Final Project

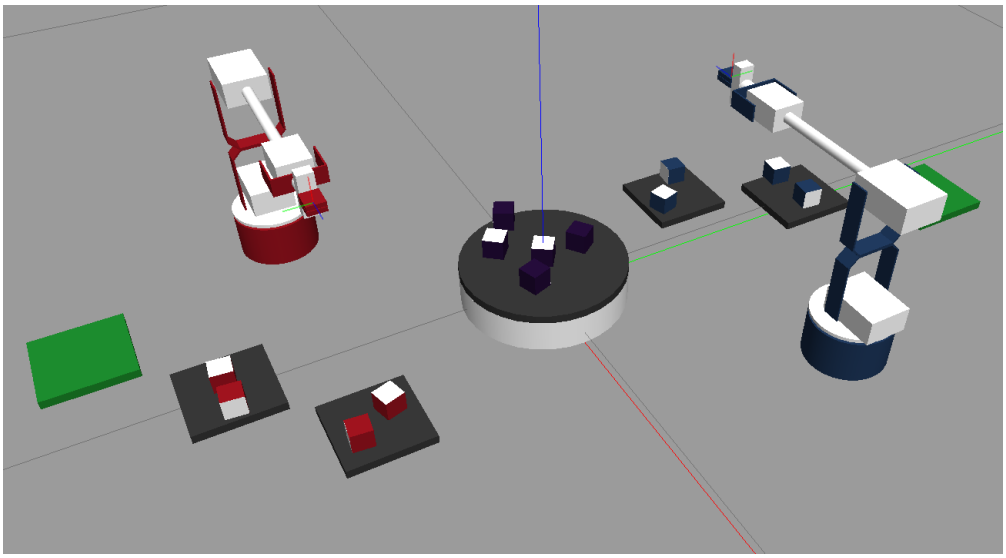
MEAM 520, University of Pennsylvania

November 13, 2020

Teams will use the concepts learned during the semester to control their simulated Lynx robot in a head-to-head competition with their opponents' robot. The robots will manipulate objects in the simulated environment to score points, culminating in a class-wide tournament!

Instructions: Just as in labs, this final project is an opportunity for you to explore the concepts we learned in class in a more complicated environment. Expand on previous labs, pull techniques from the literature, or try some experimentation of your own. You should document your approach through a report similar to the reports you have written throughout the semester.

The final project is worth 70 pts. Bonus points will be awarded to teams who perform particularly well during the tournament: 5 pts to 1st place, 3 pts to 2nd place, 1 pt to 3rd place.



1 Competition Rules

1.1 Ground Rules

1. **Students are required to work in teams of four.** If you would like, you may also be randomly matched with other students by the teaching staff **Regardless, you must fill out the form on Piazza to either register your team or ask to be matched by November 20 @ 12 noon**. Any student who does not register their team will be automatically assigned to a team. A few students will likely end up in teams of 3 but this will be sorted out by the teaching staff.

2. **Teams will submit their code through Gradescope before the competition.** During the competition, TA's will run the game on a physical Ubuntu machine (not the provided Virtual Machine) while streaming the simulation live on Zoom.

3. Each match will consist of two teams (designated Red Team and Blue Team) competing head to head in a shared simulation environment, shown above.

   (a) The base of each robot is located 200mm from both the `x` and `y` axes of the world, on the world's `x-y` plane. The base frame of the Red Team robot's is parallel to the world axes, while the base frame of the Blue Team's robot is rotated by 180° about the world `z` axis.

   (b) The turntable holding the dynamic blocks is located with the center of its base at the origin, with radius 100mm and height 50mm.

   (c) The remaining environment geometry (namely that of the static object and goal platforms) is defined in the provided map `final.txt`

   The static environment will not change, but the placement of scoreable objects will be randomized before the start of each match, and your opponent will also operate in the shared environment. *Note: randomization of the blocks has not yet been implemented in the simulation but will be released as soon as it's ready.*

4. **Matches will last for 60 seconds of simulation time.** During the tournament, your code will be started automatically, and the simulation will halt after the play clock has run out.

5. **Teams must only interact with the simulation through a single Arm Controller object for their robot.** Teams are forbidden from directly sending or receiving data over any ROS topics or services, in order to create a level playing field for all. **Teams are expressly forbidden from sending commands to their opponent's robot.**

1.2 Scoring Points

1. Scoreable objects are `20mm`×`20mm`×`20mm` blocks, which come in two varieties:

   (a) **Dynamic Blocks (Purple):** These objects are randomly dispersed on a rotating turntable within the workspace of both robots. Dynamic blocks have a `Value` of 3.

   (b) **Static Blocks (Red and Blue):** These objects are randomly dispersed on static platforms. A symmetric set of such platforms exists within the workspace of each robot, outside its opponent's workspace. Static blocks have a `Value` of 1.

   All scoreable objects have one side that is white, corresponding to the `+z` axis of that object's coordinate frame.

2. During the match, teams will manipulate objects to move them onto their team's goal platform, shown in green. **All points will be scored after the match concludes, so that only their final state determines the awarded points.** The goal platform of each robot will lie outside the reachable workspace of their opponent.

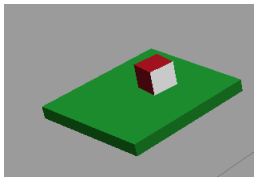
3. Each individual scoreable object supported by a team's goal platform will receive points according to the formula `Points = Value × (1 + Altitude / 10 + SideBonus)`, where:

   (a) `Altitude` is the distance from the center of the object to the surface of the goal platform in millimeters

   (b) `SideBonus` is 1 if the white side of the object is pointing upward, 0 otherwise.

   The team's final score is the sum of the point value of each scoreable object. Below are some example final configurations of blocks and their corresponding point values.

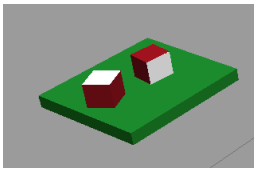
Example 1
static block, no bonus



individual blocks:
`1 × (1 + 10/10 + 0) = 2`

Total Points: `2`

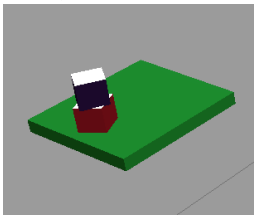
Example 2
two static blocks side by side, one bonus



individual blocks:
`1 × (1 + 10/10 + 0) = 2`
`1 × (1 + 10/10 + 1) = 3`

Total Points: `5`

Example 3
dynamic block on top of static block, both bonus



individual blocks:
`1 × (1 + 10/10 + 1) = 3`
`3 × (1 + 30/10 + 1) = 15`

Total Points: `18`

1.3 Tournament

1. The tournament will consist of two phases:

   (a) **Round Robin, December 3:** *(10 matches per division, each team plays 4 times)*

      i. The class will be divided into 4 divisions, each containing 5 teams.

      ii. Each team will play against all other teams in their division

      iii. The two teams with the best record will advance, determined successively by:

      - Number of wins
      - Total points scored by the team
      - Total points scored by the team's opponents in each round

      iv. Matches will take place during lecture, with the exception that one division will compete during an evening session the same day from 7-8:30pm (Philadelphia time) to accommodate time zone differences.

   (b) **Single Elimination, December 8:** *(7 matches total)*:

      i. The remaining 8 teams will go head to head, with only the winner of each match advancing to the next round

      ii. All matches will take place during lecture. All students should attend (timezone permitting) to support their colleagues!

      iii. The last Lynx standing is the champion!

2. In the event of a tie, the winner of a match will be determined successively by

   - `Altitude` of the highest scored block
   - Number of Dynamic blocks scored
   - Sum of `SideBonus` over all scored blocks
   - Coin flip, rematch, or TA judgment as a last resort

2 Code and Logistics

Starter code will be released by November 20. Keep an eye on Piazza for the announcement.

2.1 Competition Environment

We've created a new simulation environment with two robots as well as the new environmental features.

1. **Update the Gazebo simulator:** On the Virtual Machine, open a terminal and run the command

   `cd ~/meam520_ws/src/meam520_sim && git pull`

   This will update the code on your machine to match the current version.

2. **Update the Core:** From Canvas, redownload the `Core.zip` file for your respective language and extract it in the same location as for Lab0.

To run the simulation environment, run the command

```
$ roslaunch al5d_gazebo final.launch
```

2.2 Implementing Your Solution

You will interact with the simulator in the same way as usual, with a few new features. Download the `Final` folder from canvas to get the starter code, which provides an un-implemented solution file with some brief usage examples. Your final submission must include a function which will be called as either:

`$ python final.py red` (Python) or `$ final('red')` (MATLAB), when you are the Red Team.

`$ python final.py blue` (Python) or `$ final('blue')` (MATLAB), when you are the Blue Team.

Note that your solution must automatically handle being run as either red team or the blue team based on the argument passed into your code. Your lynx object should be constructed as

`lynx = ArmController('blue')` or `lynx = ArmController('red')`

in order to communicate with the appropriate robot. To sense the environment, we've included a few new interfaces to the simulator, with brief examples given in the starter code. In particular, using

```
[name,pose,twist] = lynx.get_object_state()
```

will give you either three lists (Python) or three cell arrays (MATLAB) where in the i^{th} entry of each is respectively the unique name, world-frame pose, and twist (linear and angular velocity as a 6×1 vector) of that scoreable object. (Note that these results are not guaranteed to always be in the same order, which is why the unique "name" identifier is given as well.) **Also, note that these are given with respect to the world frame - not the base frame of the robot!** To sense your opponent, we've also included the function

```
[q,qd] = lynx.get_opponent_state()
```

which will return the same data as `lynx.get_state()`, but for the opponent's robot. Finally, we've provided a map of the platforms (not including the turntable) in `maps/final.txt` in the `Final` folder.

With these tools and all the solutions you've implemented throughout the semester, it is now your task to score as many points as you can! There are many different strategies you might employ to handle for example the randomization of the blocks and the presence of your opponent also trying to get dynamic blocks. **Students will likely have very different approaches - that is ideal, and we are excited to see your creative solutions.** It will be helpful to think carefully about the tradeoffs between different strategies, especially in the presence of uncertainty due to your opponent's behavior.

2.3 Submitting Your Code

You will submit your code through GradeScope prior to each phase of the tournament. So if you make it to the Single Elimination rounds, you can update your strategy after Round Robin! Precise submission details will be made available closer to the deadline, however some important things to note:

1. You may submit multiple files that are called from your `final.py` / `final.m` but please do not include any subdirectories in your submission. You may include any of your solutions to prior labs as well as solutions we've shared with you.
2. **No other files other than what you submit and the `ArmController` will be available to your code during execution, including solutions we've provided like `calculateFK` or `calcJacobian`.**
3. Prior to the tournament, there will be some limited opportunities to test-drive your code on our physical machines, perhaps as a "scrimmage" with an opponent. Look out for these opportunities on Piazza.

Code submissions will be worth 10 pts and will be graded according to:

- 5 pts: General code structure, readability, comments, etc.
- 5 pts: Whether the code runs without error during the tournament (Note: This does not require your robot to actually win any rounds, just for your code not to error out.)

3 Final Report (due 12/10, no penalty extension until 12/14)

You should submit a final report describing your approach and results from the final project.

Due date: The final project report is officially due on 12/10 (last day of class). You will all be automatically given a no penalty extension if you wish to submit your final report "late" by 12/14. After that, the standard late penalty of 25%/day applies.

Format: The report consists of two sections: 1) A group report describing your project and results, and 2) An individual write-up discussing the contributions of the individual team member and evaluating the contributions of the other team members

Group Report: The group report is limited to twelve 8.5"×11" pages of single-spaced, Times New Roman 11pt text with 1" margins. References, if included in the report, do not contribute to the page limit. It is okay to submit fewer than twelve pages. Other than this, the format of the report is up to you, but you should make sure that it is clear, organized, and readable.

The report is worth 50 points and will be graded according to the following rubric:

- **Completeness** (10pt): Was the project of appropriate scope? Did it address all relevant questions with no obvious holes?
- **Method** (10pt): Was the approach technically sound and reproducible? Was it complete and free of error or bias?
- **Evaluation** (10pt): Were all relevant results reported? Are the cases chosen sufficient to demonstrate advantages and limitations?
- **Analysis** (10pt): Was the analysis complete, free of error, and based on data/observations?
- **Clarity** (10pt): Was the report clear and organized?

Individual Report: The individual write-up is worth 10 points and should contain the following components:

- **Peer Evaluation:** The specific contributions of each individual to the group effort. Assign a grade to each of your team members (including yourself) according to their contribution. Each team member should be scored out of 2.5 points. The score assigned to each member for the individual report will be sum of scores given to that member from this peer evaluation.
- **Discussion:** Discuss the most challenging issue during the project that you ran into and how you dealt with it. If desired (but not required), you can also discuss how this project demonstrates technical growth on your part since the beginning of the semester. This discussion will not go into your individual report grade, but it will provide us with context when reading the group report and help up to better understand your effort.

Frequently Asked Questions:

1. **Can I use ideas/methods/concepts from other classes?**

   Yes! I support making the project related to your work outside of MEAM 520. You may want to check in with your other professor/advisor if you plan on double counting your project.

2. **I have this really cool idea, but I'm afraid it won't work...**

   Your project is not required to "succeed." The goal here is for you to use the knowledge gained through this course to tackle a more complicated scenario than our labs. Similarly to labs, you should describe what you tried, evaluate and analyze your results, and draw conclusions based on your data and observations.

   Final reports that report "failure" but characterize how and why they failed are as valuable as projects that succeed.

3. **Do we have to do the project in a group?**

   The project is scoped for a group of four people. Under special circumstances, it may be the case that you have to do the project with fewer people in your group. If you fall into this category, you must receive permission to do this before starting on the final project.

4. **Can we talk to you about the project?**

   Of course! Feel free to create a post on Piazza, drop by office hours, or request an individual meeting. The entire teaching staff is available during this time to discuss any ideas you have on final project approaches.

5. **I would prefer to work on an open-ended final project rather than the competition. Can I do that?**

   You are allowed to do this, although I do not recommend it. In this case, you should submit a project proposal as a private post on Piazza, and you will need to receive approval before Friday, Nov. 20, when final project teams are finalized. You will be responsible for ensuring that your project topic and scope are appropriate for the class, and your reports will be graded according to the same rubric as the other final projects in the class. If you do the project with any team members who are not in the class, your report should also include only methods, data, and analysis that are the work of team members in the class.