Car Wheel Velocity Controller

Garrett Wilson and Nathan Zimmerly

ENGR 352

June 7, 2016

Contents

1   PID Controller

- Based simulation on
  http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=ControlDigital

- Based code on Frohne's code, but adjusted for:

  - not using Bluetooth – commenting out Bluetooth code and hard-coding in the speeds we want

  - using a different H-Bridge – switching how the controller sets direction since this H-Bridge has four control wires, changed PWM output ranges since forward and backwards are treated the same in this case (just with different control outputs)

- turn on/off the two LED's on the board when interrupts arrive from the left and right wheel sensors, so we know if they are working

- make integral sum terms static so on every interrupt they aren't reset to zero

- limit the control output to be between 0 and 100 since that is the range for our PWM output

- set a max value for the integral sum terms

- tried adding in the set point into the control signal, and then removed it after talking with Frohne and realizing that the error won't actually go to zero when it's going the desired speed

- allow negative control signals which turn the wheels in the opposite direction instead of cutting the signal off at zero

- lots of PID tuning

1.1   ??

2   Future Work

- We encountered issues debugging since we tried having printf statements while it was running, which changed how the controller acted since sending those over serial was too slow. Thus, Frohne recommended we save the values into an array and then after the controller runs for a bit we then print out to the console.

- Something was wrong with our measurements. Michael and Chris also ran into this, and they thought it had to do with both wheels operating at the same time. They didn't have the issues when only controlling one wheel. This might be the case here too, but in either case, we'd need to fix these measurement issues if we are going to get the controller to work well.