Write a simple C/C++ program that reads a tab-separated data file (attached) consisting of encoder samples. Each sample consists of a number of encoder ticks (L\_ticks and R\_ticks) since the last sample, and rotation direction (L\_direction, R\_direction, 1 = forward, -1 = reverse) for 2 wheels, and an integer timestamp (time). The timestamp represents the number of milliseconds since the program started.

Your program should assume that the robot starts at location (0,0) with a heading of 0 (aligned with the x-axis).

Your program should use the encoder samples to incrementally update the pose of the robot given the following physical configuration:

\begin{figure}\begin{center}
\begin{tabular}{ccc}
\psfig{file=figs/pioneer.ps,wi...
...ffdrive.eps,width=2.6in} \\
(a) & & (b)
\end{tabular}
\end{center}\end{figure}

b

b = 300 mm

r = 20 mm

Where b is the baseline distance between the wheels, and r is the wheel radius, both given in millimeters.

The number of transitions on the encoder wheel is 64 – 32 black stripes and 32 white stripes. The encoder ticks on every transition (both white to black and black to white).

Your program should output the pose of the robot after each update to a tab-separated file consisting of x, y position (in meters) and heading (in radians).

Your program should calculate pose using 2 different methods. First, using the complete kinematics model, and then using the approximation, generating two separate output files.

Using excel, plot the x and y paths on the same graph. In your lab report, describe any differences between them.

Next, calculate the error between the full kinematic model and the approximate model, by calculating the Euclidean distance between each pair of points in the two paths. Plot this distance over time and include it in your lab report. Discuss any observations you can make about the error trend.

Your code should be well designed, well commented, and well-structured such that it is portable and usable across many different situations. Keep in mind that this is code that will someday (soon) take readings from live sensor data instead of a file.

Submit your source code, output calculations, and lab report.