Challenge #4: Predict Position

Objective: Understand how to predict the position of an object at any time.

Background:
Q: Now that we understand displacement, velocity and acceleration, how can we put them all together to predict the behavior of objects?

A: Newton's laws of motion!

You already know one of the laws: distance = speed x time.
Let's rewrite that as

Displacement = velocity x time or just

d=vt.

So we should be able to predict the displacement of an object moving at velocity v if we know the time.

That's a special case when acceleration = 0. What about when acceleration is not zero but constant?

Let's take our special case: d=vt and add a mystery term.

So d = vt + ?

Well, we know that the mystery term must have the dimensions of displacement, or in our case cm, so whatever it is it has to end up having cm as the units.

Well we know we want acceleration as part of that mystery term, and we know that acceleration has the dimension displacement/time/time, and the units cm/s/s or cm/s^2.

Q: What can we multiply cm/s^2 by to get rid of the denominator (s^2). What are the dimensions of that?

A: We have to multiply by cm/s^2 by s^2. The dimensions are time squared.

Now we know the mystery term: at^2 ! Well, sort of. We have to put a ½ in front of the a for it to be completely correct (calculus, sorry)

So now the whole equation for predicting the displacement of an object is

THIS: $d = v_i t + 1/2at^2$

Wait: what's that v_i thing? That's just the initial velocity. If our object starts at v = 0, then the whole first term disappears. If we rewrite this as $d = 1/2at^2 + v_it + c$, it looks suspiciously like the equation for a parabola, no?

This is one of Newton's equations of motion. It means that we can pinpoint the location of an object at any time if we know it's initial velocity and its acceleration.

Tasks:

1. Run "challenge#4" on the 'bot to generate a datalog.
    a. Download the datalog as challenge4.txt
    b. Go to Vernier Pro and File:Import from:challenge4.text.
    c. Delete the empty and junky cels.
    d. Double click on the column 1 header and change the title to time, short name to t, and units to ms.
    e. Double click on column 2 header and change title to total rotation, short name to rt, and units to degrees.
    f. From the menu Data:New Calculated Column, Name to velocity, short name to v, units to deg/ms. Now click on the function tab and choose delta. Click on the variables(column) tab, and choose total rotation. Now type the slash "/" and then function delta, variables, time. You should now have a line that looks like this: delta("total rotation")/ delta("time"). This is of course velocity. The delta function takes the change in the quantity since the last cel.
    g. One last column- Look at the time column and make a new calculated column that subtracts the value of the first time cel from the time. This makes it so that the time starts from 0. Label it Adjusted time, short name t adj, units ms.
    h. On the graph pane, option click to get the menu, and check total rotation on the y axis, and then check the right y axis box, and select velocity.
    i. Make sure that adjusted time is on the x axis.
    j. Remember, the slope of the velocity is the acceleration. You can use the "analyze:linear fit" to get the slope of the line of velocity by putting a checkmark next to velocity when the dialog box comes up.
    k. Write down the slope value below.

    Acceleration value, a (in deg/ms/ms) for your robot:

2. Modify the accelerate robot smooth loop counter to stop when timer 1 > 1 second. Predict the distance the robot will go using the following equation:
    $d = 1/2at^2 + v_it + C$
    where the initial velocity can be gotten from looking at the graph of velocity
3. Run the modified program and see how close your robot got to your predicted displacement distance.

4. Write down the prediction and the actual distance here:
    Predicted displacement: Actual displacement: