Lab 1: Kinematics

CDA 4621
Fall 2024
Lab 1: Kinematics
Due Date: 9-23-2024 by 11:59pm

Objective

The objective of this lab is to learn about motion control and kinematics for a robot to navigate through a set of waypoints.

Requirements

Programming: Python
Robot: Webots 2023b (FAIRIS)

Robot Sensors Utilized

Encoders

FAIRIS_Lite gives access to the encoders on each of the motors on the robot. Each of these encoders are Class attribute of the MyRobot Class, and are referenced by front_left_encoder, front_right_encoder, rear_left_encoder, rear_right_encoder. Which gives individual positional sensors for the corresponding motor encoders. See more details from FAIRIS_Lite Docs. These encoders are Python objects created and maintained by Webots, for more information see Webots Docs. These sensors act as an accumulator to count how many rotations each motor has rotated (in radians). This can be used to calculate how far the robot has relatively traveled.

IMU (Inertial Measurement Unit)

FAIRIS_Lite gives access to the IMU which is attached to the robot. This sensor is a Class attribute of the MyRobot Class, which can be accessed by referencing the compass. This sensor is a Python object created and maintained by Webots, for more information, see Webots Docs. The main purpose (for this Lab) is to use the compass readings of the IMU to determine which direction the robot is facing. This can be combined withe encoder readings to determine the position of the robot in the world frame. The class function to get readings will return values in degrees relative to the orientation provided in Figure 1.

GPS

FAIRIS_Lite gives access to the GPU which is attached to the robot. This sensor is a Class attribute of the MyRobot Class, which can be accessed by referencing the gps attribute. This sensor is a Python object created and maintained by Webots, for more information, see Webots Docs. The main purpose (for this Lab) is to compare your predicted pose with the ground truth provided by the sensor. This will be used to determine which student produced the best results and is awarded the extra credit. **Note:** This sensor should only be used for comparison purposes if used for navigation the student will receive an immediate Zero for this lab.

No other sensors, including Lidar, may be used for this lab.

Orientation Reference

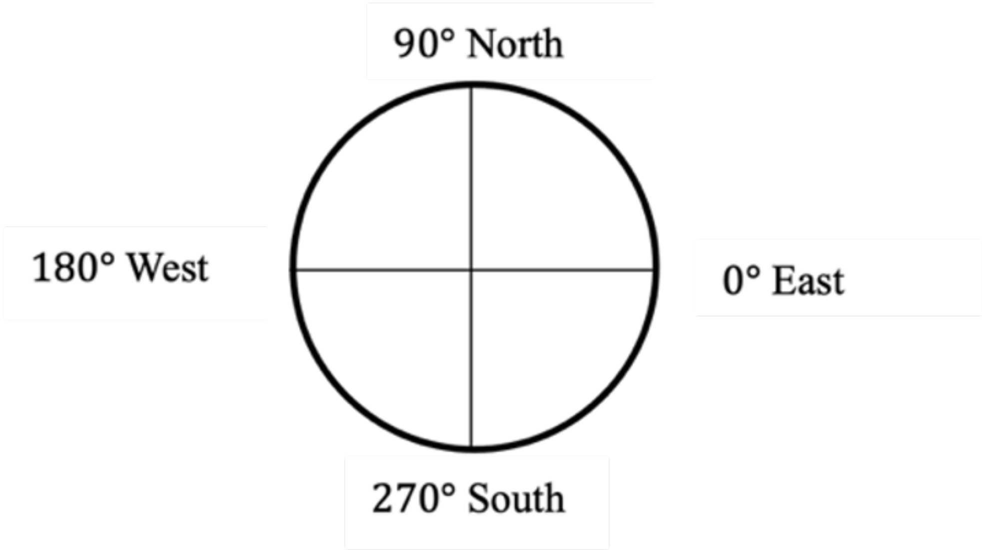

Figure 1. Waypoint Navigation Trajectory

Robot Trajectory via Waypoints

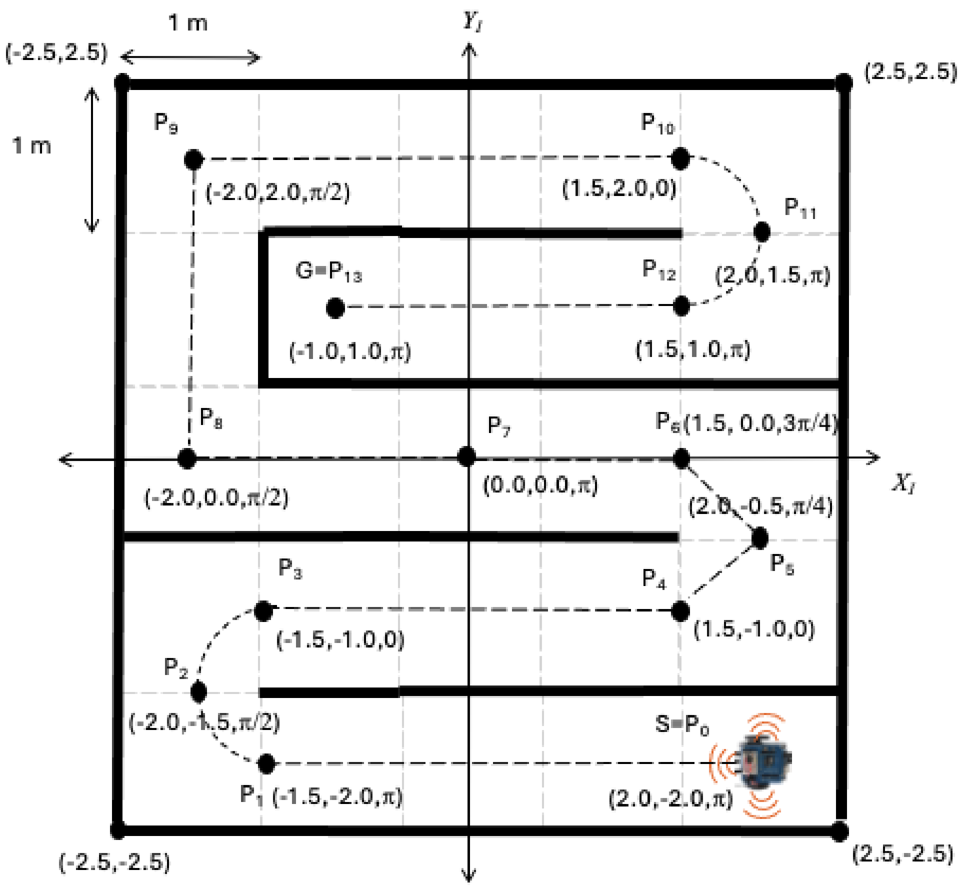

Figure 2. Waypoints to be followed by the robot from S to G.

Waypoint Navigation Task

The robot should navigate through the waypoints shown in Figure 2, where dashed lines represent precise trajectories between some of the waypoints. Navigation between other waypoints does not have to follow a specific trajectory. Trajectories should be made of straight lines and circles of any size. All measurements are either in meters or radians. The robot starts at pose S = $P_0 = (2.0, -2.0, \pi)$ and finishes at G = $P_{12} = (-1.0, 1.0, \pi)$.

- Robot navigates through waypoints $P = \{P_0, \ldots, P_{12}\}$:
  - $P_0 = (2.0, -2.0, \pi)$
  - $P_1 = (-1.5, -2.0, \pi)$
  - $P_2 = (-2.0, -1.5, \pi/2)$
  - $P_3 = (-1.5, -1.0, 0)$
  - $P_4 = (1.5, -1.0, 0)$
  - $P_5 = (2.0, -0.5, \pi/4)$
  - $P_6 = (1.5, 1.0, 3\pi/4)$
  - $P_7 = (0.0, 0.0, \pi)$
  - $P_8 = (-2.0, 0.0, \pi/2)$
  - $P_9 = (-2.0, 2.0, \pi/2)$
  - $P_{10} = (1.5, 2.0, 0)$
  - $P_{11} = (2.0, 1.5, \pi)$
  - $P_{12} = (1.5, 1.0, \pi)$
  - $P_{13} = (-1.0, 1.0, \pi)$
- Compute and follow the trajectory at maximum possible motor speeds.
- Compute and print prior to navigation all robot velocities V_{li} corresponding to the left wheel velocities, and V_{ri} corresponding to the right wheel velocities. Choose V_{li} and V_{ri} to follow the trajectories shown by the dashed line.
- Compute D_i corresponding to the distance between P_{i-1} to P_i, and T_i corresponding to the time it takes to travel D_i.
- Calculate total traveled distances D and total travel time T between S and G.
- Print during navigation actual V_{li}, V_{ri}, D_i, T_i between pairs of waypoints.

Obtain (x, y) position from the GPS sensor and compare it to positions calculated from the encoder. Either enter data into an Excel spreadsheet or use Python libraries such as matplotlib to graph results also include velocities and travel times between waypoints.

Compute and have the robot navigate at a minimal travel time T from S to G independent of the provided waypoints. The robot needs to stop at P_{13} with ±2cm precision (10 points will be awarded to the robot navigating at minimum total travel time in the class). Report final position, time, and provide corresponding analysis in the report.

Extra Credit Metric Reporting

Students must include the following in their final project report:

(1) **Final Pose Prediction Accuracy:** Report how accurately your algorithm predicts the robot's final pose compared to its actual end position. Include the predicted distance and orientation (x, y, θ) relative to the goal, using percent error as in Equation 1.

(2) **Pose Relative to Goal:** Compare the robot's actual pose to the goal location, including distance and orientation (x, y, θ), using percent error from Equation 1.

(3) **Fastest Completion Time:** Record and report the fastest time your robot reached the goal, noting any relevant conditions or settings.

Percent Error Formula

$$E_s = \left| \frac{P_s - V_s}{V_s} \right| \times 100\% \tag{1}$$

Where E_S is the Error Values as a percentage and $s = x, y$ or θ; P_s is the predicted value and $s = x, y$ or θ; and V_s is the Actual Value and $s = x, y$ or θ.

Code Evaluation (90%)

- **Kinematic Computations (35 points):**
  Detailed and accurate pre-computation of velocities, distances, and travel times between all pairs of waypoints. All values are calculated and printed before the robot moves. The printed information is clear, well-organized, and easy to understand, with minimal or no errors.
- **Robot Travels the Correct Path (35 points):**
  The robot follows the exact path as specified, including all waypoints and required trajectories. The movement is smooth and precise, with no significant deviations or errors. The robot reaches the goal with the expected orientation and minimal correction.
- **Program Prints Prior to Navigation (5 points):**
  The program calculates and prints all velocities, distances, and times between waypoints using kinematic equations before navigation. The information is clearly presented and accurate, aligning precisely with expected results.
- **Program Prints During Navigation (5 points):**
  The program prints all velocities, distances, and times between waypoints during navigation using encoder readings and Python time functions. The printed information is accurate, well-organized, and reflects real-time data.
- **Comparison Between GPS and Encoder Computed Localization (10 points):**
  Accurate comparison between GPS and encoder computed localization throughout the navigation. The comparison is well-documented and clearly presented, and any discrepancies are minimal and appropriately analyzed.
- **Extra Points (10 points):**
  The student who achieves the fastest navigation time from P_1 to P_{13} will be awarded 10 points.

Report & Video Evaluation (10%)

A project submitted without a report, or without including the corresponding task video, will not be graded and will receive a “0” for the complete lab.

Report Sections: The report should include the following (points will be deducted if anything is missing):

- **Mathematical Computations:** Show how you calculated the speeds of the left and right servos given the input parameters for each task.
- **Conclusions:** Analyze any issues encountered when running the tasks and discuss how these could be improved. Discuss your results, including any navigation errors. Conclusions need to reflect an insight into what was learned during the project. Statements like “everything worked as expected” or “I enjoyed the project” will not count as valid conclusions.
- **Video:** Upload the video to Canvas showing the robot executing the task (details below).
- **Authorship Statement:** Include and sign the following statement at the end of the report:

  “I developed all the code and written report with the following exceptions:
  [Student name: signature, date]
  [Code sections (specify lines): references and additional comments]
  If any external tools were used to generate code, you need to include the prompt and output.”

Videos: Record a single or multiple videos of the robot performing the waypoint navigation. Include in the videos a written and/or audio description of what the robot is performing during navigation. The videos are critical for task evaluation and must be clear and audible. Ensure that the robot is shown performing the correct path and that the program’s print outputs are visible in the console, as outlined in the lab.

Task 1 Video: Record a single video of the robot performing the waypoint navigation.

Lab Submission

For Lab 1 there are two separate Canvas Assignments: Lab 1 Code Submission and Lab 1 Report Submission. Please follow the Submission policy outlined below. **Note:** Failure to adhere to the submission policy will result in a reduction of 20 points.

Code Submission Policy

Students will need to upload a Python file (i.e. Webots controller) for each Task. The Python file should follow this naming convention *USF_ID_LabX_TaskY.py*, where *X* is the lab number and *Y* is the task number. Example *rockybull5_Lab1_Task1.py*. Additionally, if a student has created any functions that are not present in the controller but rather the *MyRobot.py*, the student will also need to upload that python file as well using the following naming convention *USF_ID_MyRobot.py*.

If custom functions are used in a student's submission and they are not in the controller or the MyRobot Python files, an automatic Zero will be assigned for the code submission.

Each file should be uploaded to the same submission. **DO NOT** upload a zip file, Canvas allows multiple uploads per submission. Failure to meet these submission policies will result in 20 points off.

Report Submission Policy

Students will need to upload their videos to their USF OneDrive or USF Sharepoint and generate a shareable link. Please ensure that you make it viewable to everyone or at the very least the TA and the Professor (use their emails to add them to the access list).

Students will need to upload a PDF file of their lab report. Be sure that the lab report contains the link to the videos and any Metrics requested by the Lab. These Metrics will be used to determine which student receives the extra credit. Use the naming convention *USF_ID_LabX_Report.pdf*. Only PDFs will be accepted.

If the student used ChatGPT to generate code they will also need to provide the conversation used to generate the code. This can either consist of screenshots of the conversation or a shareable link that the system provides. This should be a separate document and not part of the report.

Each file should be uploaded to the same submission. **DO NOT** upload a zip file, Canvas allows multiple uploads per submission. Failure to meet these submission policies will result in 20 points off.

TA may ask you additional questions to gauge your understanding via Canvas Message or MS Teams. Failure to reply may result in point deduction.