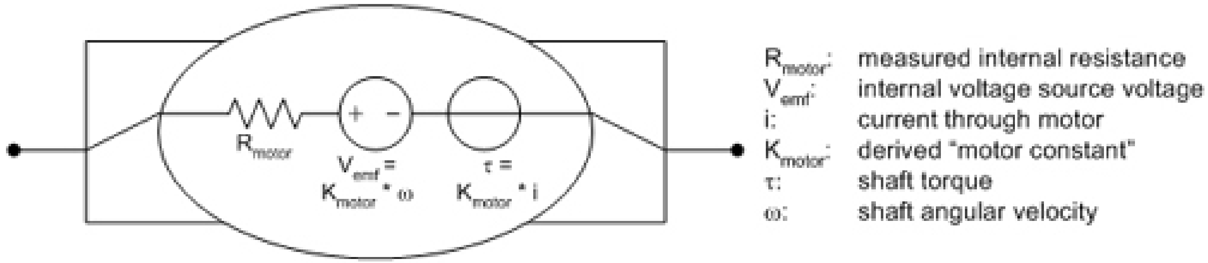
1 Basic E&M of Motors

Motors are both very simple and very versitile devices. There are only two things you need to know to use them. First, the more power you supply a motor with, the faster and more forcefully it will spin. See **Programming Motors** for details on how to supply power with the HandyBoard. Second, motors wires and electrical usage is symmetric. If it spins with voltage applied one way, it will spin the other way if you apply the opposite voltage, switch the wires, or reverse the current.

But motors can be used as input as well as output devices, and both speed and torque can be controlled, with a more sophisticated model.


R_{motor}
+
-
$V_{emf} = K_{motor} * \omega$
$\tau = K_{motor} * i$
R_{motor}: measured internal resistance
V_{emf}: internal voltage source voltage
i: current through motor
K_{motor}: derived "motor constant"
τ: shaft torque
ω: shaft angular velocity

Simple Motor Model

To a first order approximation, motors act like a resister and a variable voltage source. The torque provided by the motor is a function of the current passing through it. The voltage across the motor is a function of the angular speed of the motor's shaft.

See `http://icb.olin.edu/fall_04/ecs/lectures/motor.ppt` for more information.

2 Basic Motor Control

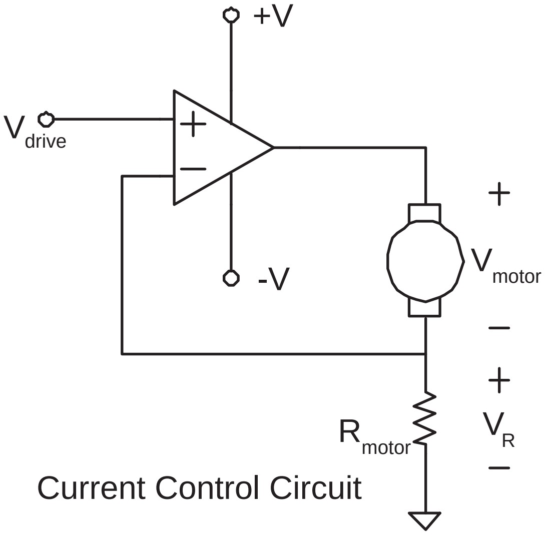
The fundamental "input" to a motor is current. Current through the motor produces a magnetic field, which produces torque, which gets the motor spinning. The spinning wires in a magnetic field produce an electric field– that is, the voltage across the motor.

In practice, voltage can be used to set the speed of a motor. The voltage initially only dissapates across the resister, which produces a current. As the motor spins up and produces its own electric field, the voltage drop across the resister (and corresponding current) will decrease until it produces just enough torque to keep the motor at a constant speed.

This is how the HandyBoard controls motor speed. The L293D (or T1 SN754410NE) motor driver chip converts a low-amperage (TTL) voltage to a proportional voltage up to up to 9.6 V, with up to 600 mA.

3 Advanced Motor Control

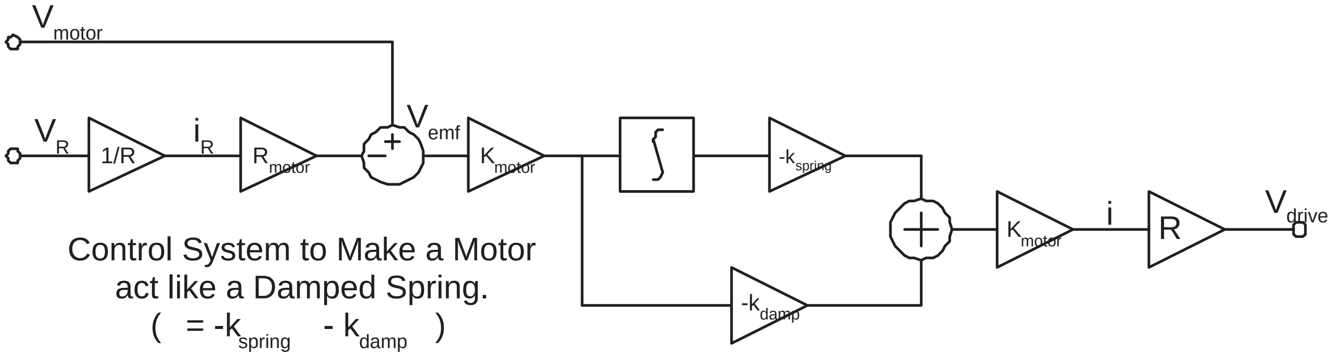
The following circuit can be used to control the torque (current) of a motor directly. The voltage across the resistor can also be read to monitor its speed.



Current Control Circuit

+/-V: a supply voltage, usually 15 V
R_{motor}: motor's internal resistance
V_{drive}: supplied, $V_{drive} = i_{desired} R_{motor}$
V_{motor}: derived, V_{motor} = Vemf + V_R
V_R: derived, $V_R = V_{drive}$

Because motors can act as both input and output devices, if you combine a control system with the electrical input and output of a motor, they can act in more sophisticated ways. Below is a control system which makes a motor act like a watch spring. You can turn it to an angle, and feel its resistance build up, let it go, and it will go into decaying oscillations around a neutral angle.



Control System to Make a Motor act like a Damped Spring.
$(\quad = -k_{spring} \quad - k_{damp} \quad)$

4 Inside Lego Motors

To see inside Lego Motors, go to `http://www.philohome.com/motors/motor.htm`.

The following table, from `http://web.mit.edu/sp.742/www/motor.html`, provides basic information about lego motors. It can also be used to deduce the K_{motor} and R_{motor} (from the model above) of the motor.

5 Motor Varieties

We use four main kinds of motors:

Hobby Motors : By attaching a lego gear and HandyBoard connection pins, we can make normal hobby motors into robot motors. These aren't very reliable, but provide a fair amount of power for their size.

Generic Lego Motors : These simple grey boxes provide very little torque, but are very sturdy.

Geared Lego Motors : These provide an imporessive amount of torque, at a fairly slow speed. Don't try to stop them with your fingers.

Servo Motors : Signals to servo motors make them turn to a particular angle. They don't spin. HandyBoards have some built-in functionality for servo motors. See section 2.7.1 of the manual for more information.

6 Programming Motors

The motor connections are numbered `0` through `3`. `m` below refers to this number.

The motor control functions are:

- `fd(int m)`: supply full power forward
- `bk(int m)`: supply full power backward
- `motor(int m, int p)`: supply the specified power, from `-100` to `100`

- `off(int m)`: supply zero power to the specified motor
- `alloff()`: supply zero power to all motors
- `ao()`: a synonym for `alloff()`.

Typical configurations for car-like robots have separate motors controlling the left and right wheels, or have a set of “drive” wheels and a wheel or wheels that one motor can turn.

7 Setting up Motors

The easiest way to make a lego-ready motor from a normal metal spindle hobby motor is as follows:

1. Center a 8-spoke gear on the spindle of the motor.
2. Fill the space in the middle of the gear with hot glue. Try to not let the hot glue stick to the base of the spindle.
3. If glue did get on the base of the spindle, cut the glue off, or break the bond by spinning the spindle (without breaking the bond between the spindle and the gear).
4. Solder wires to the power pins on the motor. Ideally, get a set of two attatched wires.
5. Solder a the other end of the wires to the first and third pin in a block of pins.
6. Hot glue a thin strips of lego to the top or bottom (a 2x4 works well).

8 Debugging Motors

If a motor stops turning gears, check to make sure that the shaft isn’t spinning in the hot glue that connects it to its gear.

If the motor is sputtering, try mounting a second L293D chip on top of the first to help handle the load (it shouldn’t matter– L293D’s can handle 600 mA of current– but it seems to).