Chapter 4

Results & Discussion

4.1 Results

The proposed algorithm for controlling a linear platoon of mobile robots was tested in simulations and experimentally on real small-size mobile robots. In the following table 4.1 the results obtained from the experiments on real mobile robots are presented

Table 4.1 Summary of the results

Read the sensors Notable : 0≈ 1.9~3.3 volt 1≈ 3.4~5 volt					**Value of Average sensors volt**	**Value of PID**	**Move the left motor (motor A)**	**Move the right motor (motor b)**	**Move the robot**
S1	**S2**	**S3**	**S4**	**S5**					
0	0	0	0	0	0	3	Left	left	line is in the exact center
1	0	0	0	0	~ 1	~ 220	Left (CW)	right (CCW)	Shift left
1	1	0	0	0	~ 3.3	~ 380	Left (CCW)	Slow left (CW)	Shift left
0	0	0	1	1	~ 4.4	~ -400	Right	Riht (CCW)	Shift right
0	0	0	0	1	~ 4	~ - 200	Right	right	Shift right
1	1	1	1	1	5	255	Left(CCW)	Right(CCW)	Returned to line

4. 1.1 some non-logical results

If ReadSensors()Return value may be fractional also, like it is 2.5 when line is beneath sensor 2 and or like it is 2 when line is beneath sensor 2 or like it is 3.6 when line is beneath sensor 4 or like it is 3.5 when line is beneath sensor 1sensor5 no problem Because dealing with the average (*weighted mean*)

4.2 Discussion

Once you have PID running in your robot, you will probably notice that it still doesn't follow the line properly. It may even perform worse than it did with just proportional! The reason behind this is you haven't tuned the PID routine yet. PID requires the Kp, Ki and Kd factors to be set to match your robot's characteristics and these values will vary considerably from robot to robot. Unfortunately, there is no easy way to tune PID. It requires manual trial and error until you get the desired behavior. There are some basic guidelines that will help reduce the tuning effort.

1.Start with Kp, Ki and Kd equaling 0 and work with Kp first. Try setting Kp to a value (generally less than 10) and observe the robot. The goal is to get the robot to follow the line even if it is very wobbly. If the robot over shoots and loses the line, reduce the Kp value. If the robot cannot navigate a turn or seems sluggish, increase the Kp value. Try your level best to not not introduce Kd or Ki and fine tune Kp.

2. Once the robot is able to somewhat follow the line, assign a value (generally less than 1) to Kd (skip Ki for the moment). Try increasing this value until you see less wobble.

3. Generally finely tuned values of Kp and Kd is sufficient for pretty efficient line following and Ki need not be introduced.

4. Once the robot is fairly stable at following the line, assign a value of .5 to 1.0 to Ki. If the Ki value is too high, the robot will jerk left and right quickly. If it is too low, you won't see any perceivable difference. Since Integral is cumulative, the Ki value has a significant impact. You may end up adjusting it by .01 increments.

5. Once the robot is following the line with good accuracy, you can increase the speed and see if it still is able to follow the line. Speed affects the PID controller and will require retuning as the speed changes.

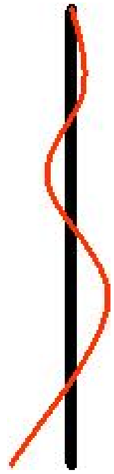

Figure 4.1 Badly tuned

Figure 4.2 Finely Tuned

If the PID controller parameters (the gains of the proportional, integral and derivative terms) are chosen incorrectly, the controlled process input can be unstable, i.e. its output diverges. Instability is caused by excess gain.

So, for stability, gain must not be too large.

Generally, stability of response is required and the process must not oscillate for any combination of process conditions and set points, though sometimes marginal stability (bounded oscillation) is acceptable or desired

Problem faced with PID controllers is that they are linear, and in particular symmetric. Thus, performance of PID controllers in non-linear systems is variable.

In this case the PID should be tuned to be over value, to prevent or reduce overshoot.

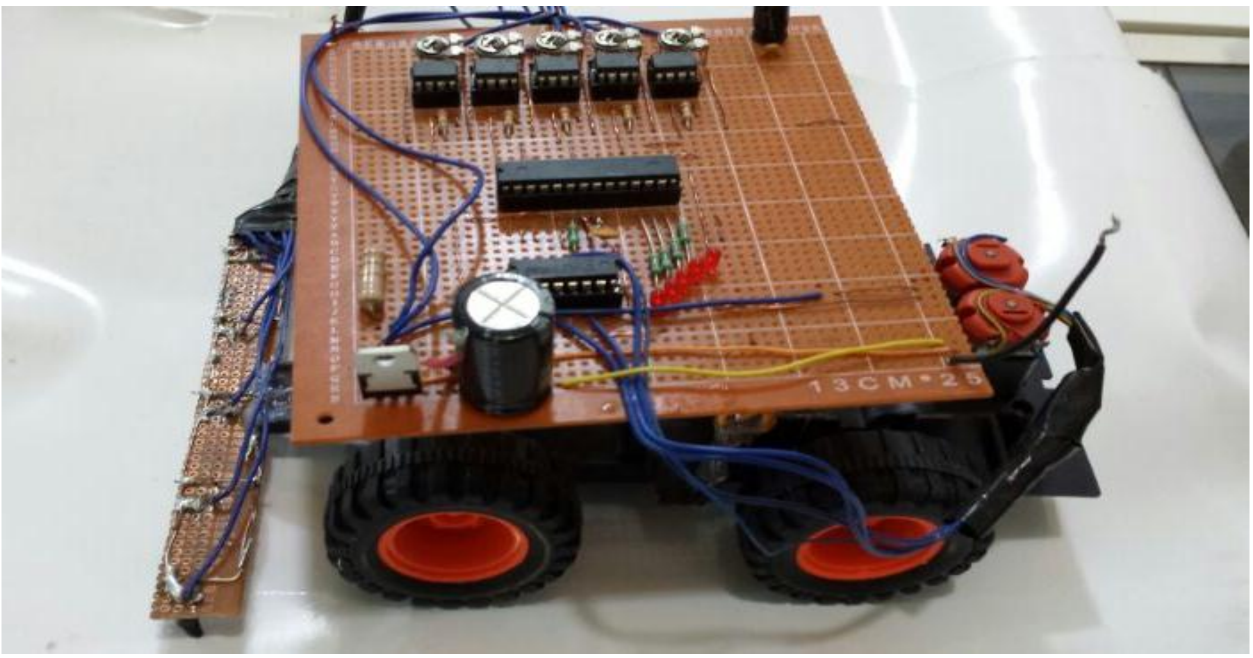

Figure 4.3 Design application practical 1

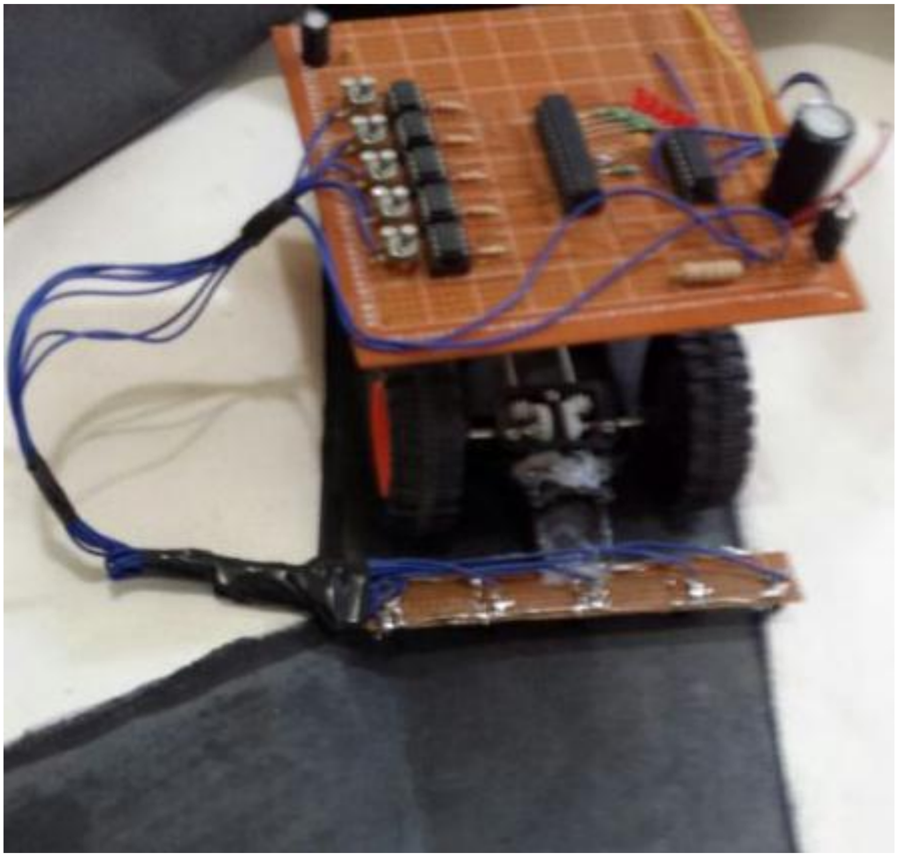

Figure 4.4 Design application practical 2

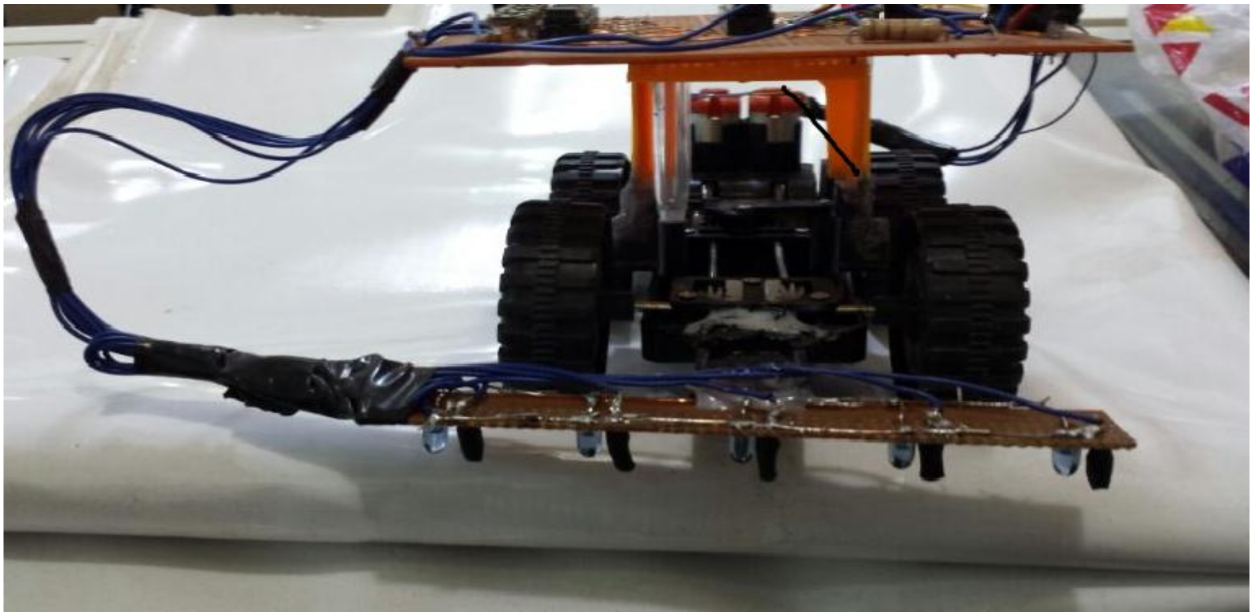

Figure 4.5 Design application practical 3

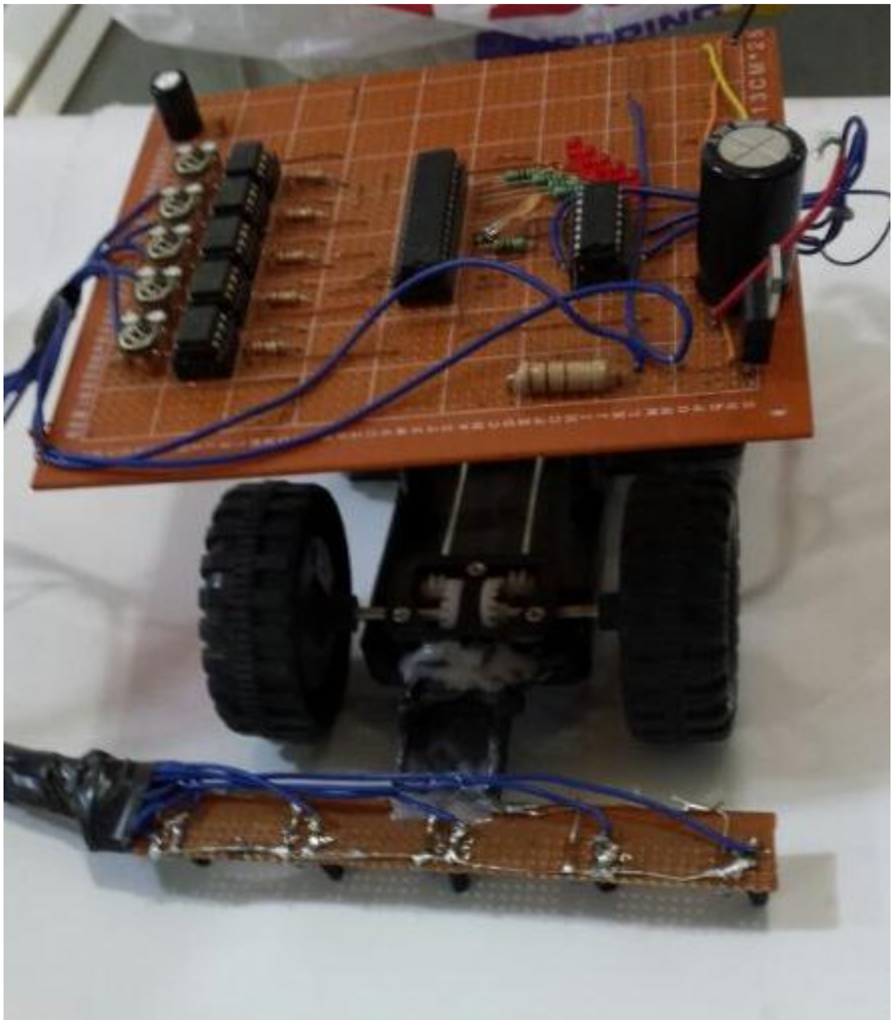

Figure 4.6 Design application practical 4