Assignment 1: Motors, Odometry and Navigation

General comments by Ole Torp Lassen, September 26, 2016

The purpose of this assignment was to practice with the callibration and parameters of the the LeJOS navigation framework. Many aspects influence on navigation accuracy and some fzzy.will always remain. Therefore think in terms of algortihms that can deal with a little unceratinty and inaccurate measurements. Much inaccuracy can, however, be removed by paying attentiuon to th sources of error.

The main challenge is to get the meassures right. Even the factory-supplied meassures of the tire diameter may not be sufficient, depending on the build of the chassis and the weight-distribution, tipping point, sacking wheels etc. The only way to get it right is to

1. to meassure as accurately as possible by hand and since
2. to calibrate systematically.

Part 1 : Direct motor-control

Measure the dimensions of the robot as accurately as possible and substitute values into the kinematics formulae to calculate degrees/seconds for the respecive wheels for some move. Then callibrate first wheel-diameter and then track-width separately from eachother:

1. To calibrate wheel diameter. E.g., calculate degrees/second to drive 1m straight ahead in 10 seconds.
2. Set accelaration lower than default

```
Motor.A.setAcceleration(500);
Motor.B.setAcceleration(500);
```

3. Make sure to synchronize (see the API) motors (synchronize each fragile operation). i.e.:

```
RegulatedMotor[] syncList = {Motor.B};
Motor.A.synchronizeWith(syncList);
Motor.A.setSpeed(X);
Motor.B.setSpeed(X);
Motor.A.startSynchronization();
Motor.A.forward();
Motor.B.forward();
Motor.A.endSynchronization();
try{Thread.sleep(10000);}catch(Exception e){}
   Motor.A.startSynchronization();
   Motor.A.stop();
   Motor.B.stop();
Motor.A.endSynchronization();
```

4. Run the program and meassure the results.
5. Did it drive straight or did it curve? Are the tires straight on the frames? Try switching wheels? Try other wheels?
6. When you get it to drive straight check if it did drive 1m, or more or less? Adjust degrees/second until 1m is reached exactly +/- a few millimeters. To short - subtract a little from diameter, to long - add a little to wheel diameter
7. When you have got it right, calculate the actual wheel diameter from the degrees/second
8. To calibrate Trackwidth,calculate the degrees/second to make a full circle.
9. Mark the start-point on the floor
10. Run the program to see how close it comes back to its starting point.
11. Did it rotate to little? – subtract from trackwidth and adjust degrees/second, to much? - add to trackwidth and adjust degrees/second. It is possible to get it quite close, i.e., within a few millimeters, even with a rather large circle.

When you have calibrated, you can calculate the moves necessary for the racetrack. Remember to synchronize!

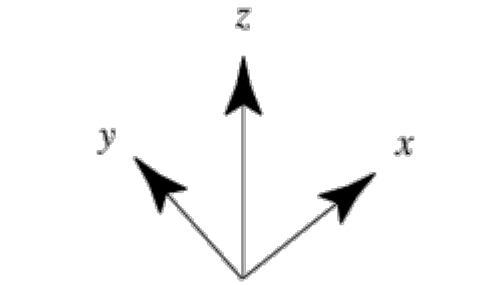
Part 2: Pilot-steering

Use the calibrated parametres, Set acceleration and speed (both angular and linear) sufficiently low to avoid the backlash effect on start and stop. An Arc-radius is meassured from the center of the wheel axis in the units you used for the chassis (mm, cm). You may need to calibrate a little again.

Part3: Navigator and waypoints.

Use the callibrated parameters.

NB: Be aware, that the waypoints are in a right-handed coordinatesystem! I.e., positive x-axis is straight ahead, positive y is left (z is vetically up). This is easilly verified with a simple experiment.



right-handed coordinate system

When using goTo : remember to allow each move to finish (the API has a waitForStop() method for navigators). Since goTo is nonblocking, it will otherwise return immediatly after initiating each move, resulting in ”weird movements”.

For the same reason, check out the `singleStep()` method for navigators when using followPath:

```
1. robot.singleStep(true);
2. robot.followPath();
3. robot.waitForStop();
```

Speed:

MaxSpeed is 100 *actual battery voltage (max 900 degress/second at 9v). Actual speed may be a fraction: specified speed * actual battery voltage/fully charged. Actual battery voltage shows in the top left cormer of the display, max is 9.0.

However, speed and battery shouldn't affect the odometry, but just make the robot move slower.