Real-Time Systems

Lab 4: Adding Guidance and Planning

Due: February 23, 2017 (Before Class)

Introduction

Now that you've written and analyzed a controller for Ringo, it's time that you support that with a planning task and a guidance task to be able to have better control and accomplish a mission. Feel free to use online resources to come up with ideas for a guidance law and planner, though you really can do it on your own. This is another chance for you to be creative and "design" something.

Write any support tasks that might be required. For example, you likely need a sensing task, or perhaps a task resetting Ringo to another state once you reach a goal in the mission. At the end of this lab you should (likely) have **at least** 4 tasks sensing, planning, guidance, and control. These can be subdivided into more tasks if needed.

Finally, provide an estimate for the WCET for each task in the system. This will be needed for the next lab.

Intro to Guidance and Planning

"Guidance," "planning," and "control" are really just abstractions for algorithmically and architecturally breaking up the task of moving a robot from point to point in an environment. As such, it is difficult to provide a precise definition of each of them. "Control" usually means low-level control where an input to an actuator is computed to achieve some step response. As an example, in Ringo, as we did in Lab 3, this means following a line or executing a single rotation. A controller, on its own, is typically insufficient to accomplish a mission.

"Planning" usually means devising a plan containing high level motions and tasks that the robot must do to accomplish a mission. As an example, for Ringo, this could mean go from point to point through an environment that has obstacles. In that scenario, a valid plan would break the mission down into waypoints that avoid obstacles and perhaps meets some performance objectives (like not taking an entire day to get there). To prepare your Ringo to accomplish the final exam, I suggest you make your planning task accept a mission in the form of navigating a particular shape. You might accept polygons. Once a polygon is specified, the planner will devise a plan for navigating that shape. This might include waypoints, or distances and rotations, or something else.

"Guidance" often refers to a middle layer that bridges higher level plans from a planning algorithm with the small motions the controller is capable of achieving. If you wrote a controller capable of following a line or heading, and a controller capable of rotations, then the guidance law should take the plan of waypoints, rotations, distances, etc. and break this plan down sequential steps that can be achieved by your controllers.

Please note that the division of a mission into plans, guidance laws, and controllers is arbitrary and depends on mission requirements and your design. The final exam will ask you to navigate one or more polygon shapes in some reasonable amount of time. You should design a planner and guidance law, that, together with your controllers is capable of accomplishing that type of "mission." Once you have a guidance law and planner, be sure to interface it with the controller task(s) written before to demonstrate your Ringo can achieve mission objectives.

NOTE: There are MANY resources for learning about robot planning online. Feel free to use them. But be warned that much of it is overly complex for our purposes. We're focused on RTS principles, not so much on guidance, planning, and control. For additional resources, and learning, have a look at [2, 3, 1, 4, 5] for a few of my favorites (you should be able to look these up online and/or get the book from the library).

Worst Case Execution Time (WCET)

You can use a static analysis tool or a measurement based technique to estimate WCET of each task. A reasonable list of WCET tools can be found at `https://www.rapitasystems.com/WCET-Tools`. Alternatively, you can do a statistical analysis by running your task enough times and estimating WCET. Here "enough times" is ill-defined as discussed in class. Whichever method you use, remember to add a ~20% cushion to the time for a margin of "safety."

What to submit

1. **(35 points)** Zip the entire Arduino project so that we can just unzip and execute the source code.
   - Please include a diff of the source code between this lab and previous one. Please use a "diff" tool of some kind that highlights the changes from previous files (e.g. `https://www.diffchecker.com/`, meld, github, etc.).
2. **(30 points)** Documentation of the task(s) you have implemented for this lab. (0.5 - 1 page long)
   - Provide an algorithmic overview of the guidance law
   - Provide an algorithmic overview of the planner
   - Provide a WCET estimate for each task you have
   - Describe each implemented task briefly and how you decided how to break up the behaviors into different tasks (or why you chose to use a single task). Classify any tasks into periodic, aperiodic, or sporadic, and provide justifications on why a task is of a certain type and what were the thoughts while deciding the frequency of the tasks.
3. **(35 points)** Record a short video of the robot going around a more challenging shape (a non-rectangular polygon of some kind) in a limited amount of time. This will be good preparation for the final exam.
   - Be sure to describe or narrate the test that Ringo is doing that demonstrates that you have accomplished the objectives of the lab.
   - Please upload the video to YouTube, Vimeo, or something similar and then provide me with a link. Just put the link somewhere in the writeup you did in #2 above.
4. Upload all of this into "handin" at `https://cse-apps.unl.edu/handin`

References

[1] On three-layer architectures. In *Artificial intelligence and mobile robots*, pages 195–210. AAAI press Cambridge, MA, 1998.

[2] Jim Blythe and W. Scott Reilly. Integrating reactive and deliberative planning for agents. Technical report, DTIC Document, 1993.

[3] R Peter Bonasso, R James Firby, Erann Gat, David Kortenkamp, David P Miller, and Mark G Slack. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):237âĂŞ256, 1997.

[4] C Goerzen, Zhaodan Kong, and Bernard Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4):65âĂŞ100, 2010.

[5] S J Russell and P Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.