T-106.5300 Embedded Systems

Students: **411398 – Daniel Koslopp**

411411 – Talita Tobias Carneiro

Car project Assignment Documentation

This document has the objective to describe the approach used for controlling speed and direction as well the implementation of the program for the Car project.

Firstly, it is wise to list the requirements for this project, and then based on that we can think about an appropriate controller for the problem. The requirements were to control speed and direction in the manner that the car can maintain good speed through the entire track with smooth oscillation without lose track.

The block diagram in figure 1 shows by a control perspective the approach used in this project. It shows both controller used, speed and direction.

Hence, let us start with the direction control.

Since we do not have the physical model of the car, we can use an error-based controller. It means that we use the error provided by the difference between our reference and the measurement in our bumper sensor.

The measurement in this case is a function of weight, in which each sensor of the car represents a weight of 0.25. The first sensor has a weight of -1, which is increased by +0.25 until reaches +1, as in figure 2. Thus, depending on which sensor is active the measurement will correspond to the exact direction.

On the software, all weights of the active sensors are summed and divided by the number of the active sensors.

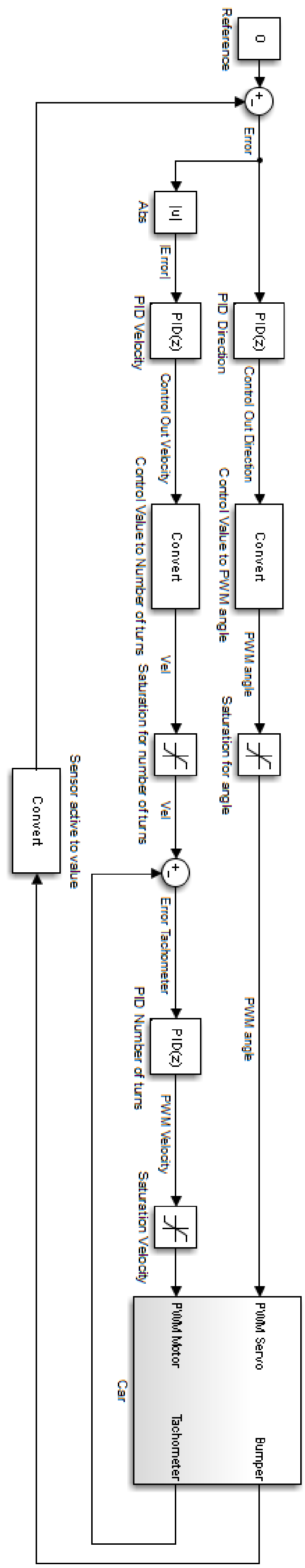

Figure 1. Control block diagram.

For instance, let us consider the case that two sensors are active. The -0.75 and the -0.5. The total measurement error will be (-0.75 + (-0.5)) / 2 = - 0.625. The signal will corresponds to the direction (right or left).

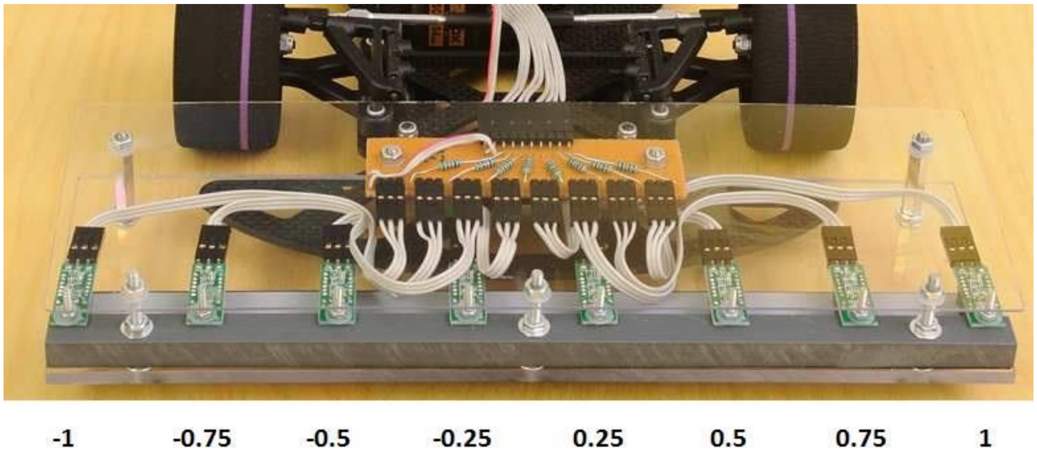

Figure 2. Weighting of the sensors.

The error is calculated by taken the difference between the reference (which is zero) and the measurement error. It is important to understand that the reference is zero in order for the car is always in the middle of the track (line).

The direction controller output is used in a function that gives the direction for the PWM. Done by writing in the respective register a rounded number from an equation that considers the variation of the angle (90 to -90), the time range for the direction of the PWM (1.15ms to 1.80ms), the frequency of the comparator and a variable called saturation that states the maximum and minimum limits (saturation) of the PWM.

The speed control was implemented in a more indirect manner. We can say that it has a subsystem in it. The idea to use a PID control is the same, but in two stages.

Firstly, we use the absolute value of the prior error (direction controller) yielding by a PD controller to VEL. This variable VEL is now the reference for our speed control, where its unit is number of turns expected. The measured signal comes from the tachometer (measured_pulse variable), hence the error is taken from the difference between VEL and measure_pulse. The error is used in the speed controller (PID controller), which yields the output signal controller_out_tacho. This process is more understandable by considering figure 1.

The control signal then is applied to the register of the PWM of the motor in a function that depends on a maximum and minimum limit (saturation) and a rounded equation of the controller signal and the frequency of the PWM.

As an extra feature, we implemented a protection function, which is used in the case that car is lost for two seconds without finding the line. In this case, it is turned off for safety reasons.

Some practical aspects of our code are:

- For the direction controller (PID controller) we use an interruption at each 1ms, which is our sampling time.
- The sampling time of the tachometer is 200ms, it were implemented in the same interruption but with a decision structure to make this in a 1ms time-base.
- The control parameters for the direction controller were set in accordance with our desired characteristics, which were smooth change of direction but with fast response. This were achieved by setting the integral gain to 0.5, the derivative gain to 1 and the proportional one to 1.
- In the velocity controller (first phase of our speed controller) we use the absolute value of the error signal. This is done because for velocity the error cannot be negative. It would not make sense, since we do not care for the direction in this case.
- The PID controller of the tachometer (PID number of turns) is calculated in the main routine because it is slower than the others are. Another way to understand the whole velocity control is that the first control (PID velocity) gives the desired velocity according with the error obtained while the second one (PID number of turns) makes sure that the velocity desired is what we really have in the wheels.