Topic: Implementing the Subsumption robot control architecture
Mentor: Saugata Biswas

Description:

- In this architecture the complicated intelligent robot behavior is divided into several simple behaviors in the form of modules. These modules(behaviors) are organized in the form of layers [1].

- Each layer consists of a particular goal. Each layer will subsume the goal of its underlying layers [1].

- For example, a wandering robot might consist of two layers. An obstacle avoidance layer as the lower layer and wandering behavior as the upper layer.

- In this case, the decision of wandering around (move forward/backward/left/right) takes into account the decision of the lowest layer(obstacle avoidance).

What students will lean:

- Basics of robot architecture(theory).
- Design patterns during software implementation.

References:

[1] https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Subsumption_architecture.html
[2] http://dspace.mit.edu/bitstream/handle/1721.1/6432/AIM-864.pdf
[3] http://en.wikipedia.org/wiki/Subsumption_architecture

Topic: Implementing the RRT motion planner for robot manipulator
Mentor: Niranjan Deshpande, Praveen Ramanujam

Description:

- In this project, participants will structure a planner based manipulation for a robot arm.
- Many planning based algorithms exist for manipulation, refer the purpose of OMPL[4] or OpenRAVE[7].
- In this hackathon, we concentrate only on search based algorithms and narrow it to RRT[1].
- Please go through the wiki link [1] to get the basic idea of RRT.
- Preliminary, participants should be able to design a scalable software through which any new planners can integrated easily.
- An efficient way of setting the parameters should be designed.
- It should also scale for different robots.
- They can use [5] as a reference and modify it for youbot. If they are not confident to build their own RRT, they can use RRT from other libraries and control the robot. One such implementation where OpenRAVE and youbot are communicating can be seen in [6], Participants can also use ROS-OMPL as well.
- Goal is to produce a scalable and parameterized software, well documented using doxyen and working code in youbot.

- In order to perform the task, the object to be grasped can be assumed for its position. The final joint angles can be assumed.
- If you use ROS-OMPL, one can use Moveit to do so, however, it is recommended the participants build their own planner package. It is completely ok, if it does not function perfectly.

What students will learn:

- Basics of RRT planner(theory).
- Design patterns during software implementation.

References:

[1] http://en.wikipedia.org/wiki/Rapidly_exploring_random_tree
[2] http://www.cs.cmu.edu/~motionplanning/lecture/lec20.pdf
[3] http://coecsl.ece.illinois.edu/ge423/spring13/RickRekoskeAvoid/rrt.html
[4] http://ompl.kavrakilab.org/
[5] https://github.com/RoboJackets/rrt
[6] https://www.youtube.com/watch?v=4g1GAfRt1ig
[7] http://openrave.org/docs/latest_stable/

Topic: Robust Position control of the Arm using velocity interface.
Mentor: Shehzad Ahmed
Description:

1. The focus of the project is to have robust position control of the robotic arm.
2. The idea is to command the position of the joints and develop a controller which then control the joints motion using velocity interface.
3. The component should not be robot dependent. It must be developed like that it is re-usable and integrable.

Topic: Robust Pose control of the base using velocity interface.
Mentor: Shehzad Ahmed
Description:

1. The focus of the project is to have robust pose control of the robotic base.
2. The idea is to command the pose of the base and develop a controller which then control the motion using velocity interface.
3. The component should not be robot dependent. It must be developed like that it is re-usable and integrable.

Topic: Vision based Grasp Verification for Robocup@work scenerios.
Mentor: Ashok Meenakshi Sundaram