IEOR 140: Project 1

Team 6: MoonSoo Choi & Sherman Siu

**Project Description:**

In this project, we built a robot that traces various shapes, including square, triangle, pentagon, semi-circle, and quarter-circle. For the squares, we programmed the robot to trace a square both clockwise and counter-clockwise direction. We also traced a square with and without the *DifferentialPilot* Class and examined what differences these two ways of programming may have.

What was the most difficult part of the project?

::: **NOT** using the DifferentialPilot Class when working on Milestone 1 even though tracing a square was a disaster when we squeezed our physics/mechanics knowledge in order to compute number of revolutions required for a robot to trace a square. *Motor.rotate();* method only accepts **integer** as its input. In other words, even though 385.71… was the number of revolutions required in order to turn 90 degree angle, because we rounded up the number to 386, each turn was off by approximately 0.3 degree rotation, and this minor error cumulated into quite a visible error, and we could not build a perfect square in the first milestone. Furthermore, the robot is not a perfect operator, so even if the math is correct, the differences in the conditions in the motors, for example, would still cause slight differences in direction, distance, and rotations. As a result, teammates clashed a few times, arguing whose calculation method and answer were correct.

What was the most interesting part of the project?

::: Playing with programming robots for the first time in our lives! As all of the teammates significantly lacked knowledge in Java, we probably spent more time than other teams on project 1. Nevertheless, none of the teammates actually felt frustrated for spending a long time on the project. Even though making coding mistakes and watching our lego robot miserably failing at times disheartened us, through eventually succeeding in making the robot properly function, we felt a sense of accomplishment.

What is the approximate error in distance angle per circuit around the square?

Before using the *DifferentialPilot* Class in Milestone One, we had varying errors from 5 to 10 degrees total (after drawing a square clockwise and counter-clockwise, and comparing the original starting position of the robot and finishing position of the robot). After using *DifferentialPilot* Class, the angle error is quite minor (between 2~5 degrees, assuming that there are no other serious technical problems with the robot)

What are the most significant sources of uncertainty and variability that contribute to this error?

* Speed & Acceleration: The lower the speed (and acceleration) of a robot, the more chance a robot has in making an accurate turn. If the robot travels or accelerates too fast, inertia will cause more errors.
* **NOT** using DifferentialPilot Class for Milestone One (we used it for Milestone Two):

Without *DifferentialPilot*, we had to calculate the number of revolutions by hands, and indeed it allows a high chance of producing errors.

* Condition of the motors: Because the robot uses two separate motors to operate each wheel, the differences in each motor’s condition affect the variability. In our case, even though the math in the program indicates that the robot should turn 90 degrees, the robot actually turned a little less (or little more) because one motor is better than the other.

**Link to the Programming codes?**

Milestone 1: SquareTracer1.Java

Milestone 2: SquareTracer2.Java

Milestone 3: GeometryTracer.Java