Ian Mackie

William Duong

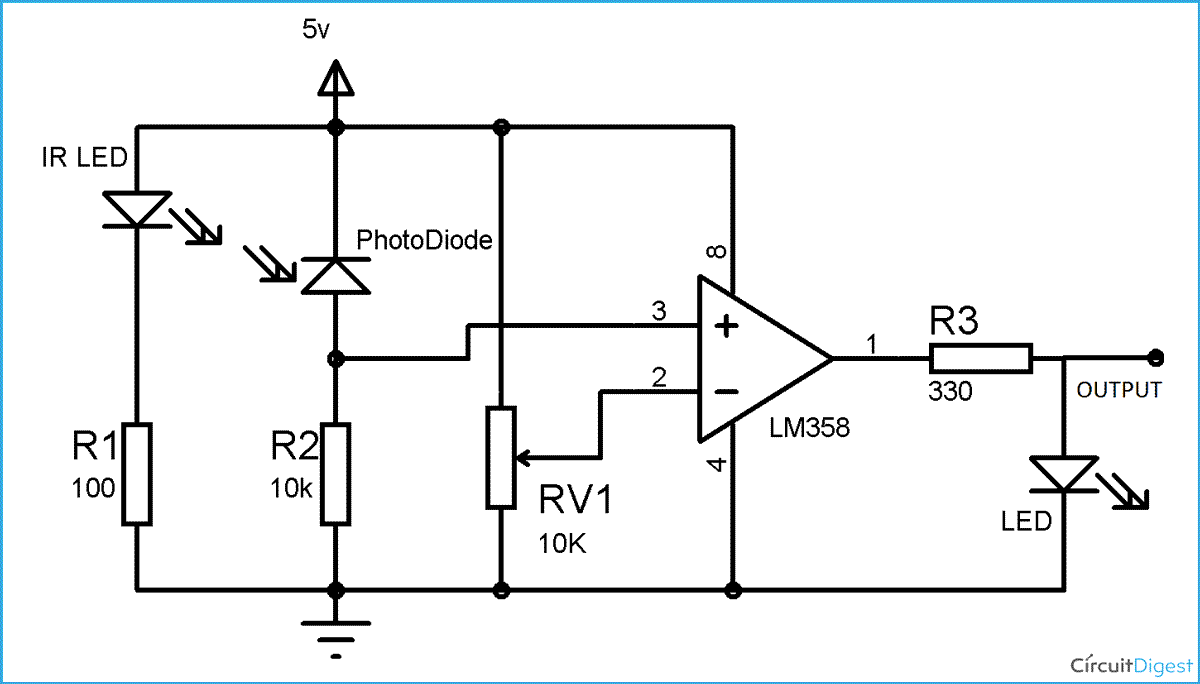
**CECS 311 – Final Project: Analog Line Following Robot**

**OBJECTIVE:** This group will design a basic line following robot guided by using a light dependent resistor (LDR) in conjunction with pulse width modulated (PWM) final drive system. Specifically, we propose that the usage of a proportional-integral-derivative (PID) control loop will provide a smoother and more accurate locomotion system than a traditional ‘stop-go’ line following robot based on the same LDR sensor.

**Design Part I: PID Control System**

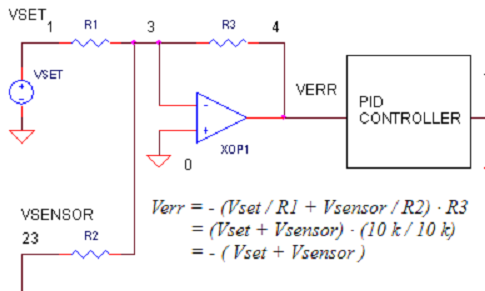
The main design of this robot is dependent on two modules: the PID control system for error correction and the sensor-motor system for final drive output. To consider the first module, we must start with the sensors. Using a cadmium sulfide (CdS) photocell and an infrared LED, we can ‘measure’ the light reflected off our travel path – in this case, either a white, reflective background or a black, non-reflective obstacle – and convert it to an analog voltage through the use of an op-amp (see ‘OUTPUT’ of figure 1).

**FIGURE 1:**



Let us consider the ‘output’ of the sensor circuit to be V(sensor). To use our PID controller, we also need a reference value – which we will call V<set>. This value represents the ideal voltage that our motors operate on – that is, in a straight line, both motors will be driven by this constant voltage. Therefore, any difference between these two values is V(error), a deviance value represented by V(error) = V(set) – V(sensor), and is important because the PID feedback loop operates to reduce this value to zero. See figure 2 for generalized schematic of V(error) prior to the PID circuit.

**FIGURE 2:**



Before examining the PID controller in a block diagram, we should consider a broad outline of its individual elements and their purpose.

The **(P)**roportional term represents the main drive in a control loop and reduces a large part of the overall error. The output value from this term alone is a gain that will tend to generate particularly aggressive correction if sufficiently large. The **(I)**ntegral term reduces final error in a system and does so by summing up total errors over a given period of time. This is helpful is situations where the gain from the **(P)** term is particularly small. This overall summation of smaller errors can produce a sufficiently large enough correction over time. The **(D)**erivative term counteracts the previous two when the output changes quickly and helps instead with overshooting the final correction value. The mathematical derivation of each of these values is outside the scope of this paper, but the sum total corrective output **(CO)** of all terms can be seen in figure 3:

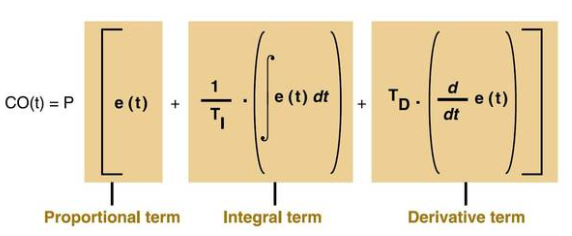
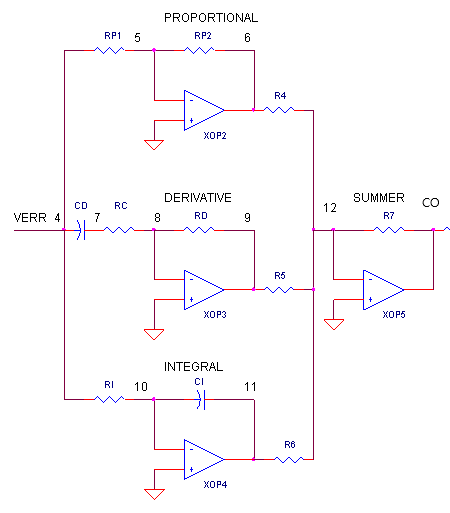
**FIGURE 3:** 

Figure 4 shows a simplified schematic of our proposed PID controller which takes an error signal and passes it through the PID terms prior to final summation into a single corrective output (CO). This output process is a generalized model but the takeaway is that our output (CO) represents the gain we require (in voltage) to power our motors – after taking our error modeling into account.

**FIGURE 4:**



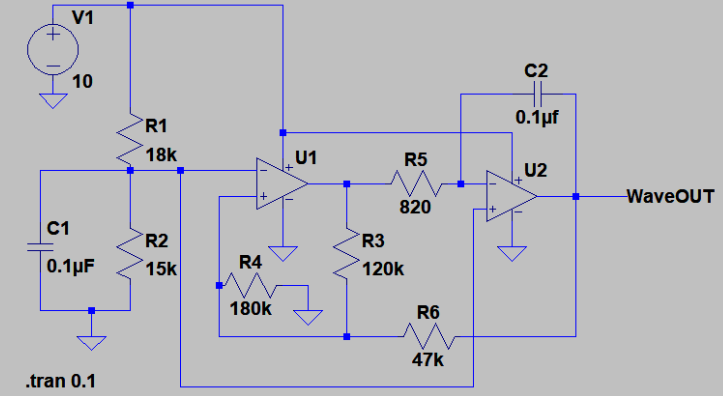
**Design Part 2: Sensor-Motor Final Drive**

The second main module of our design involves the driver powering the two motors on our robot. In a more traditional and simpler setup, turns in one direction or another may involve simply stopping the motor of one wheel while continuing power in the other. While this is a viable solution, the zigzag nature of the navigation is less than elegant.

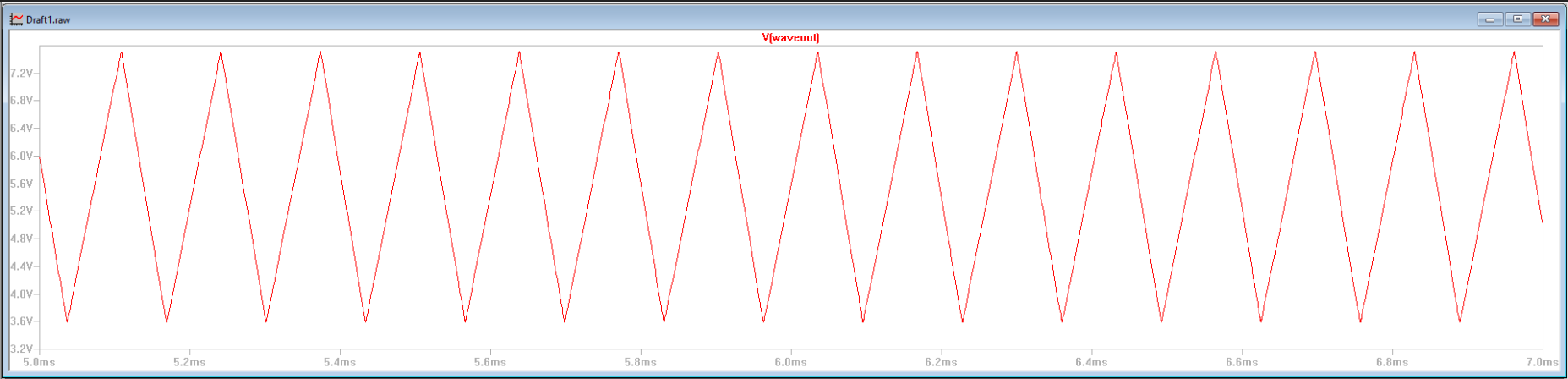
To offer a different solution, we propose that instead of a series of discrete ON/OFF signals being sent to the motor, we utilize a pulse width modulated signal (PWM). Again, the subject and history of PWM exceeds the scope of this paper, but it is sufficient for our purposes to briefly summarize the topic. In essence, PWM is an ON/OFF pulse signal with a constant period or frequency and this is important for defining the concept of a duty cycle. The duty cycle itself is the proportion of time within the total period that is spent ON and, as such, is expressed as a percentage. If we consider this duty cycle to correspond to an average voltage outputting to power the motor, then by varying the PWM duty cycle we can vary the motor speed – without entirely stopping the motor or ramping it to maximum voltage. This is our ideal locomotive goal.

To implement a PWM signal, we must first start with a baseline reference frequency. In our solution, we chose to generate a triangle – see figure 5 for a reference of the circuit diagram and figure 6 for a simulated waveform. Then, by overlaying our PID signal with our generated reference signal, we can produce a PWM voltage signal to serve as the final driver for our motor – see figure 7 for a graphical representation of this objective.

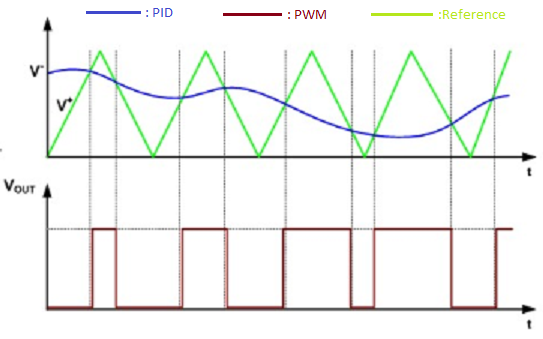
**FIGURE 5:**



**FIGURE 6:**



**FIGURE 7:**

****

**As previously described, we can see in Figure 7 the dynamic between our PID/sensor output and the reference frequency generated by the triangle wave. In short, anytime our sensor detects the black tape/obstacle, its resistance increases and its corresponding voltage output also increases. If we send this signal through a comparator op-amp as (V-) and use the reference triangle wave as (V+), we can see a pulse waveform generated whenever (V+) exceeds the ‘threshold’ voltage of the sensor. To reiterate, as the sensor detects obstacles, the average voltage (i.e. active duty cycle) sent to the motor it controls is lowered and that wheel spins slower than its counterpart. This dynamic accomplishes our goal of turning a car without the effect of ‘stop-go’ that we were trying to avoid.**