

So, first of all, a few problems with the hardware:

-accelerometer is very noisy you should reproject the pcb with a low noise plane

-the sd card, is low speed, if you use fat system the read/write time is +10ms

Software problems:

-the ewma filter is wrong, this is correct

motorCurrentFilt = motorCurrentFilt - (motorCurrentFilt >> 4) + (motorCurrent >> 4);

-same with the accx filter

Milan's note: The original EWMA is not wrong, just takes 12-bit inputs and returns a 16-bit output.

unsigned int averageFilter(unsigned int input)

{

static unsigned int step = 0;

static unsigned int sum = 0;

#define SRATE 6

if(step % (1 << SRATE)) {

sum += input;

} else {

outputFilter = sum >> SRATE;

sum = 0;

step = 0;

}

step++;

```
    return outputFilter;  
}
```

And now a few words of the algorithm

My algorithm is very simple, I use the law of physics $v=d/t$

You can measure speed with an encode placed on the motor

I measure the time with an 1ms/div timer

When the car learn the track, for each point of the track I set a maxim speed based to the acceleration from the accelerometer