Master of Science Topics

Robotics and Human-Robot Interaction

Title: Teaching robots with adversarial imitation learning

Coordinator: *Prof. Adina Magda Florea (adina.florea@upb.ro)*

Description:

Imitation learning is the process by which one agent tries to learn how to perform a certain task using information generated by another, an expert agent performing that same task. Recently, the specific problem of imitation from observation (IfO) has received much attention. In this case the imitator only has access to the state information (e.g., video frames) generated by the expert. One category of methods is represented by adversarial approaches to IfO, e.g., the generative adversarial imitation learning (GAIL) algorithm [1], which learns explicitly how to act by directly learning a policy and draws an analogy between imitation learning and generative adversarial networks. The approach was further developed in [2] where the authors propose a method for learning joint reward policy options with adversarial methods based on Inverse Reinforcement Learning (IRL). The method can learn underlying reward functions and policies solely from human video demonstrations.

Tasks:

- investigate different methods for adversarial imitation learning, in particular IfO methods;
- implement at least 2 different models;
- propose improvements;
- perform evaluations on low dimensional control tasks (cartpole, mountain car) and in the MuJoCo simulated environment [3].

[1] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In NIPS, pages 4565–4573, 2016.

[2] Peter Henderson, Wei-Di Chang, Pierre-Luc Bacon, David Meger, Joelle Pineau, and Doina Precup. Optiongan: Learning joint reward-policy options using generative adversarial inverse reinforcement learning. In Association for the Advancement of Artificial Intelligence, 2018.

[3] http://www.mujoco.org/index.html

Title: Human-robot interaction

Coordinators: *Prof. dr. ing. Irina Mocanu (irina.mocanu@upb.ro)*
Dr. ing. Alex Awada (alex.awada@upb.ro)

Description:

The aim of the project is to create a system capable of enabling multimodal human-robot interactions such as gesture and voice interactions. A gesture recognition system will be developed based on the analysis of RGB images (or RGB-D - RGB images and depth). Gesture recognition will be ported on the TurtleBot robot, which will perform a certain command for each recognised gesture (for example, move to a certain point in case of waving the hand - simple, predefined actions will be used for robot commands). To enable voice interactions, a service should be built making use of existing web APIs that facilitate the processes of text-to-speech, speech-to-text and natural language understanding, for the Romanian and/or English language(s). The integration of additional ways of interactions between the user and the TurtleBot robot is allowed.

Bibliography:

1. https://news.developer.nvidia.com/hello-world-robot-responds-to-human-gestures/
2. https://arxiv.org/pdf/2007.09945v2.pdf
3. https://www.sciencedirect.com/science/article/pii/S2212827120314359
4. https://www.frontiersin.org/articles/10.3389/frobt.2021.612750/full

Title: Object manipulation using TIAGo robot

Coordinator: *Prof. dr. ing. Irina Mocanu (irina.mocanu@upb.ro)*

Description:

The aim of the project is to create a system for object manipulation performed by the TIAGo robot. Using the RGB-D images acquired by the robot, the features of the object will be detected in order to grasp it. In cluttered scenes, when the desired object is closely surrounded by other objects, a simple push could spread them apart and make the desired object available for grasping. First, the system will be tested in a simulation environment using Gazebo simulator. Then, the system will be deployed on the TIAGo robot.

Bibliography:

1. Gazebo simulator: http://gazebosim.org/
2. What are GQ-CNNs? — GQCNN 1.1.0 documentation. url: https://berkeleyautomation.github.io/gqcnn/info/info.html.
3. Bing Tang and others. "Learning Collaborative Pushing and Grasping Policies in Dense Clutter". in: 2021.

Title: Dialogue Management Service supporting the Romanian language in assistive robotics scenarios

Coordinators: *Assist. Prof. Alexandru Sorici (alexandru.sorici@upb.ro)*
Dr. ing. Alex Awada (alex.awada@upb.ro)

Description:

A dialogue management system, in the context of assistive robotics, is an application that models and guides a conversation carried out between a human user and a robot, by means of multiple modalities (e.g., written text, voice interaction), taking into account external cues (e.g., environment state, user fatigue or emotional state).

The purpose of this project is to develop a dialogue management service that works across different platforms and supports the Romanian language.
The service will be built in a modular, micro-service oriented architectural style, making use of existing web APIs that facilitate text-to-speech and speech-to-text processing for the Romanian language. The dialogue management service is expected to enable scriptable interaction scenarios, whereby the answers given by the robot are informed by both conversational context, as well as emotional state of the user (i.e. the response in the dialogue depends on what the user has said previously and on how he is feeling).
At the same time, the start of a dialogue can be triggered by contextual cues (e.g. the light is turned on in a room, a motion sensor is triggered), apart from direct voice interaction.

Title: Enhancing the functionality of the AMIRO User Interface

Coordinators: *Assist. Prof. Alexandru Sorici (alexandru.sorici@upb.ro)*
Dr. ing. Alex Awada (alex.awada@upb.ro)

Description:
AMIRO (AMbIent RObotics) is a ROS-based system enabling monitoring (e.g. detecting when the door of the lab is opened or closed) and actuation (e.g. raise or lower the blinds in the lab, turn the smart lights on/off or change their color) of an indoor lab environment, as well as access to external context information (e.g. health parameters of lab personnel) by a socially assistive robot.
The service also defines basic robot behaviors (e.g. navigating to a given position, identifying a person, searching for an object, speaking or listening for a voice command) that can be composed in a hierarchical manner to create more elaborate human-robot interaction scenarios (e.g. guiding a user to a location, finding a person in the lab to inform them of a notification).
The purpose of this project is to enhance the current UI (user interface) for the AMIRO system by creating a back-end tool (robot behavior management) that allows the creation of different robot behaviors by dragging, dropping and linking different elements (such as: predefined behaviors, predefined or new animations and dialogs, custom Python codes, etc.) into a scene. The tool output must support the export of constructed behavior compositions into tasks executable through the AMIRO system.

Title: Learning Multi-Robot Swarm Decentralized Behaviours using Deep Reinforcement Learning

Coordinator: *SL.dr.ing. Dan Novischi(dan_marius.novischi@upb.ro)*

Description:
In swarm systems, many identical agents interact with each other to achieve a common goal. Typically, each agent in a swarm has limited capabilities in terms of sensing and manipulation so that the considered tasks need to be solved collectively by multiple agents. A promising application where intelligent swarm systems take a prominent role is swarm robotics. Robot swarms are formed by a large number of cheap and easy to

manufacture robots that can be useful in a variety of situations and tasks, such as search and rescue missions or exploration scenarios. A swarm of robots is inherently redundant towards loss of individual robots since usually none of the robots plays a specific role in the execution of the task. Because of this property, swarm-based missions are often favorable over single-robot missions; or, let alone, human missions in hazardous environments.

Recently, deep reinforcement learning (RL) strategies have become popular to solve multi-agent coordination problems. In RL, tasks are specified indirectly through a cost function, which is typically easier than defining a model of the task directly or finding a heuristic for the controller. Having defined a cost function, the RL algorithm aims to find a policy that minimizes the expected cost. Applying RL within the swarm setting, however, is challenging due to the large number of agents that need to be considered. Compared to single-agent learning, where the agent is confronted only with observations about its own state, each agent in a swarm can make observations of several other agents populating the environment and thus needs to process an entire set of information that is potentially varying in size. Accordingly, two main challenges can be identified in the swarm setting:

- High state and observation dimensionality, caused by large system sizes.
- Changing size of the available information set, either due to addition or because the number of observed neighbors changes over time.

The project goal is to investigate these problems in the context of generating swarm behaviors - such as (but not limited to) aggregation /dispersion (i.e. dynamic graph building), target localization and pursuit evasion - using deep reinforcement learning techniques such as DQN's, DDPG and Neural Network based Trust Region Policy Optimization.

[1] Hüttenrauch, Maximilian, Adrian Šošić, and Gerhard Neumann. "Guided deep reinforcement learning for swarm systems." arXiv preprint arXiv:1709.06011 (2017).
[2] Hüttenrauch, Maximilian, Adrian Šošić, and Gerhard Neumann. "Local communication protocols for learning complex swarm behaviors with deep reinforcement learning." International Conference on Swarm Intelligence. Springer, Cham, 2018.
[3] Hüttenrauch, Maximilian, Sosic Adrian, and Gerhard Neumann. "Deep reinforcement learning for swarm systems." Journal of Machine Learning Research 20.54 (2019): 1-31.
[4] https://github.com/ALRhub/deep_rl_for_swarms
[5] https://github.com/hex-plex/KiloBot-MultiAgent-RL

Title: Learning Mobile Robots Autonomous Navigation Behaviors using Deep Reinforcement Learning

Coordinator: SL.dr.ing. Dan Novischi(dan_marius.novischi@upb.ro)

Description:

One of the major challenges for robot control and navigation is safe and robust collision avoidance, so that the robot can navigate from its starting position to its goal position. This holds for user programmed robots as well as for policies learned by the robot itself to control and navigate. In the area of autonomous robots, learning strategies, especially reinforcement and deep learning, gained much interest in recent years. Unfortunately, most of the robot learning approaches use only simulations or artificial environments and are not transferred to real robots in real (not controlled) environments. The challenges to achieve that, especially for reinforcement deep learning based robots, are:

- **Simulation:** Since reinforcement learning tests millions of input/output combinations, we need a fast (parallel) simulation environment. Another requirement for the simulator is to model the real world accurately and to support a wide range of environments from simple to complex (i.e. MRS Open Simulator, Roblearn, RotorS).
- **Robot deployment:** To deploy and to continue the learning/testing on a real robot we need (besides the robot) a software framework (i.e., ROS) and a strategy to transfer the learning results from the simulation to the robot and its sensors. The different sensors should be fused.
- **Network:** The deep reinforcement learning system should observe the speed and memory constraints introduced by the robot platform. Therefore, a Network architecture, size, number of parameters, learning strategy, as well as input and output parameters, have to be found.

The goal of this research is to investigate the above mentioned challenges using DRL techniques that are well suited for continuous tasks, such as Deep Deterministic Policy Gradient (DDPG), in order to develop autonomous mobile robot behaviors in the absence of a map or a motion planner.

[1] Surmann, Hartmut, et al. "Deep Reinforcement learning for real autonomous mobile robot navigation in indoor environments." arXiv preprint arXiv:2005.13857 (2020).
[2] Shabbir, Jahanzaib, and Tarique Anwer. "A survey of deep learning techniques for mobile robot applications." arXiv preprint arXiv:1803.07608 (2018).
[3] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. PMLR, 2016.
[4] https://github.com/RoblabWh/RobLearn
[5] https://github.com/unstablecursor/robotino_rl