Integrating BDI and Reinforcement Learning: the Case Study of Autonomous Driving

Michael Bosello

Università di Bologna – Department of Computer Science and Engineering, Cesena, Italy

Supervisors: A. Ricci, G. Pau

Introduction

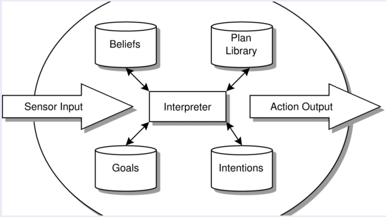
Objective

- We want to exploit ML in agent-oriented programming

⇒ BDI-RL integrations

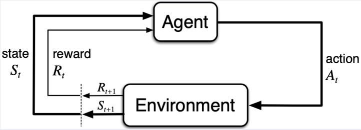
BDI

- Model for cognitive agents
- Belifs, Desires, Intentions
- Agent cycle
  - → deliberation
- Hard-coded plans
  - → means-end reasoning



Reinforcement Learning

- Agent learns by interacting with the environment
- trial and error
- States, Actions, Rewards
- Estimations based on experience



Introduction

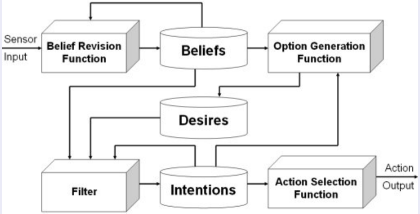
Objective

- We want to exploit ML in agent-oriented programming

$\Rightarrow$ BDI-RL integrations

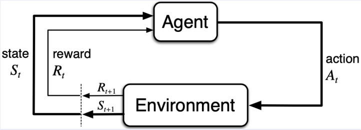
BDI

- Model for cognitive agents
- Belifs, Desires, Intentions
- Agent cycle
  - $\rightarrow$ deliberation
- Hard-coded plans
  - $\rightarrow$ means-end reasoning



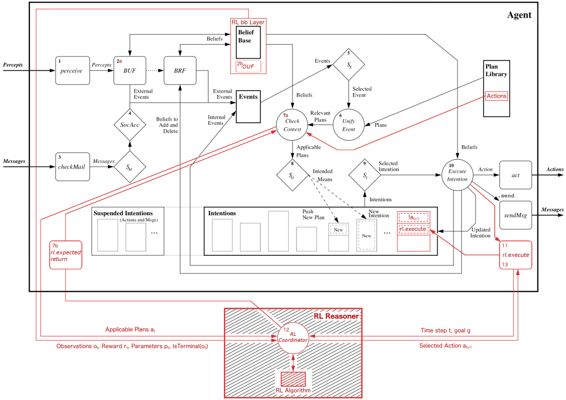
Reinforcement Learning

- Agent learns by interacting with the environment
- trial and error
- States, Actions, Rewards
- Estimations based on experience



Proposed framework

- BDI-RL integration
- The developer can:
  - write some plans — ***Hard Plans***
  - let the agent itself learn other plans — ***Soft plans***



Contribution: Jason-RL II

Model

- It represents the general concepts of RL
- It abstracts from the specific algorithm and problem

Jason implementation

- Notion of soft-plan (with independent contexts)
- Definition of states/actions/rewards/termination for each soft-plan

```
rl_parameter(policy, egreedy).
rl_algorithm(follow_street, dqn).
rl_observe(follow_street, lidar_data(list(1080))).
rl_terminal(follow_street) :- crash.
rl_terminal(follow_street) :- new_position.
rl_reward(follow_street, R) :- reward(R).
rl_reward(follow_street, 50) :- new_position & position(P) & target_point(P).

+!follow_street : target_point(P) & position(P) <-
    move("stop");
    .println("reached ", P).

+!follow_street : starting_point(P) & position(P) <-
    rl.execute(follow_street);
    !follow_street.

@action1[rl_goal(follow_street), rl_param(direction(set(forward, right, left)))]
+!move(Direction) <- move(Direction).
```

Case study

- Autonomous Driving as case study
  - To assess the advantages of the BDI-RL integration

The agent has to

- follow high-level directions
- navigate without incidents

- RL struggles in *temporally extended planning*
- Fine-grained navigation is an hard-to-engineer behavior

⇒ Hard-coded plans handle the high-level planning

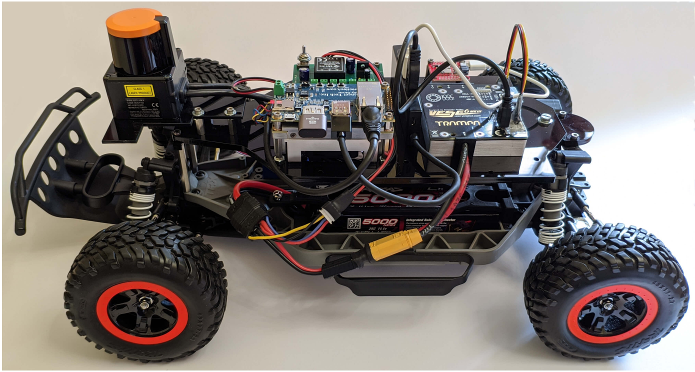
⇒ Moving without incidents is achieved by RL

F1tenth platform I

The same code can run on:

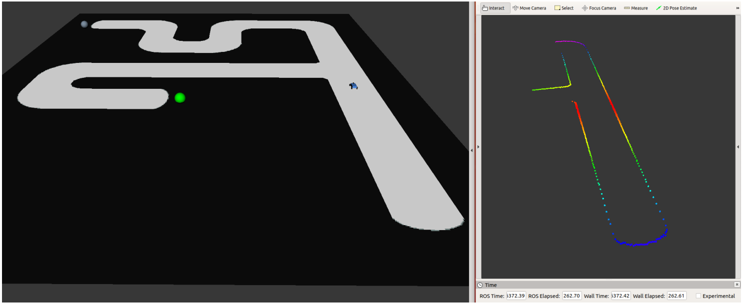
- A *very realistic* 1/10 scale car prototype
- The ad-hoc simulator

Hardware/software stacks similar to full-scale solutions



- Main experiment in the simulator
- Real car experiment in a simple track

F1tenth platform II

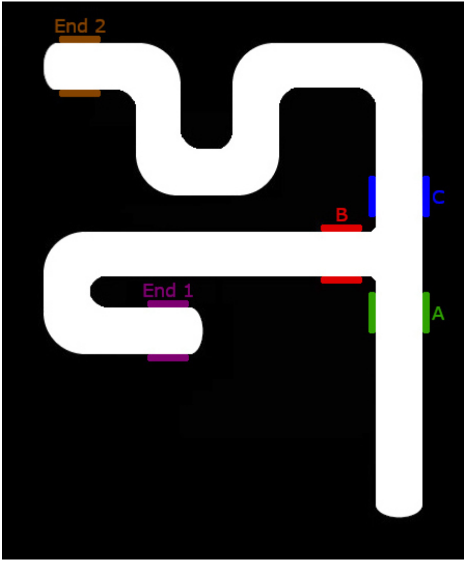


- ROS (Robot Operating System)
- Sensor node: Lidar, odometry
- Actuator node: VESC

Control node

- Sensor data and actuator commands updated asynchronously
- Automatic emergency braking
  - It has priority over agent decisions
  - The car goes backward before returning the control to the agent

Environment

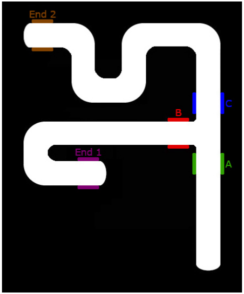


- Percepts
  - LIDAR vector
  - Position label (thanks to odometry)
  - Target label
  - Emergency braking activated
  - Position label has changed
  - Environment reward
- Driving actions
  - Go forward
  - Turn right
  - Turn left

  ⇒ Fast learning
- Environment actions
  - Reset to position
  - Ask for new target
- Environment rewards
  - Emergency braking = -1
  - Proportional to
    - ★ car velocity
    - ★ distance to the nearest obstacle

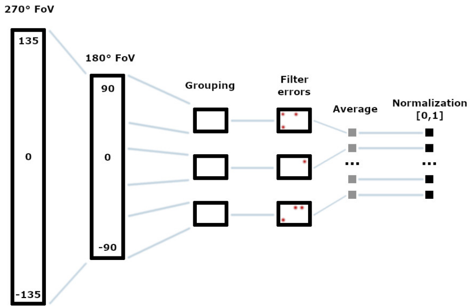
BDI agent

- Two hard-plans that defines the high-level directions
  - They plan the moves to `END 1` and `END 2`
  - Defining sub-targets and self-rewards
- Three soft-plans that manages navigation in different situations
  - `follow_street`, `turn_left`, `go_forward`
  - Same code
  - Trained in different conditions and with different goals
  - ⇒ Distinct behaviors
- Four hard-plans for each soft-plans that handle the soft-plans outcome.
  - They govern the various situations and the learning cycle
    - ★ Emergency braking activation
    - ★ Wrong direction
    - ★ Target reached



RL software

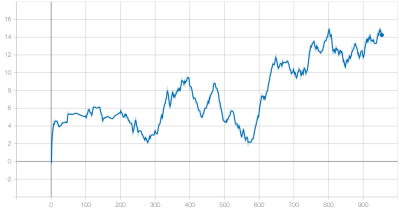
- DQN
  - Standard enhancements: experience replay, state history, target net
- LIDAR pre-processing

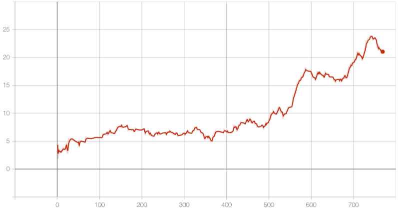


- Significant effort in:
  - Environment engineering
  - Reward shaping
  - Prameters tuning
  - Functions choosing

Intersection experiment results

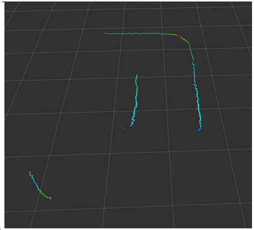
- The agent can reach its target thanks to the integration of soft and hard plans
- Case study considerations
  - Problem that requires both time-extended planning and learned policies
  - Long-term plans are achieved thanks to the BDI reasoning
    - ★ Hard-plans manage planning and soft-plan failures





- follow_street: orange
- turn_left: blue
- go_forward: red

Simulation to real

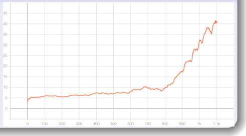


Use of LIDAR

- LIDAR measurements greatly affected by reflection
- Training of RL agent with real LIDAR data is considered an *open problem*

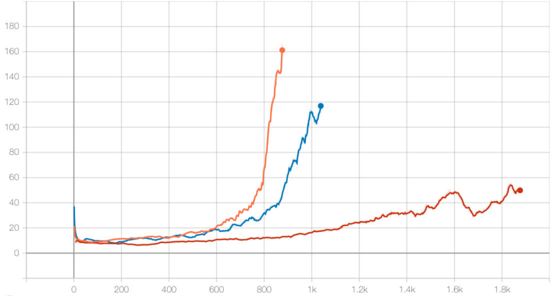
Real car results

- The driver-agent successfully learned a control policy
  - using real LIDAR data
- 1D CNN used to process LIDAR data for the first time



NNs comparison

- CNNs perform better on structured and spatially related data
- 1D CNNs are very effective in processing LIDAR data



- 1D CNN: orange
- Dense: blue
- 2D CNN: red

Demo

Intersection experiment demo

Real car demo

Discussion & Conclusion

- Inclusion of soft-plans in the **reasoning flow** of the agent
  - Automating the imperative side
  - Preserving the declarative side

Advantages wrt end-to-end RL

- RL struggles in temporally-extended planning
  - A plain RL agent will need more training time
  - It may not be able to learn such behavior at all
- Multiple RL contexts
  - Coordinated inside the same agent
  - To achieve complex and time-extended tasks
  - Use of the most appropriate RL algorithm for each sub-task
    - As multiple RL algorithms can coexist

Next steps

- Intersection experiment with the real f1tenth
- BDI-RL vs plain RL *quantitative* comparison