CS449/549: Learning for Robotics
Bilkent University
Fall 2023
Due date: November 05

Homework 2

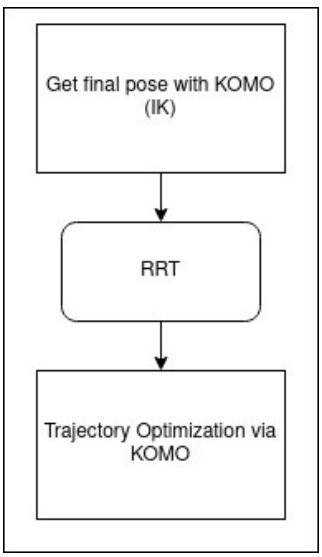
Congratulations! You and your team have been selected for the Cargobot project at a company. Cargobot is a mobile-based robot designed for cargo transportation. The company is entrusting you with the task of building a simulation and motion planning model for this robot. The project consists of three main steps:

1. Path planning for the Cargobot mobile base in the given parkour.
2. Transporting the object to desired locations with the Cargobot in the kinematic world (i.e., no BotOp (dynamics engine) needed. Hint: use Skeleton, KOMO classes).
3. Pushing the object to desired locations in the simulator, which has a physics engine (BotOp).

The specific details for each part of the project are provided in the following sections. Please ensure to read the instructions thoroughly.
NOTE: In your solutions, robot should not penetrate with the walls!

1 Cargobot: Mobile Base

The engineering team has successfully designed a mobile base for the robot. Now, it's your turn to implement the path planning algorithm for the robot. Your objective is to find the (optimal) path that enables the robot to reach the designated goal area from its initial position. To accomplish this task, you should adhere to the scheme that was highlighted in the figure below (as covered in the tutorial session).



First, please import "cargobot_base.g" and "maze.g" files. You do not need to change anything in these ".g" files.

NOTE: It is okay if RRT cannot find any feasible solution in your first try. Please design your overall solution/algorithm in a way to call your motion planner several times until it finds a feasible solution (approximately, 5-10 trial seems to be enough, but try till it finds).
Hint: Check "initWithPath_qOrg" function for the last part.

As an output, an optimized, smooth path is expected which starts from the initial configuration and ends in goal (blue) area.

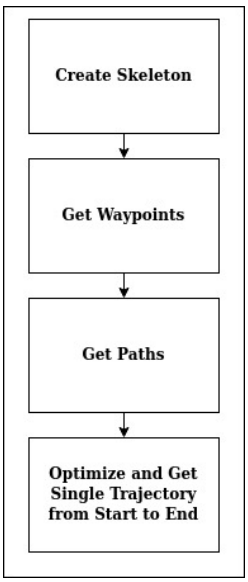
2 Cargobot_Carrier: Cargo Delivery

After you finish the path planning, it is now time for cargo delivery. The engineering team has integrated the mobile base with the Panda robot. You should now pick up the cargo from the initial configuration and place it in the goal (blue) area. While doing this, you should utilize the Skeleton object/class. One of the most important aspects here is creating a Skeleton list that is robust.In other words, in the Skeleton, both the task should be accomplished and intermediate points that the KOMO can handle should be placed, and the correct features should be selected.

First, please import "cargobot.g", "cargo.g" and "maze.g" files. You do not need to change anything in "cargobot.g" and "maze.g" files. In this question, you are given 2 checkpoints and 1 final destination. Robot should pass through these checkpoints.

Hint: You can add more intermediate points (similar to checkpoints) to robustify your solution in your Skeleton.

In this question you should follow this structure:



As an output, an optimized, single trajectory should be seen without any collision. Solu-

tion can be achieved in a kinematic environment and visualized via "KOMO.view_play()" function.

3 Push the Cargo

In the company, the engineering team has devised a novel approach for handling heavy cargos. They encase them within large spheres and aim to push them to their delivery points. Prior to conducting real-world tests, they seek to simulate this process in an environment with physics engines that mirror actual conditions. As such, your task is to push the cargo to the goal area. In this scenario, you will utilize BotOp as the simulation environment and the cargobot_base as the robot. Given that this simulation aims to mimic real-world conditions, you are restricted from directly grabbing or manipulating the cargo in any other manner. Your sole task is to exert force on the cargo until it reaches its designated destination. To surmount this challenge, you will need to formulate your own heuristic. This task differs significantly from the previous question, requiring the implementation of your own algorithms and approaches.

This question is also in a competition format. The best teams will be rewarded with **Bonus points**. Time is one of the important criteria. Try to solve the task as quickly as possible. Additionally, the chosen solution method is another crucial criterion.

Even if you cannot solve the entire path, try to pass as many checkpoints as you can.

First, please import the following files: "cargobot_base.g", "push_cargo.g", and "push_maze.g". You do not need to make any changes in these ".g" files.

For this question, please explain your solution methodology by writing a short report, with a maximum of 2 pages. Also, display your results and record a short video of it. If you manage to solve the entire path, please report your solution time. Otherwise, state how many checkpoints you were able to accomplish.

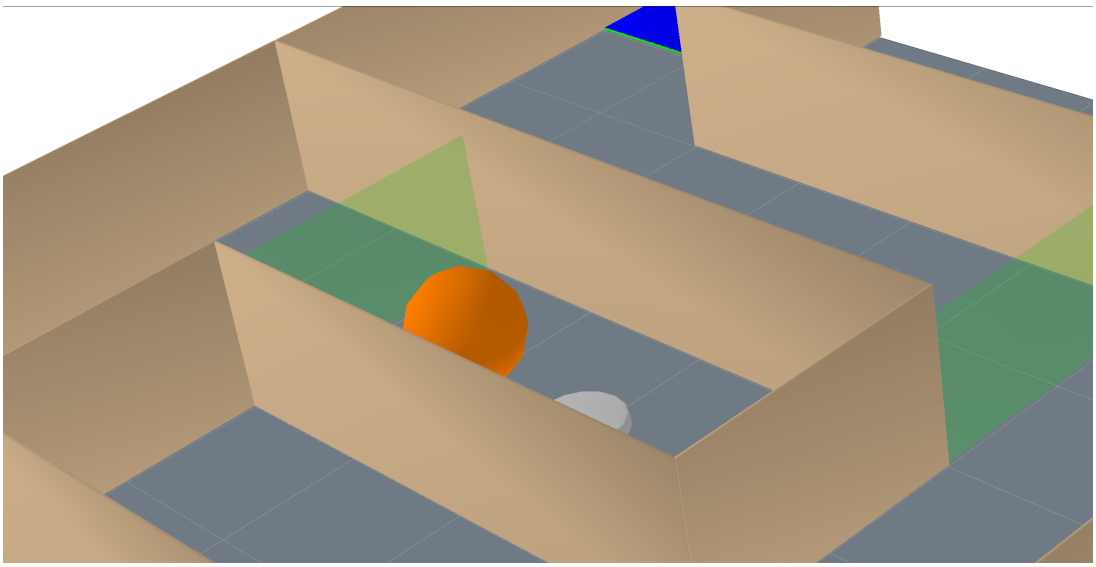


Figure 1: A scene from the parkour

4 Submission

Please follow this instruction to submit your work. You should submit:

- Single ".zip" file which involves all of the files in the working form.
  Please ensure that all the files run smoothly without any need for modification when executed.
  Use this name convention "Groupnumber_HW2.zip"
- Short report for the 3. question
- Short video of the solution for the 3. question