WORKSHEET

U3.8 Twists and the exponential

Odometry and drift · University · about 35 min

Name _____ Class _____ Date _____

What this lesson is about

A steady command is a twist, a turn about one point. The exact odometry step, the matrix exponential behind it, and the logarithm that plans one smooth arc.

- A command held steady is a twist
- Every steady twist is a turn about one point
- The exact step
- How much it matters
- Going backwards: the logarithm

The task: land on the spot in one arc

The green square is 25 cm to the robot's right and 30 cm ahead of it. The robot must arrive in it facing 120 degrees (a third of a turn clockwise), using **one** steady `drive()` command held for `T = 3` seconds, then `stop()`. Work out the twist with the logarithm, print it as a line starting `twist:` with `w = ...`, `vx = ...` and `vy = ...` (`w` back in degrees a second, the speeds in cm/s), turn it into a command with the full-scale speeds below, and drive it. The printed twist is checked against the logarithm, to a tenth, and every number in it and in the `drive()` should be worked out by your program.

```
from bugbot import *
import math
connect()

V_MAX, V_LAT, W_MAX = 20.0, 15.0, 120.0     # the design's full-scale speeds: cm/s forward,
cm/s sideways, deg/s
DX, DY, TURN = 25.0, 30.0, 120.0            # where to end up, in the start's body frame,
and how far to turn
T = 3.0                                     # how long to hold the command, s
```

Plan your program here, then type it in and press Run.

Do it on the robot

www.bugbotlab.com/learn/u3-8-twists-and-the-exponential/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. The robot lands a little short of the arc, because of the quarter second of lag at the start. The lag holds back all three parts of the twist alike, so holding the command for `T + 0.25` seconds seems the fix. Try it: this robot's turn is strong, so it now over-turns to about 136 degrees, and across ten simulated robots, each a little different, the landing is no better on average. Instead, in a copy outside the task box (the task bans loops), hold the command until `heading()` says the turn is done, stopping a few degrees early for the coast after `stop()`. How many degrees early, how close does the heading get, and why does the position hardly improve? (Compare each robot's real speeds with the design's full-scale ones.)
2. **Finite motions do not commute.** Twists themselves add like vectors, but the motions they produce, their exponentials, do not: `exp(A) exp(B)` is not `exp(B) exp(A)` in general. Turn 90 degrees and then drive 20 cm, or drive 20 cm and then turn 90 degrees: work out both end poses with `exact_step`, then drive both. Why are they different, and which pairs of moves give the same answer in either order?
3. Replace the Euler step in your U3.2 odometry with `exact_step` and repeat the long lap from the project. How much of the final error was the integration?