

# Characterizing FLL Robot Behavior

By Dale Jordan

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

When solving missions for the FLL challenge, consistency often shows up as a major issue, especially for teams that rely on dead-reckoning techniques. Many times these missions seem to run successfully for awhile, even days and then randomly fail. Often, teams will make minor changes to the programming of the robot path, just to see it fail again in the future. After awhile this can lead to a great deal of frustration, and teams usually experiencing this problem don't have a clear understanding of how well the robot is capable of performing.

This paper covers a method for determining how precisely a robot will perform using a couple of simple dead-reckoning tests. Once the dead-reckoning performance is understood, teams can use this information to decide if dead-reckoning will work on a particular mission, or if more sophisticated navigation techniques will be needed.

## Background

When the robot travels in a certain pattern, its final stopping position will vary from run to run. The stopping position will have a positional error in two dimensions ( $x, y$ ), and an error in the angle ( $\theta$ ) that the robot is pointing. Initially, when a robot is carefully set up in base the amount of error in its position is small. As the robot travels out of base, the positional error increases. The NXT internally uses a PID controller to help the robot drive straight. This controller works by adjusting the power levels to the drive motors to attempt to keep the rotation counts of these motors equal. This is a great improvement over not having any kind of control, but it does not eliminate the accumulation of errors. At the end of each run, the robot randomly stops at a different position (Figure 1). If the position of the robot is marked at the end of each run, after several runs a scatter pattern will develop. If enough runs are made, the maximum errors in each dimension will give an indication of the expected variance in the robot's performance.

The issue then becomes, how many runs are necessary to obtain a measure of the true robot variance. If the runs are mostly distributed uniformly then just a few runs should give a reasonable estimate of performance. If one set of runs vary widely from another set, then more exhaustive tests will need to be performed. Also, changing environmental factors such as motor power settings and operating at different battery voltages should be tested.

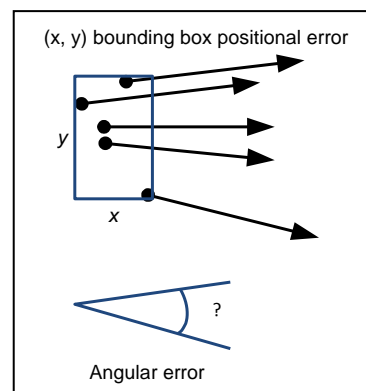


Figure 1 -- Sample Error Scatter Plot

If a team continues to rerun these tests during development, especially when missions start failing, they can easily see if the robot's performance has changed.

## The Dead-reckoning Tests

Two basic, easy to run, tests are used to characterize the robot's consistency. These tests can be run at different travel distances, power levels, battery levels, periods of time, etc. to determine consistency.

- **Straight Test** – For this test the robot is programmed to go forward a specified distance and stop. After the robot has stopped, the final position is recorded. This test is repeated several times to record a scatter pattern.
- **L-Test** – In this test, the robot is programmed to go forward some distance, turn 90°, and then go forward an additional distance.

To demonstrate these tests, a modified version of the ORTOP class robot (Figure 2) was used. Alignment pointers and a measurement guide were added to the cow-catcher assembly. The alignment pointers are used to precisely align the robot at a starting position for each run and the measurement guide is used to mark the robot stopping point and angle.

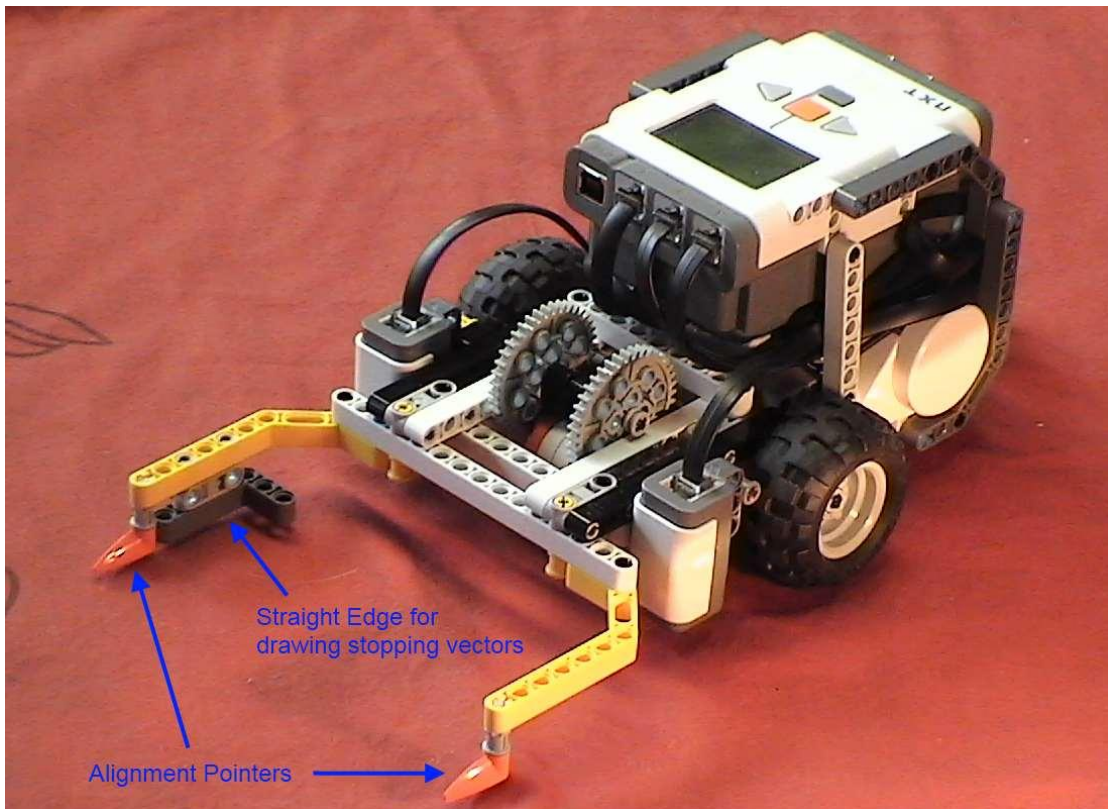


Figure 2 -- Test Robot with Alignment Pointers

## General Field Setup

The field has a starting position with alignment marks so the robot can be precisely aligned for each run. Graph paper is taped down at the desired ending location so the robot scatter plot can be easily recorded and measured (Figure 3).

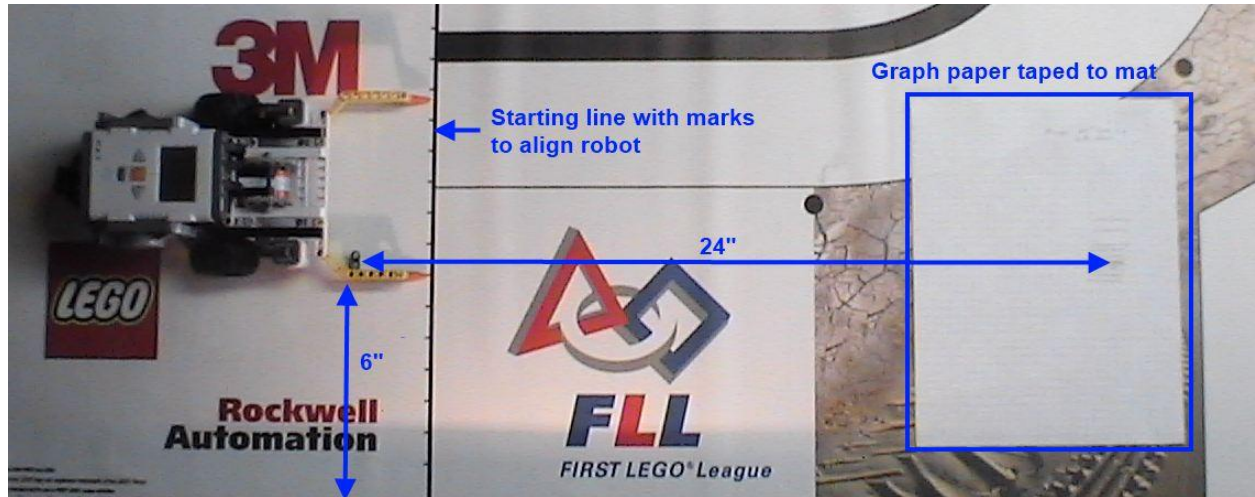


Figure 3 -- Field Setup for 24" Straight Test

## Straight Tests

For the straight tests, combinations of the following parameters are varied and the results measured:

- Distance to travel – 24", 48"
- Motor Power settings – 50%, 75%, 100%
- Battery Level – 8.4V (fully charged), 8.0V

A simple program was written in NXT-G to make the robot go forward and stop (Figure 4). To conduct a straight test, the parameters for the move block are set to the desired rotation count and motor power level.

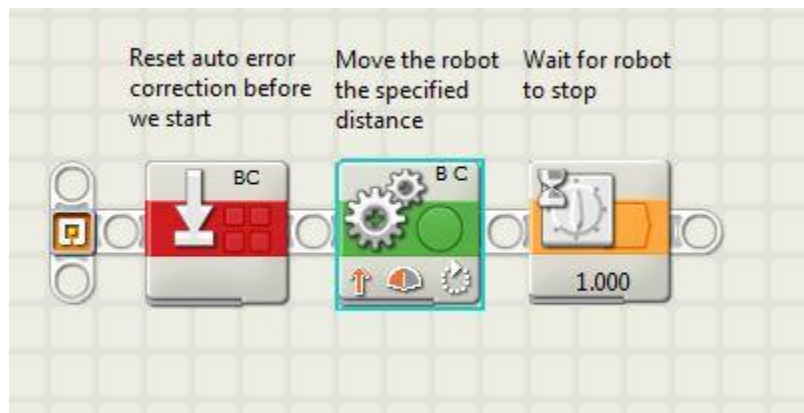


Figure 4 -- Drive Straight Program

To convert the distance travelled to motor rotation degrees, the diameter of the wheel (56mm) was used with the following equation:

$$count^\circ = \frac{360^\circ \times 25.4 \frac{mm}{in} \times length_{in}}{\pi \times diameter_{mm}}$$

This translates to a rotation count of 1247° for 24" and 2495° for 48".

The robot was run 10 times for each chosen combination of parameters. The results of each run were drawn on the graph paper using the measurement guide added to the cow-catcher assembly (Figure 5 ). After a sequence of runs, the (x, y,  $\theta$ ) values relative to the target point were recorded. This was followed by drawing a bounding box around the minimum and maximum stopping points, and then finding the minimum and maximum angle at which the robot finished (Figure 6).

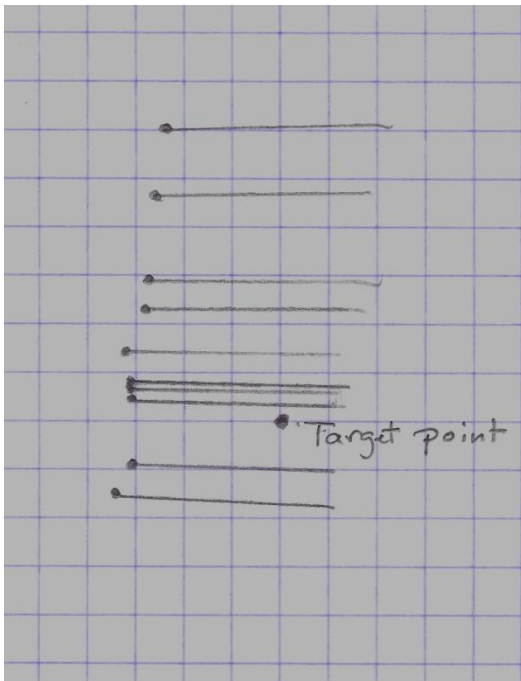


Figure 5 -- Sample Lines from a Run Sequence

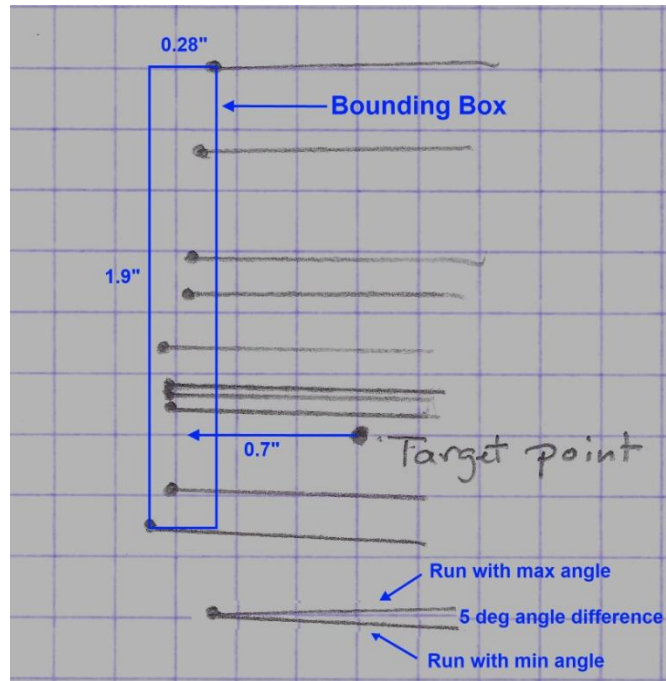


Figure 6 -- Identifying Min and Max Errors

## Data Analysis

### 100% Power for 24"

Several sequences were run with the power set at 100%. Each series gave significantly different results. The robot appeared to be on the verge of losing traction at startup. In order to effectively use the 100% power setting the program would need to be modified to ramp up the power to 100% at startup. This shows that starting the motors at 100% power is a poor choice for dead-reckoning for this robot.

### 75% and 50% Power with Battery at 8.4V and 8.0V for 24"

The motor power level and battery voltage were varied for a combination of 4 tests (Figure 7). For each power level, one set of runs showed a minimal side-to-side variation, where the second set of runs showed a much broader distribution. This indicates a single set of runs is not sufficient to characterize behavior. The test for the 75% power and a full battery charge of 8.4V (Figure 8) was repeated several times. These tests were made at different times. Typically, each sequence of runs has a small variance, but occasionally a sequence occurs with a wide variance. After a few times, all new runs ended inside a composite boundary box of all previous runs. At this point, a good characterization of the robot's variance for these parameters has been determined.

### 24" Tests (50-75%), (8.0-8.4V)

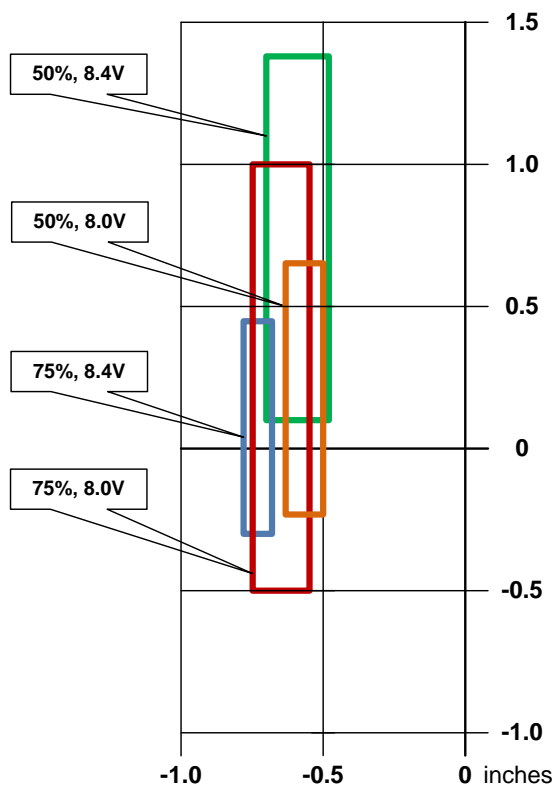


Figure 7 -- 24" Straight Test Results

### Multiple Runs 24", 75%, 8.4V

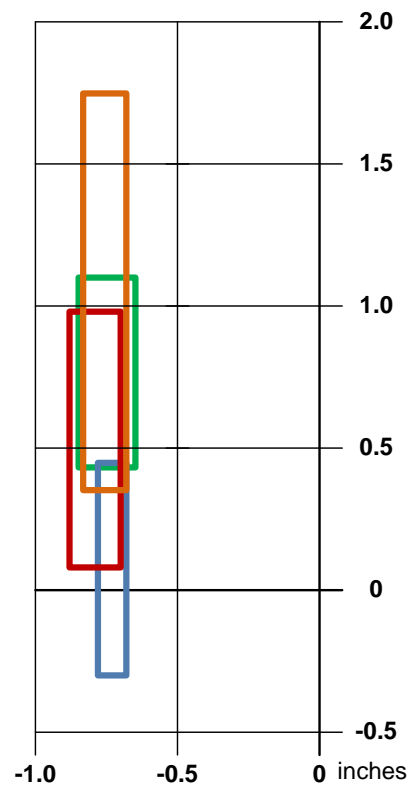


Figure 8 -- 24" Multiple Runs

### 75% Power at varying battery levels for 48"

For the 48" tests, 75% power was just used to simplify the number of tests being run. The battery voltage was varied. These tests were repeated on a second day to compare the variances over time. There was a significant difference in the runs over the two days. This signifies that more tests should still be run before settling on the robot variance for this distance.

## 48" Straight Tests 75% power

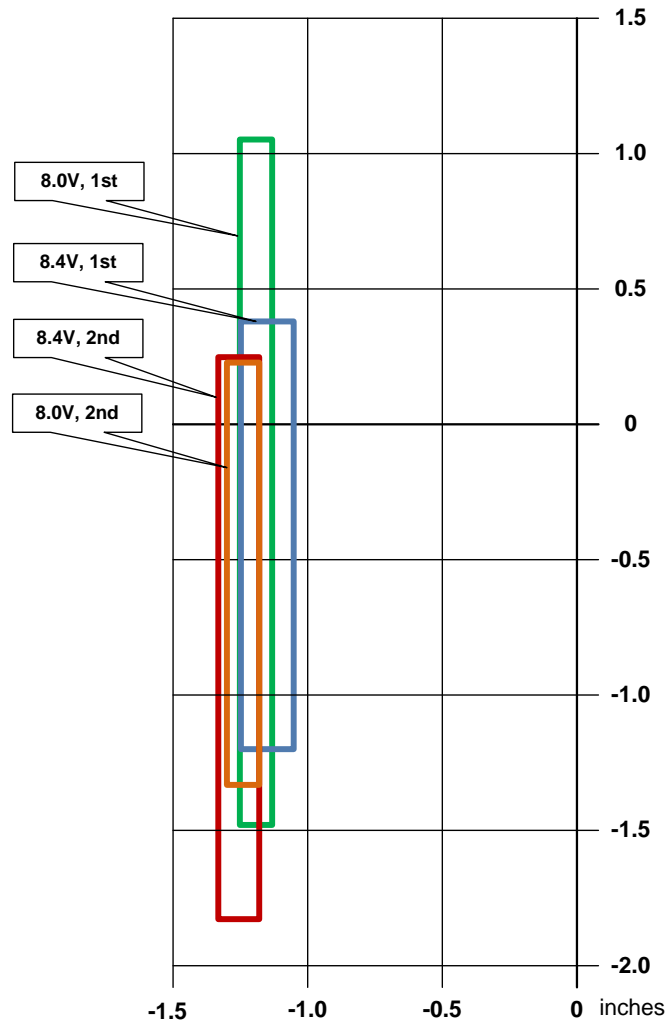


Figure 9 -- 48" Test Results

### Overall Observations of Straight Test

1. The distance the robot travelled is consistently short of the expected value. Several sources for this error were investigated:
  - **Rolling circumference** – The rolling circumference of a tire is less than the free hanging tire size (that which is printed on the tire or the measured tire diameter). This difference may be up to 3%. This could easily account for the shortage in the distance travelled.
  - **Wheel slip on the mat** – The amount of the wheel slip can be estimated by comparing the actual distance travelled at different power levels. The graph shows that generally, less distance is travelled as power is increased. This difference is small and for lower power settings can be ignored.

- **Not travelling in a straight line** – As the robot veers off in a direction, the PID controller takes over to pull it back, producing a wavy travelled path. Carefully examination of the robot's path showed it had less than 1/8" deviation before correction. This does not produce a significant change in path length.
- **Gear slop inside the motors** – The LEGO motors have many gears internal to them. This leads to the wheel being able to move a few degrees before the rotation sensor detects it. This typically only happens during motor starting and stopping. This is more likely to be key contributor to the variance in the direction of travel.

Even though the distance was well short of the target value, the variation is small. Corrections for this can easily be made in a number of ways:

- **Trial and error** – The rotation amount can be iteratively adjusted to get the desired distance travelled. This is perhaps the easiest thing to do and requires little effort.
  - **Compute the effective wheel circumference** – By measuring the distance the robot actually travelled and then using that to determine the rotation counts for the desired distance.
  - **Automatic calibration** – Program the robot determine the proper distance to rotation count scaling. One way to accomplish this is to use two lines (a known distance apart) along with a light sensor on the robot to detect these lines and measure the rotation sensor count between the two lines.
2. There was a very small error in the direction of robot travel ( $< \frac{1}{2}$ " ) for all the different modes.
  3. The side-to-side variations are quite extensive. Both run lengths show a variation of about 3" for all the operating modes.
  4. Some sequences of runs showed a much smaller variation than others. This suggests tests should be repeated over time and with differing parameters to do a better job of characterizing the robot performance.
  5. The worst case difference angle measurements for the 24" travel distance was 6°. This is significant because when the robot makes a turn and travels another 24", as much as a 2.5" variance is added to the side-to-side travel, generally making dead-reckoning unacceptable to accomplish most missions.

## L-Tests

L-Tests provide information about how well the robot will perform when it must turn along its travel. For the L-Test, the robot travelled in a straight line for 24", turned 90° and travelled another 24" before stopping. Two versions of the program were written. The first program (Figure 10) is the most basic using move-turn-move operations. The second program (Figure 11) adds delays after each move and turn operation so the robot's momentum can completely settle down before the next step is started. In both these programs the rotations counts for the 24" travel distances was adjusted from the straight test to make the actual distance travelled very close to 24".

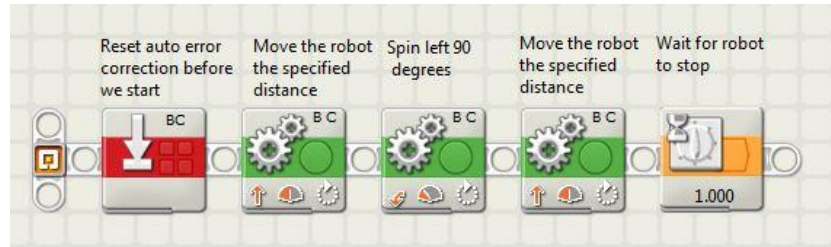


Figure 10 -- Basic L-Test Program (Move-Turn-Move)

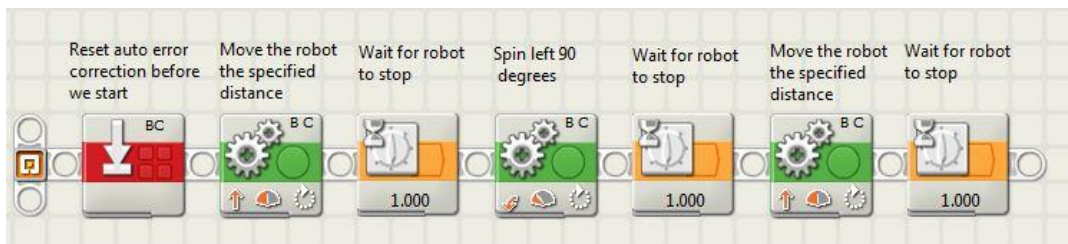


Figure 11 – Modified L-Test Program with delays to let robot completely stop before continuing

## L-Test Observations

All tests were performed at the 75% power level. Eight sequences were run at varying voltage levels using the basic and delay programs mentioned above. These tests were run over a period of 2 days, with no changes made to the robot other than charging and discharging the battery. With this limited testing, a wide ranging set of results occurred (Figure 12).

# L-Tests

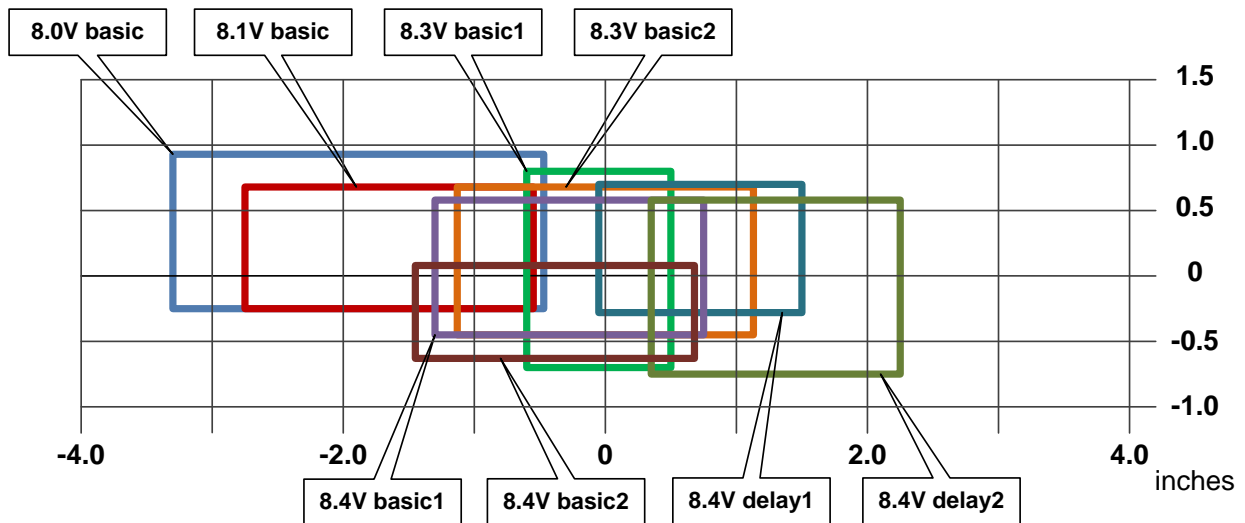


Figure 12 -- 75% L-Test Scatter Plots

1. **Basic vs. Delay** – The robot smoothly slows down at the end of a move or turn operation, so there should be little momentum in the robot to influence future operations. However, a significant shift in the robot position is seen when the additional delays are added. This signifies that any momentum in the robot is likely to have an impact on the next operation. Teams should consider this effect when designing how a robot will move when travelling between locations or when performing several operations in succession.
2. **Battery Voltage** – Battery voltage has a significant effect on the final position of the robot. As the battery was dropped from a full charge of 8.4 to 8.0 volts, the robot’s final position shifted substantially to the left.
3. **Robot angle** – From the data collected (not shown in Figure 12), the angle at which the robot stopped at varied between  $-7^{\circ}$  and  $+6^{\circ}$ . If the robot were to travel another 24”, this could account for as much as a 5” error in the side-to-side position.
4. **L-Test vs. Straight Line** – For the L-Test and Straight Test, the robot travels the same total distance. The only difference was the presence of a turn in the middle. By constructing bounding boxes for the worst case errors for the 48” Straight Test (Figure 9) and for the L-Test (Figure 12) and combining them on one graph (Figure 13) a comparison of the errors can easily be seen. For the straight test, significant errors only grow in one dimension, but when adding a turn, the errors rapidly accumulate in both dimensions.

# L-Test vs. Straight Test

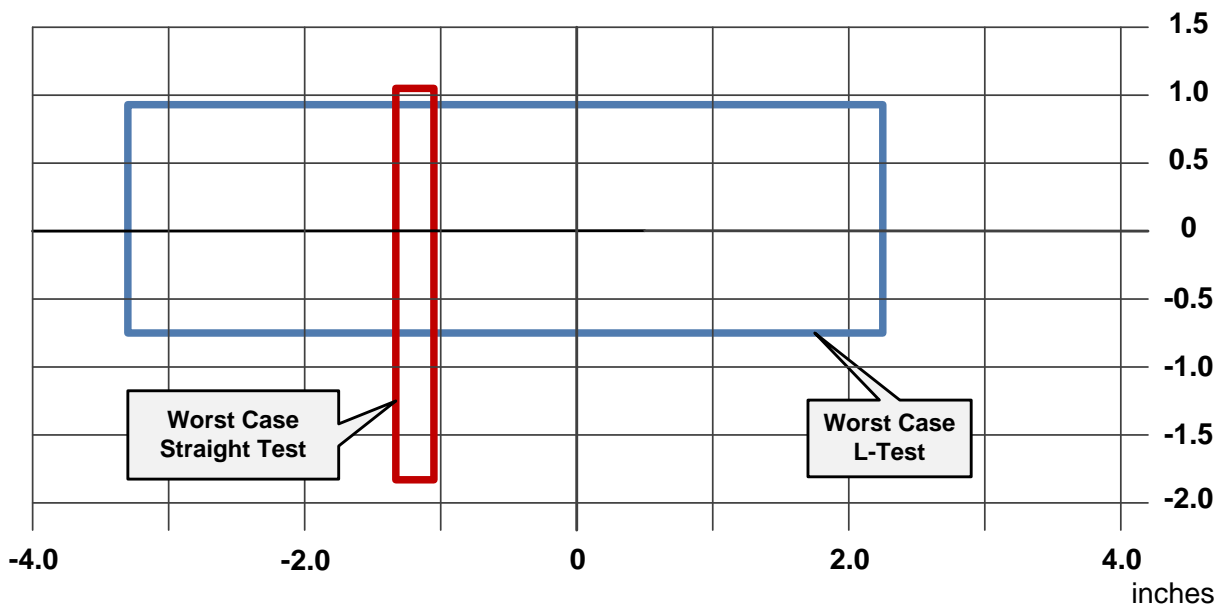


Figure 13 -- L-Test Comparison with Straight Test

## Summary

Straight and L-Tests provide a simple way to characterize the robot's travel variance. Many conditions impact this variance. The most common are: motor power level, battery voltage, and robot momentum. To characterize the robot performance, these parameters should be varied. In addition, it may require running tests several times before the most significant variances are found.

Once these variances are established, reasonable decisions can be made about when to effectively use dead-reckoning to successfully complete FLL missions.