AMR 2017/2018: Final Projects

A final project consists of:
• studying some papers, notes or documentation on a specific subject
• performing simulations or numerical tests on a software platform
• writing a report
• giving a presentation (with slides)

As a rule, each project must be carried out by a group of **3 students**. Projects are assigned to groups on a FIFO basis. To apply, send me an e-mail message with the composition of your group (one message per group, all members in cc:) and a list of **3** projects in order of preference. 1 or 2-student groups can also apply, but I will merge them into larger groups. The deadline for applying is **May 28**. Late applications will not be accepted.

Two projects are **shared** with **Robotics 2**; these are for students that have qualified for both AMR and Robotics 2, and are simultaneously valid as final projects for both courses.

Once your group has been assigned a project, we will set up a meeting to discuss the project in detail. There will be three deadlines for submitting your projects: **June 30**, **September 30**, **December 31 (2018)**. To submit your project, send me an e-mail with the report. Once a deadline is passed, I will fix a common date for presenting all the projects completed during the associated time window.

This is the list of the available projects, followed by a short description of each project.

1. **Robust gait generation for humanoids using MPC** (shared with Rob 2)
2. **CoM/ZMP estimation on the NAO humanoid** (shared with Rob 2)
3. **Swing-up of the cart-pendulum via learning-based MPC**
4. **Lazy collision checking in sampling-based motion planning**
5. **Anytime asymptotically-optimal motion planning**
6. **Emergency stop procedures for the HRP-4 humanoid robot**
7. **A suite of feedback controllers for unicycle robots**

---

1. Robust gait generation for humanoids using MPC (shared with Rob 2)

Synopsis
The objective of this project is to develop a method for humanoid gait generation that is robust to perturbations on the robot parameters. In particular, starting from an existing gait generator based on Model Predictive Control, a *tube-based* strategy should be used to guarantee that the constraints (on balance and kinematic feasibility) are still guaranteed for the perturbed model. Additionally, a structurally stable gait generator should be considered, in order to verify whether it is possible to robustify the boundedness constraint at its core. The simulation platform will be the NAO humanoid. The software platform may be MATLAB, DART or V-REP.

Reading material
Papers on robust MPC control of linear systems
Scianca et al, "Intrinsically stable MPC for humanoid gait generation", Humanoids 2016

2. CoM/ZMP estimation on the NAO humanoid (shared with Rob 2)

Synopsis

This project is aimed at developing a Kalman Filter for estimating the position of the Center of Mass and of the Zero Moment Point on the NAO humanoid robot. The available sensors for estimation are the joint encoders, the pressure sensors beneath the feet, and the robot inertial measurement unit. The filter should be first tested in simulation and then validated via experiments on the real robot. In addition, the CoM/ZMP estimates should be used for closing the feedback loop in the existing MPC gait generator. The project requires some C++ coding.

Reading material

NAO documentation
Oriolo et al, "Humanoid odometric localization integrating kinematic, inertial and visual information", Autonomous Robots, 2016

3. Swing-up of the cart-pendulum via learning-based MPC

Synopsis

The objective of this project is to show that is possible to improve the performance of Model Predictive Control (MPC) schemes with policies learned through Reinforcement Learning or policy search approaches. To this end, the projects envisages the use of a simple learning scheme (like a tabular Q-learning or a REINFORCE policy search algorithm) in order to build an optimal control policy for swinging-up a cart-pole system and stabilizing it at an inverted equilibrium. A Linear Time-Varying MPC scheme should then be designed to track the trajectory associated to the learned optimal policy. Comparative results should be provided to support the claim that this learning-based MPC approach performs better than classic MPC. The software platform is MATLAB.

Reading material

Sutton and Barto, "Reinforcement Learning: An Introduction", 1998
Borrelli at al, "Predictive Control for Linear and Hybrid Systems", 2017

4. Lazy collision checking in sampling-based motion planning

Synopsis

Collision checking is recognized as the most expensive process within sampling-based motion planners. Motivated by this, some methods known as *lazy* algorithms postpone collision checks until they are strictly required. Such approach has been widely used with planners that builds graph-like structures, such as PRM, where the high connectivity can provide a sufficiently large set of potential plans. The objective of this project is to test the lazy approach within a planner that builds a tree in the configuration space, such as the classical RRT. Two variants of the RRT algorithm should be implemented, respectively with lazy collision checking at edge level and at both node and edge levels. A critical comparison based on simulations should be performed. The software platform should be V-REP. Intermediate-level C++ programming skills are required.

Reading material

Bohlin and Kavraki, "Path Planning using Lazy PRM," ICRA 2000
Hauser, "Lazy Collision Checking in Asymptotically-Optimal Motion Planning," ICRA 2015

V-REP software documentation

---

5. Anytime asymptotically-optimal motion planning

Synopsis
RRT* is a sampling-based algorithm with an asymptotic optimality property; this means that, contrarily to the basic RRT, it accounts for the quality of the generated solution trajectories. On the other hand, it introduces some computational overhead that is problematic when planning under time limitations. The anytime version of RRT* addresses this issue by quickly computing an initial sub-optimal solution and repeatedly improving it during the simultaneous execution. The objective of this project is to implement such approach for the case of a mobile robot that is assigned a navigation task. A performance assessment of the method should be made, testing the algorithm in scenarios of increasing complexity. The software platform is V-REP. Intermediate-level C++ programming skills are required.

Reading material
Karaman et al, "Anytime Motion Planning using the RRT*," ICRA 2011
V-REP software documentation

---

6. Emergency stop procedures for the HRP-4 humanoid robot

Synopsis
Consider a scenario where a humanoid robot is walking and an imminent danger is detected, so that the robot must be brought to a complete stop without falling. The aim of this project is the implementation of an emergency stop procedure for the HRP-4 humanoid. Two approaches should be compared: the first uses the concept of capture point to determine where the foot has to be placed so that the robot can stop without falling, and the second is based on intrinsically stable MPC. The software platform is MATLAB and/or V-REP.

Reading material
Papers on capturability
Scianca et al, "Intrinsically stable MPC for humanoid gait generation", Humanoids 2016

---

7. A suite of feedback controllers for unicycle robots

Synopsis
The goal of this project is to implement and test in V-REP a suite of feedback controllers for a unicycle robot. In particular, the considered robot is a differential-drive vehicle with odometric localization and equipped with a GPS. Both trajectory tracking and posture regulation controllers will be considered. The selected control schemes will be those studied in the course, with the addition of two more sophisticated trajectory control schemes. Comparison of the various controllers over a campaign of simulations is expected, both under odometric and GPS localization.

Reading material
Course slides on control of WMRs
Additional notes
V-REP software documentation