



Task 5B: Mini Theme Run

Objective

The main objective of this task is to do a mini theme run in function mode.

Overview

Now that we have most things in place, let's proceed with a mini theme run!!

The only addition to be made is to feed in a more '**interesting**' sequence of waypoints to your robot, instead of the list of 4 goal poses, as we did in 5A.

Function Mode: Drawing a shape defined by parameterised function, $x(t)$, $y(t)$, $\theta(t)$ as t goes from 0 to T (some positive real constant).

- Where units of x and y are in pixels (to make it easy to compare with feedback of pose which is in pixels - from overhead camera).
- $(x,y) = (0,0)$ is the centre of the image/arena.
- $x(t)$ and $y(t)$ lie in the range $\{-249, \dots, 250\}$ and $\theta(t)$ is 0 for all t ,

Lissajous Curve

Lissajous curve, also called Bowditch Curve, pattern produced by the intersection of two sinusoidal curves the axes of which are at right angles to each other.

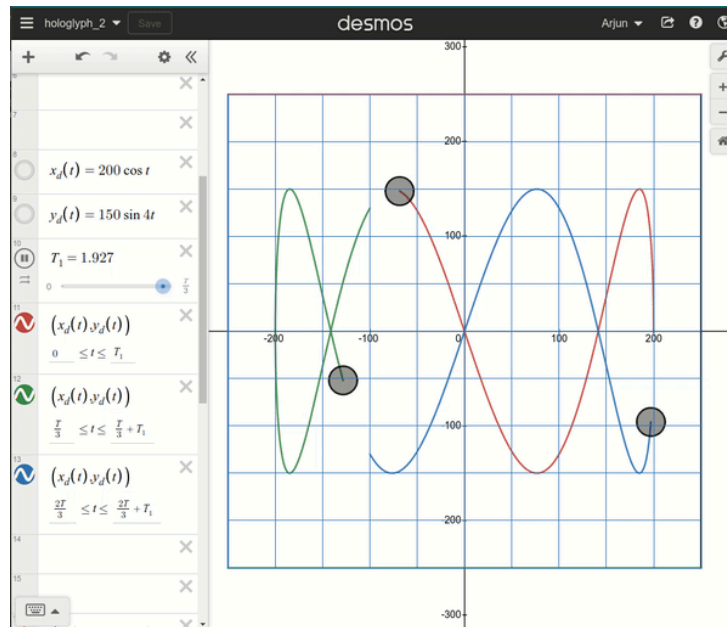
The specific Lissajous curve for Task 5B is :

$$x = 200 \cos(t)$$

$$y = 150 \sin(4t)$$

$$t = [0, 2\pi]$$

Utilize this formula to obtain the desired x and y coordinates, and then use them as the goal pose for your robots. After feeding the goal poses and implementing closed-loop control, your bot's drawing should resemble this.



Evaluator :

- For this task, we are providing you with the evaluator that primarily is meant to evaluate the Time of Run and Average Error. This will enable you to fine-tune your controller based on the results obtained from the evaluator.
- You can download the evaluator from the provided [link](#)
- "Unzip the folder and open the terminal. Go to the 'task5b_eval' folder and copy the command below

```
chmod +x graderV1
```

- To run the grader, you have to execute the following command in the terminal.

```
./graderV1
```

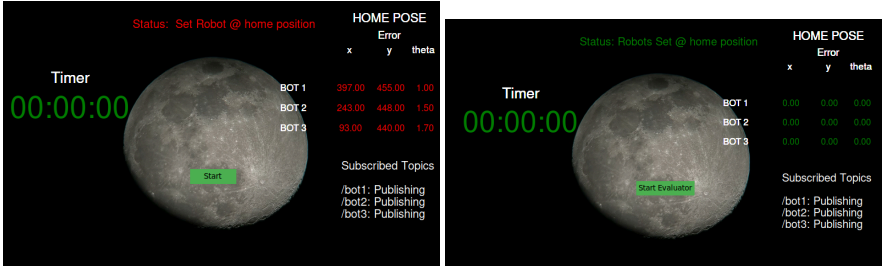
- When you run the evaluator, a 'Welcome' window will appear with a start button. Upon pressing the start button, you can proceed to the next window



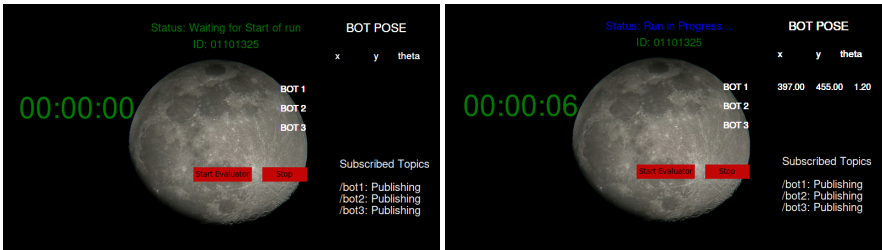
- If you haven't started the publisher node for publishing the pen_pose data, you will see the screen indicating that the poses have not been published.
- Note:** All teams must strictly follow \pen1_pose , \pen2_pose and \pen3_pose for topic names and /geometry_msgs/2DPose message type for publishing the positions of the pens and orientation of the bot. The evaluator will rely on these topics to determine the positions of the pens.
- When you have started publishing the pose topics, you will be able to proceed to the next screen where the Home Pose error will be displayed in red. The evaluator will show the status

as 'Set Robots @ Home Position' in red if the bots are not in their home positions. You need to move your bot to the home pose, which is as follows:

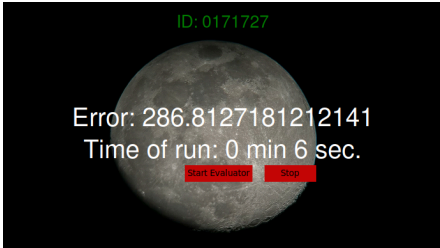
- BOT 1: ["x": 397.0, "y": 455.0, "theta": 0.0] ,
 - BOT 2: ["x": 243.0, "y": 448.0, "theta": 0.0] ,
 - BOT 3: ["x": 93.0, "y": 440.0, "theta": 0.0] .
- After moving the bot to the home pose, you will observe a change in the error color from red to green. Additionally, the evaluator will display the status as 'Robots Set @ Home Position' in green.



- Afterward, press the **Start Evaluator** button. In the evaluator window, the status will appear as 'Waiting for the start of the run.' The evaluator will commence when any of the bots publishes the 'pen down' topic as True. The timer will be initiated, and you will observe the status as 'Run in Progress' displayed in blue.
- **Note :** The teams must publish to the \pen1_down , \pen2_down , and \pen3_down topics when they wish to do pen down. Also the team must follow the message type as Bool for publishing the pen down topic.



- Create a service named Stop_Flag that utilizes Empty.srv as its message type. Invoke this service after your bots have completed their runs. This service call will indicate the completion of the runs and prompt the grader to conclude the run.
- A **stop** button has been provided to allow you to terminate the run if any issues arise. But do note that using this stop button will be considered as manual intervention.
- Finally, the error and run time will be visible on the result screen.



- After the end of the run, Task5b_grader_output.txt will be generated.

Note: The evaluator (and your controller code) will not realise if there is an offset in the drawing (due to pen offset from aruco center, etc.), and if the pen down is working as expected, etc...; for that, we will also additionally perform manual evaluation.

Submission Instructions:

- Record a video of the run. Some key points for the recording:
 - Stationary camera setup.
 - Single, no cuts/edits video.
 - Clearly show the full arena the drawing being made and the laptop connected to the robots and camera.
 - Start before running anything in ROS or the evaluator
 - End after complete end of run.
 - Show the evaluator screen clearly (ID, time of run and error calculated) at the end of the run.
 - Take a photo of the final drawing along with the robots and add/merge it to the end of the video.
 - Submit this video on the portal following the usual rules/instructions described in previous tasks.
 - Also zip the following and submit the zipped file on the portal following the usual rules/instructions as in previous tasks.
 - the txt file generated by the evaluator corresponding to the recorded file
 - all the code written by the team in a subfolder
 - a screenshot of the evaluator and drawing+bots photo added in the video. Additionally you can also share close up photos of the bots built.
-

