Using a DC Motor as a Tachometer

Team 358 - Hauppauge, NY

Here is an out of the ordinary use for a small motor, turning normal motor usage on it's head by running the motor as a weak generator and source of electrical energy rather than as a source of mechanical power and user of electricity. Speed is proportional to the electric current generated by the device. No power is necessary or desired when the motor is hooked up as a tachometer (tach) and basically becomes a generator. You cannot recharge your batteries and it would be silly to attempt to do so, but when the voltage produced is scaled by an adjustable circuit to be within the 0-5 volt range of the FIRST robot controller analog input you have yourself a simple speedometer. Commercial tachometers can be purchased for the job, but tend to be expensive in return for high accuracy and efficiency.
It's main advantages are: instantaneous speed measurement, best suited for high revolution per minute (rpm) measurement, easy to integrate mechanically & as a program analog input, easy to adjust, no extra gearing necessary, inexpensive, no processing overhead, and hardy. Disadvantages over other sensors include being bulkier.

A tachometer…

Tells you how fast something is spinning. It returns a value directly proportional to the rotational speed of a spinning mechanism such as a wheel, motor, or shaft. You see them on some cars telling you the rpm of the engine. A tachometer is not used to count wheel turns, but is a measure of the speed alone and is often used on mechanisms that spin in only one direction.

Tachometer feedback is useful for:

- Setting or limiting the speed of a spinning object
- Monitoring the spin rate for dangerous conditions
- Acting when rpm's reach predetermined values, for example, automatic gear shifting

Tachometers can be made from other sensors, not just DC motors. For example, encoders or light beam sensors counting revolutions, with some processing overhead, can be used to produce a tachometer-like rotary speed measurement. However, a real tach measures instantaneous speed while counting revolutions indicates only average speed over some time interval. Every sensor has unique drawbacks as well as advantages of their own.

Design considerations:

- Judge the suitability of a tach for your application. If you're making sure a wheel doesn't spin too fast a tach is great, however, if you need to measure how many times the wheel has spun then a tach isn't the answer.
- Match maximum tach motor rpm vs. rpm to be measured - chose a motor to match the maximum rated rpm of your rotating object. You want one that generates a healthy voltage at the maximum rpm. You will cut back excess voltage with resistors, but it's difficult to beef up voltage that just isn't there.
- It isn't difficult to match motor characteristics to the minimum/maximum rpm operating range since whatever the voltage resistors will map it into the 0-5v range of the FIRST Robot Controller (RC). Your choice of resistors can emphasize any particular range of speeds that are important to your application.

For tach applications just experiment with spinning motors at the maximum rpm to see what voltage each produces.

- How to physically mount the tachometer and mechanically engaging it with the rotating wheel, shaft, motor, etc. Will the motor you chose be required to take side loads?
- Is a one direction or a bi-directional tach necessary? In either case the tach must be wired to receive positive voltage at the RC. A sample circuit is shown later for a tach that only spins in one direction, but a modified circuit that includes directional diodes is required for converting both positive & negative tach voltage into the RC's positive 0-5v range.
- Tradeoffs such as tachometer noise vs. accuracy / quick response vs software filtering. A DC motor tach doesn't return a pure flat-line value, but one that has tiny fluctuations due to the motor brushes constantly making and breaking contact with the commutator. This might be a consideration if your application requires high accuracy, but typically you'll only notice it on a detailed plot of the tach data..
- You can borrow a commercial tach to calibrate yours if you want to know exactly how each voltage 0-5v corresponds to a specific rpm.

Wiring:

Spin the tach at the maximum rpm you expect to see and measure the voltage produced directly from the motor terminals with a multi-meter. Then it's a simple calculation to figure the resistance required to produce a positive voltage in the range 0 to 5 volts when it turns in the desired direction. You can easily test spin candidate tach motors by attaching them to a variable-speed drill so you can sample the voltage each generates.

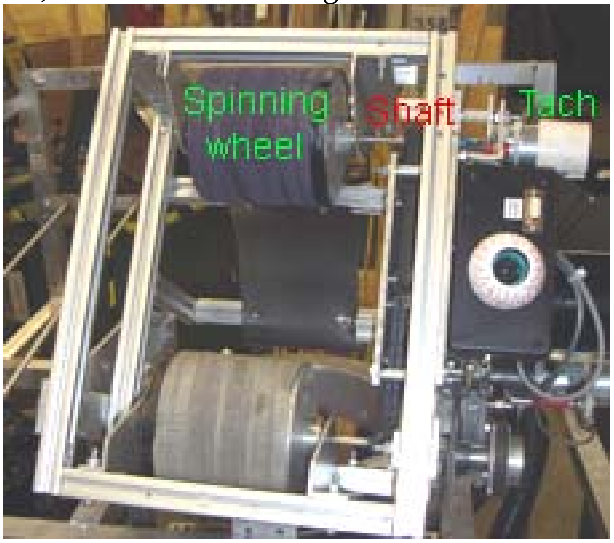
- The voltage generated by the tach during it's full range of motion must be normalized to the RC's range of 0-5v via resistors or an adjustable pot before connecting it to the RC. Calculate the resistance(R) needed using Ohm's Law, V=IR or R=V/I and your own measurement of the current produced.
- If the tach is dedicated to one direction then use a diode to limit current direction and pass positive voltage only for the RC.
- An adjustable potentiometer makes the tach easily tunable if the rpm's of your spinning system are still being changed. If desired the potentiometer can later be swapped for fixed resistors when the mechanical design has settled down and you've found the perfect resistant value.

Software:

No special software is required to use a DC motor tach – one of the reasons a tach can be a simple solution. The tach input comes to the RC via an analog input that your software polls. At it's simplest, poll when you want to know the current speed. The analog input directly corresponds to speed. The resolution of the analog input (1024) means that each analog point will equal some number of rpm. The exact rpm can only be determined by validating with a commercial tach, but it's usually not necessary to convert the analog value into human terms such as rpm. The software will check for and act on pre-determined analog values. Because it's really a motor the tach demonstrates regular minor reading fluctuations as the commutator brushes cross breaks. These can be smoothed or otherwise filtered out or ignored in software.

One Implementation

FIRST team 358, the Robotic Eagles, used this motor-based tachometer feedback to control our robot's shooter wheels in 2006's Aim High ensuring a steady, constant wheel speed of 2900rpm before the launch of each poof ball. The tach can be seen in an early mount on the top right of the following photo of our shooting system driven directly off the shaft of the top shooter wheel. The tach was later relocated to the bottom shooter wheel, but in a similar configuration.

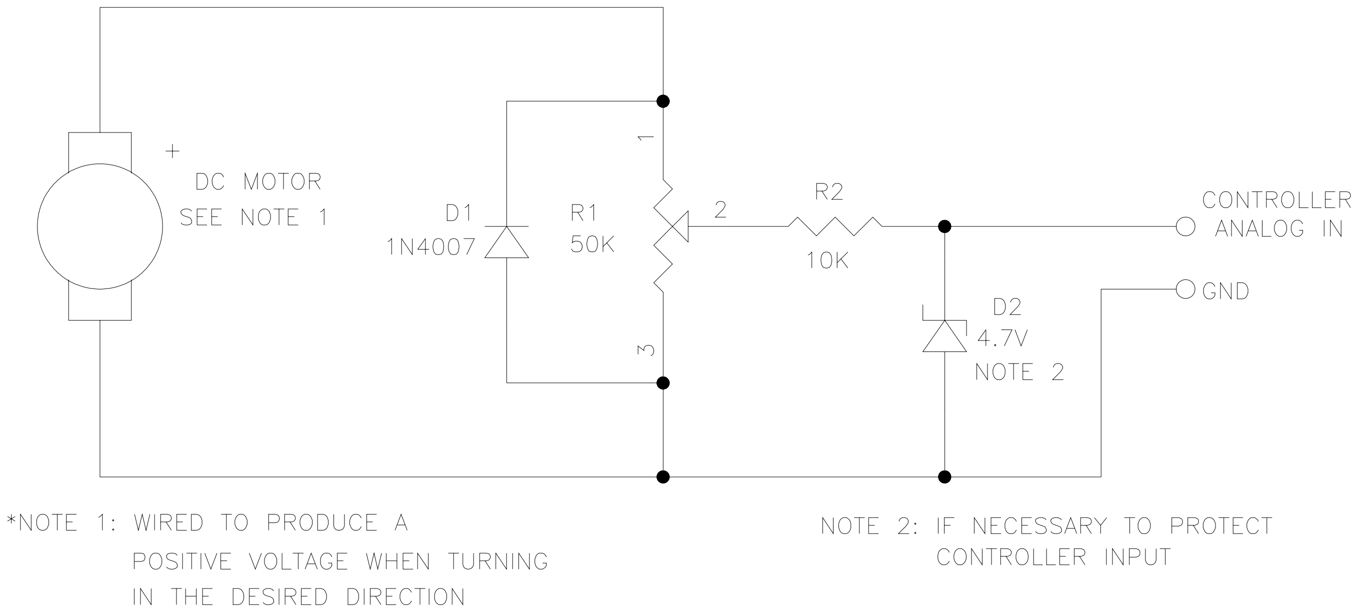


DC Motor:

We used the small Matchubi motor that has come loose in the FIRST Kit-of-Parts for several years. It has an operational maximum of 4900 rpm, perfectly matched to our max. theoretical designed shooter wheel rpm. The shooter wheels were designed to be driven at half speed, so operationally the max rpm was roughly half that, but we designed the tach to operate up to the maximum possible designed speed. In practice with mechanical losses the maximum wheel rpm was actually around 4500.

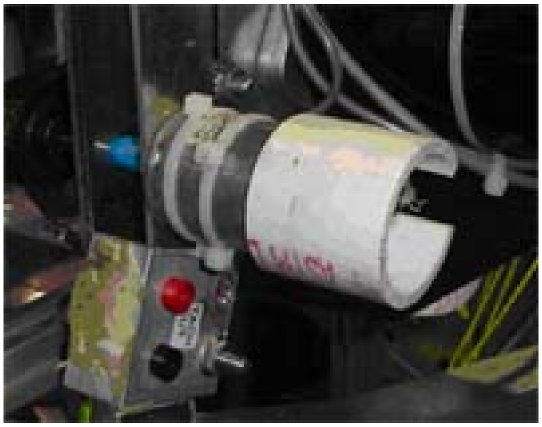
Wiring:

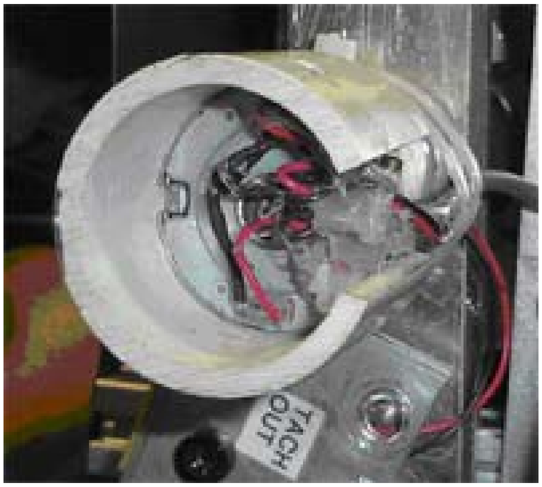
The circuit on the next page is for a one directional tach feeding an analog input on the robot controller. The DC motor connections are wired to produce a positive voltage when it turns in the desired direction. If by accident the tach is run backwards a diode limits current direction and permit positive voltage only to the RC analog input. An adjustable potentiometer limits our maximum voltage to the 5v handled by the RC and makes the resistance easily tunable, but fixed resistors can be used. The end result of this circuit is that in it's final configuration each analog input of 1 represented ~4.76rpm of our shooter wheel.



Mounting:

The mounting was quite simple, one of the primary advantages of using a tachometer. The tach motor was attached to a mounting plate, the pinion passed through a hole in the plate and the pinion was connected directly in line to the wheel drive shaft with a piece of pneumatic tubing. To protect the end of the motor/wiring from risk of physical damage a short piece of PVC, the same diameter as the motor casing, was added over the motor terminals and the circuit at the back of the motor housing. The circuit components were secured by a liberal amount of hot melt glue (making it hard to show the pot and diodes in the photographs). Cooling isn't an issue with the tach motor as there isn't any stall condition and no heat is generated.





Direct tach readout:

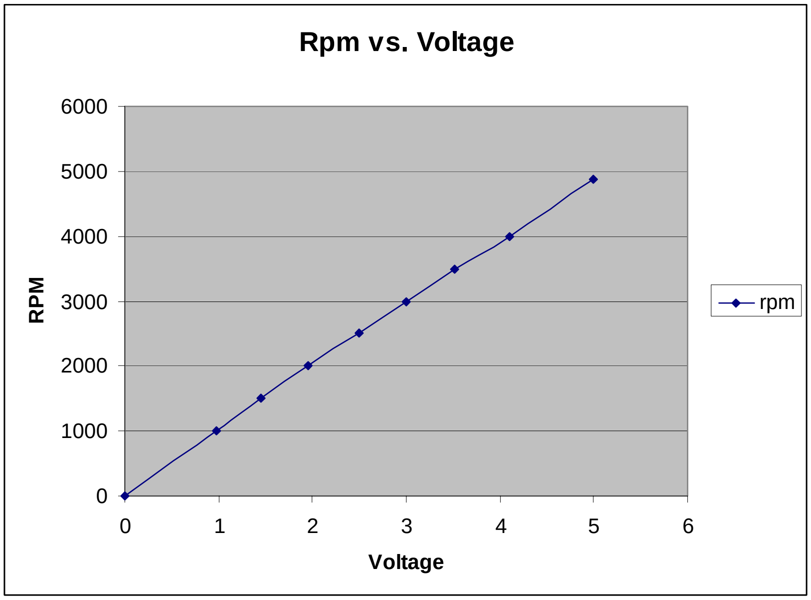
We added an optional permanent electrical connector mount (visible in the right photo above) where we could attach standard multi-meter or oscilloscope probes for a direct readout of the voltage produced by the tachometer while it was operating. These made it simple for the electrical team to fine tune the variable potentiometer, quickly verify correct operation, trouble shoot, and monitor the tach output directly.

Software:

The tach reading comes into the robot controller via an analog input that was simply polled by the software at a convenient rate. A moving average was experimented with, but found to unnecessarily slow response for little gain in accuracy. The tach input was smoothed of minor fluctuations by zeroing the last two bits of the analog input (*Get_Analog_Value(rc_ana_in02) & ~3*), essentially giving us a deadband of 3 analog points which in this case represented an rpm deadband of 14 rpm accurate to half a percent.

The following table are measurements we took of our tachometer validated with a commercial hand-held tachometer. In practical terms, the mapping of rpm to a specific analog input value was unnecessary and was not used. The desired speed of our spinning poof ball throwing wheels was found by repeatedly throwing balls until they consistently hit their target. Then we recorded and hard coded in software the corresponding analog input value to be maintained. The plot of Rpm vs. Voltage shows the relationship is very linear.

Volts	rpm	Analog input value
0	0	0
0.98	1000	200
1.45	1500	297
1.95	2000	399
2.5	2500	512
3	3000	614
3.52	3500	720
4.1	4000	839
5	4878	1023


Rpm vs. Voltage
RPM
Voltage
rpm