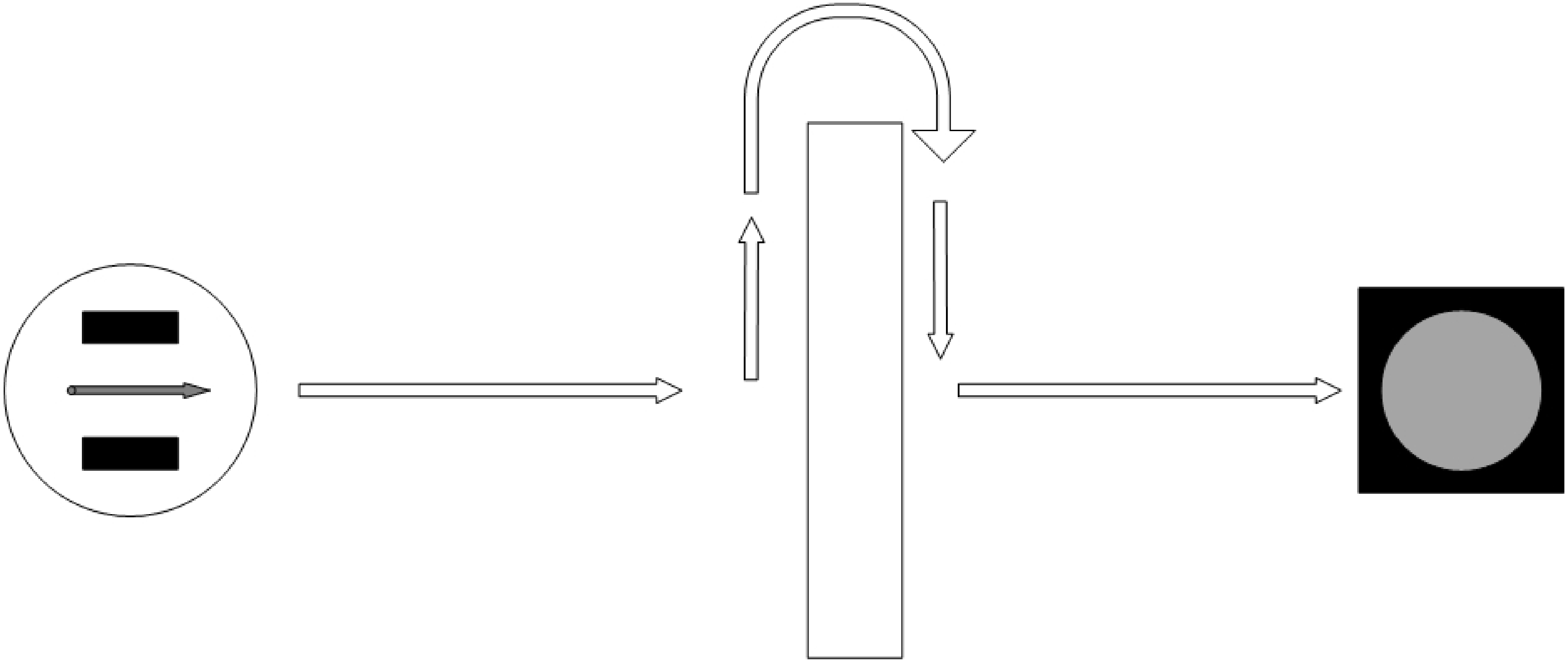


Dinamic proprity list 2

After promoting

1. Stop at beacon
2. Avoid collision
3. Follow the wall
4. Follow the beacon
5. Wander

Result



Dinamic proprity list 3

Returning priority list(s)

1. Stop at starting position
2. Avoid collision
3. Return to starting position
4. Follow the wall
5. Wander

1. Stop at starting position
2. Avoid collision
3. Follow the wall
4. Return to starting position
5. Wander

Workflow

- BasicWorkflow:

1. Initialisation
2. Start loop
   1. Refreshing sensor readings
   2. Testing behaviors
   3. Execution of behavior with highest priority
      1. Update priority list/promote behavior/ demote behavior
   4. Check if finished
3. End

Conclusion

Advantages:

- Simple
- Good results
- Flexible
- Fast

Limitations

- Set and forget
  - learning
  - adaptability
- Limited improvement options