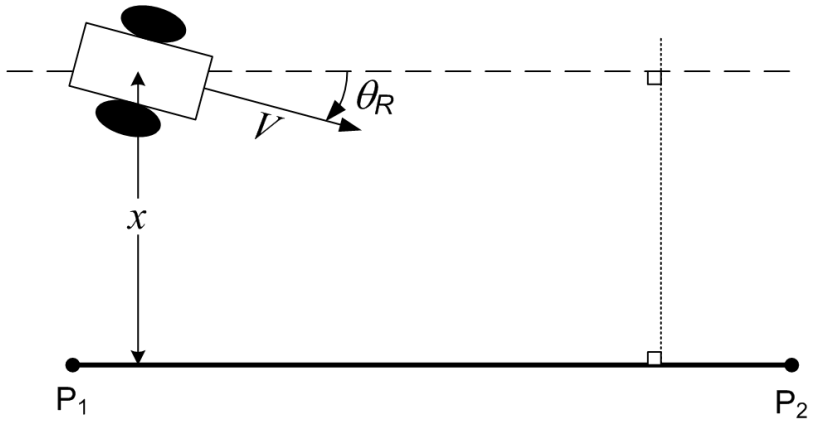
Assignment 4, Due November 21th, 2011
(Worth 10%)

1) **70%**) In class, we saw different components that can be used in a controller:
   - a proportional component (P)
   - a derivative component (D)
   - an integral component (I)

The goal of this question is to get you familiar with how to use this type of controller, and particularly in the context of robotics. As such, you will get a differential drive robot to perform a path following task.

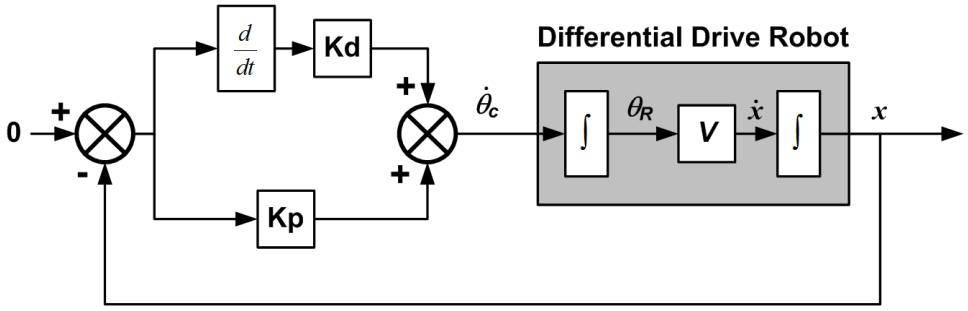


In terms of variables for this problem, we have:

- distance x to the wall;
- desired distance x_d to the wall
- desired heading of the robot , relative to the wall θ_d;
- the desired rate of turn of the differential drive robot ω_R, (*turnrate*);
- actual heading of the robot θ_R;
- the *fixed* forward velocity of the robot V.

In the context of control theory, wall following can be defined as trying to maintain $x = x_d$. This also implicitly requires $\theta_d = 0$, otherwise the robot would drive away from the wall.

We saw in class how we can use a P-D controller to maintain the trajectory of a robot on a line, while driving at a constant velocity V. The P part was getting the robot to move towards the line, and the D part "cued" the controller to reduce the correction as it gets closer to the goal. Without the D part, the robot was oscillating constantly around the line. The block diagram for this approach is shown below. The leftmost 0 simply indicate we want to be at a distance 0 from the wall. The grey box is a simplified model of the differential drive robot.



Using ROS and stageros, write a program that will make the robot follow the closest obstacle. You can start the robot fairly close to one obstacle. Use the laser data to identify the closest obstacle. Test your wall following in the example worlds provided in the previous assignments; especially the cave world. Ensure that the robot goes around obstacles and also around the boundary.

Hints

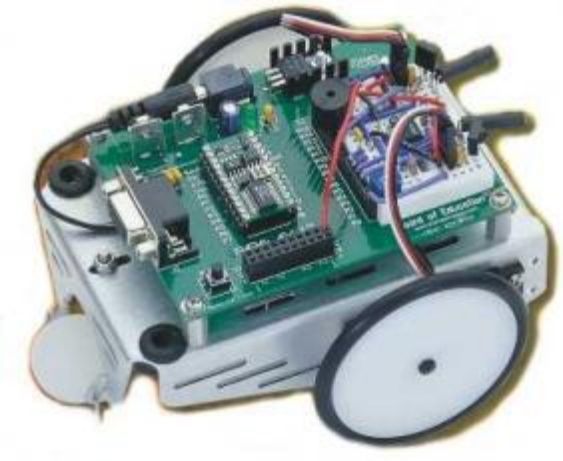
- **Direction**
  If your system diverges, it might be because the commands have the wrong direction. Double check that by placing the robot near an obstacle, and look how your error and commands change. If this is the source of your problem, simply multiplying by -1 the troublesome value would fix it.

- **Computing angles**
  Use atan2, not atan.

- **Phase Unwrapping**
  Remember that computations on angles are tricky. After any computation, you might want to shift the angle between $-\pi$ and π by adding or subtracting multiples of 2π. Simple examples of faulty computation if you do not do that, with the correct phase unwrapping (example given in degrees but the same applies for rad):
  - 0-360=-360 (wrong) -360+360 (phase unwrap) = 0 (correct)
  - 170 - -180 = +350 (wrong) +170 - 180 (phase unwrap) = -10 (correct)

- **Capping the Heading Command**
  If the robot is far from the wall, the heading command θ might be more than $\pi/2$ rad, in which case the robot will go in the wrong direction, or start making spirals. You might therefore want to cap this command c to be within +/- $\pi/2$.

2) **30%**) This part of the assignment serves as an introduction to the Boebot.
   a. Create a team of two. Email the TA the names in your team.
   b. Take one Boebot per team and ensure that it is operational. Run some sample test programs.
   c. Read through the manual: http://www.parallax.com/portals/0/downloads/docs/prod/edu/WebRoboticsv3.0.pdf
   d. While driving straight forward towards an obstacle use the speaker to emit a sound inversely proportional to the measured sonar distance. Stop at a safe distance to the obstacle.



What to submit:

Submit the source code together with a written report through WebCT.