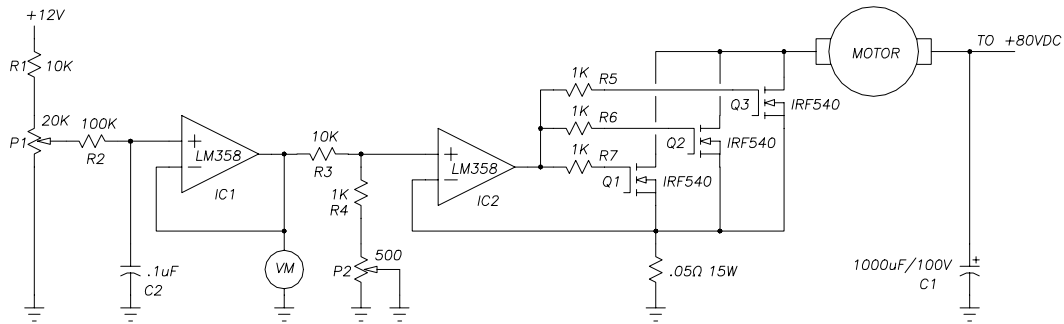# DYNOMOMETER CIRCUIT

This circuit measures motor torque. It depends on a DC motor's linear relationship between armature current and generated torque. The MOTOR in the schematic is used to load the test motor and measure its torque. The MOTOR chosen should have a high torque constant (oz-in / amp). The power supply voltage should drive the MOTOR to at least the same speed as the test motor will run at.

The MOTOR chosen has a torque constant of 50 oz-in, has a NL speed of 3,000 RPM at 80VDC and can generate 800 oz-in of torque.

CIRCUIT DESCRIPTION:

IC2, Q1 thru Q3, R5 thru R7 and the .05 ohm resistor forms a precision 0 to 16 Amp current sink. This results in 0 to 800 oz-in variable torque at the motor shaft since torque is proportional to current.

The desired torque load is set by P1. IC1 buffers this voltage and applies it to the current sink circuit via a voltage divider formed by R3, R4 and P2. The voltages are scaled so that voltmeter VM reads torque as 1 volt per 100 oz-in.

Q1, Q2, and Q3 will get very hot. Heatsink them very well. The .05 ohm 15W resistor may be built up from a parallel combination of resistors if a single resistor can't be found or is too expensive.

A drawback of this circuit is there is no compensation for the MOTOR's frictional torque. This can be compensated for but I don't think the circuit complexity is worth the trouble. Simply measure this torque and add it to the set value. It will amount to a few oz-in anyway.

OPERATION:

Couple the test motor to the load MOTOR. Turn the torque load setting P1 to zero. Run the test motor up to the desired speed. Turn P1 until the motor stalls if it is a stepper or until servo lock is lost if it is a DC servomotor.

ENHANCEMENTS:

In my setup I don't use P1. Instead I drive it with a D to A converter on a Z80 based microcomputer. The Z80 ramps up the step motor to the desired speed, then increments the D to A register until stall is detected from an encoder on the MOTOR. It then records the D to A register, ramps up the step motor to a speed 100 FS higher and repeats the process until I have 100 plot-able values from 0 FS to 10,000 FS per second. These then form the test motor's speed-torque curve. Unfortunately it is all coded in Z80 assembly language and is specific to my hardware. The flow-chart to implement this on a modern computer is pretty simple though.