AMR 2011/2012: Final Projects

0. **General Information**

A final project includes:

- studying some literature (typically, 1-2 papers) on a specific subject
- performing some simulations or numerical tests on an appropriate software platform (Matlab, Webots, KiteLab,...); sometimes, experiments are involved
- writing a report
- making 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. Send me an e-mail message (not through the Google Group) specifying **at least 3** projects your group is interested in, with an order of priority, and the composition of the group. One or two-persons groups can also apply, but I reserve the right to merge them to a larger group. Once your group has been assigned a project, we will set up a meeting to discuss the project in detail.

The deadline for project application is **June 15**. Late applications will not be accepted. There will be three deadlines for turning in your projects: **June 30**, **September 30**, **December 31**. To turn in 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. I strongly recommend attendance to all presentations even if you are not directly involved.

A few larger projects are **shared** with Robotics 2; this means that they are simultaneously valid as final projects for both courses.

This is the list of the available projects:

1. Mixed initiative control of NAO (shared with Robotics 2)
2. Task transition in kinematic control of redundant robots (shared with Robotics 2)
3. Trajectory tracking for robots with low-level servo loops (shared with Robotics 2)
4. Obstacle clustering with Kinect (shared with Robotics 2)
5. Visual servoing with AR-ToolKit (shared with Robotics 2)
6. 3D modeling and control of Khepera III in Gazebo-ROS simulator
7. Real time feature extraction from laser scans
8. Camera velocity estimation by optical flow

A short discussion of each project follows.

1. Mixed initiative control of NAO (shared project with Robotics 2)

The objective of this project is to implement a joystick module, by which it will be possible to control remotely the NAO humanoid by acting on an external joystick (to be set up and interfaced with the robot) by using the information which comes from the robot camera. The two-dimensional command coming from the joystick must be appropriately mapped to a three-dimensional displacement. An automatic obstacle avoidance algorithm will be integrated. Implementation in simulation and then on the actual NAO robot.

Literature

NAO documentation.
De Luca and Oriolo, "Local incremental planning for nonholonomic mobile robots," Proc. 1994 IEEE Int. Conf. on Robotics and Automation (ICRA'94), pp. 104-110, San Diego, CA, USA, 1994.

Notes

A ROS node which interface a joystick with NAO already exists (but has to be checked and eventually fixed). Moreover, a node by which it is possible to open NAO camera remotely has been already implemented. A joystick will be provide by LabRob. Programming skills (C++), knowledge of ROS platform (not mandatory).

2. Task transition in kinematic control of redundant robots (shared project with Robotics 2)

Synopsis

Kinematic control of redundant robotic systems is often obtained by prioritizing tasks that can be expressed as equality (e.g., trajectory execution) or inequality constraints (e.g., joint limits). Complex operations of a robot can be decomposed into a sequence of phases where a particular set of tasks with given priorities is specified. From one phase to the next one, there will be a transition of tasks, by inserting a task or removing it or changing its priority (including swapping of priorities between tasks). This project is aimed at studying and implementing a method that achieves smooth and continuous transitions among tasks effectively at the velocity or acceleration command level. A possible solution is to generate intermediate joint reference values for the two controllers of contiguous phases, but other options may be explored.

Literature

J. Lee, N. Mansard, J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," to be published in the IEEE Transactions on Robotics, 2012.

Notes

Both simulation in Matlab environment and experimental work on the KUKA LWR4+ can be foreseen.

3. Trajectory tracking for robots with low-level servo loops (shared project with Robotics 2)

Synopsis

Accurate control of motion is obtained by applying robot commands at the torque level, as computed using the robot dynamic model, its current state, and the desired reference trajectory (e.g., feedback linearization control). However, torque commands (or current commands to the DC motors) are not directly accessible in industrial robots with closed control architectures. Rather, the user can specify a kinematic command in the form of a reference position or reference velocity in the joint space. These commands are then processed by the low-level joint servo loops that will in turn impose a motion torque. This project aims at showing how to specify a reference position or reference velocity in the outer loop so as to achieve internally a desired torque command. Different low-level loops can be assumed (e.g., P, PD, PID, or generic pole-zero transfer function C(s) modeling position, velocity and current loops), possibly including a feedforward action.

Literature

- W. Verdonck and J. Swevers, "Improving the dynamic accuracy of industrial robots by trajectory pre-compensation," Proc. 2002 IEEE Int. Conf. on Robotics and Automation (ICRA'02), pp. 3423-3428, Washington, DC, USA, 2002.
- O. Khatib, P. Thaulad, T. Yoshikawa, and J. Park, "Torque-position transformer for task control of position controlled robots," Proc. 2008 IEEE Int. Conf. on Robotics and Automation (ICRA'08), pp. 1729-1734, Pasadena, CA, USA, 2008.
- T. Yoshikawa and O. Khatib, "Compliant humanoid robot control by the torque transformer," Proc. 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'09), pp. 3011-3018, St. Louis, MO, USA, 2009.
- Internal notes

Notes

Both simulation in Matlab environment and experimental work on the KUKA KR5 can be foreseen (by different groups).

---

4. Obstacle clustering with Kinect (shared project with Robotics 2)

Synopsis

The target of the project is to collect in clusters all obstacles in a scene monitored by a depth sensor, i.e., the Microsoft Kinect. The cluster associated to a robot (which pose is known) must be identified and not considered as an obstacle. The velocity of each clustered object will be then estimated considering the variation of the position of the center of the cluster. The resulting data will be used in an obstacle avoidance algorithm for the KUKA LWR4+ manipulator. The project will be realized in Gazebo, using ROS and the Point Cloud Library, following this roadmap:

1. Set up the Kinect Sensor in Gazebo/ROS
2. Obtain the point cloud
3. Clustering of the object using PCL
4. Filter small objects and noise
5. Labeling of the clusters
6. Velocity estimation of the obstacles

Literature
To be determined.

Notes
Good comprehension of C/C++ language and programming concepts required. A basic knowledge of ROS is a plus.

5. Visual servoing with AR-ToolKit (shared project with Robotics 2)

Synopsis
AR-ToolKit is a software which allows to estimate the pose of a structured marker from an image obtained with a camera. The goal of the project is to use AR-Toolkit in Webots to estimate the image position of the marked object and its depth. The pose estimator will be used on our KUKA Sixx manipulator in conjunction with a previously realized visual servoing algorithm. The project will be developed according to the following roadmap:
1. Set up AR-ToolKit to detect and estimate the object pose
2. Integrate AR-ToolKit in our KUKA Simulator on Webots
3. Test the visual servoing algorithm
4. Compare the results with the previous work

Literature
To be determined.

Notes
Good comprehension of C/C++ language and programming concepts required. A basic knowledge of Webots is an plus.

6. 3D modeling and control of Khepera III in Gazebo-ROS simulator

Synopsis
The first objective of this project is to build an accurate 3D model for the Khepera III differential drive robot for collision checking and dynamic simulation purposes. The second is to create a Gazebo controller that generates the appropriate velocity inputs for the model. Additional sensors (e.g. laser range finder, camera, Kinect, etc, already built in Gazebo-ROS) can be incorporated. This is the project roadmap:
1. Analysis of the Erratic robot already implemented in Gazebo-ROS.
2. Development of an accurate CAD model of Khepera III.
3. Creation of the relative controller by subscribing an input node in ROS and publishing odometry information as well as other data on an appropriate ROS nodes.
4. Implementation of a position controller.

Literature
To be determined.

Notes
Some previous knowledge about CAD modeling is helpful. Knowledge of C/C++ is required for the implementation part. Previous knowledge of Gazebo or ROS is a plus.

7. Real time feature extraction from laser scans

Synopsis
The project aims at the implementation of some algorithms for the extraction of general purpose features (lines, edges, corners, etc) from laser scan data. The group must implement and test at least two different methods for feature extraction from a set of data collected from Hokuyo laser sensor. A nice addition would be the recognition of circular robots (like the Khepera III like). The project will be developed according to the following roadmap:

1. Literature review of feature extraction to identify out some suitable methods.
2. Implementation of the chosen methods.
3. Test on simulated data.
4. Experiments on the Khepera III robots.

Literature
To be determined.

Notes
Any previous knowledge on feature extraction or vision and sensor data collection issues is useful. Knowledge of C/C++ is required for the implementation part.

---

8. Camera velocity estimation by optical flow

The objective of this project is to perform a literature review of some techniques for estimating the velocity of a moving camera (and hence of the robot that carries the camera) based on the optical flow concept. In particular, at least two such techniques should be implemented and compared w.r.t. ground truth data (provided by a Gazebo-ROS simulation of a quadrotor with a down-looking camera). The project will be developed according to this roadmap:

1. ROS review to identify already implemented methods.
2. Literature review to identify out some suitable algorithms.
3. Implement two (or more) such algorithm and compare with ROS algorithms.
4. Test the implementation in Gazebo-ROS (given an implementation of a quadrotor with a down-looking camera).

Literature
To be determined.

Notes
Previous knowledge about vision and image analysis is helpful. Knowledge of C/C++ is required for the implementation part.