MC33887DH Monolithic Power H-Bridge IC Provides Muscle for High Torque Robotics

(“Muscles for your Battle ‘Bot, Reflexes for Your Robot Warrior”)

by

Lonne Mays

Systems & Applications Engineer, Motorola APD

Analog Servo IC’s

Analog servo motor control IC’s, such as the venerable MC3030, have facilitated the designer’s motion control task for many years. Integrated circuits of this genre contain the op amps to process the information (i.e., voltage) from analog position sensors, the comparators to process position-command voltages (reference voltages), and the mixed signal circuitry to close the control loop while taking care of drive functions like window detection, drive direction, braking, and stall detection. The drive capability of these IC’s, however, is generally limited to very tiny low voltage motors (e.g., <12V @ <500mA steady-state).

Torque vs. Speed

When limited to such wimpy motors, the issue of obtaining high torque involves a trade-off between the actuator speed and the required torque or force. This is because a high-ratio gear train is required to achieve a torque multiplication factor. As a consequence, the speed of the mechanical actuator is reduced (divided) by the same factor. In essence, it becomes a choice between torque or speed. For many applications a high-ratio gear train is ill-suited, as the mechanical delay imposed between commanding a new position, and that new position being achieved can become greater than what the system requirements will tolerate.

Torque + Speed = MC33887DH

To have *both* high torque and high speed requires a more potent motor be employed. Such motors do not have to be physically large, but they *do* require a heftier VxA product delivered to their terminals. The output of the servo IC can, of course, be boosted via a handful of discrete PNP and NPN bipolar junction power transistors, but the biasing of this external array of parts is a non-trivial task. Care must be taken to provide bias tracking stability over temperature, given the PN junction’s notorious negative temperature coefficient. Additionally, there is the undesireable inherent inefficiency of the multiple Vce(sat) voltage drops in series with the load, as well as the power wasted in the emitter ballast resistors. Creating an external H-Bridge with discrete mosfets is also problematic, as this approach requires differing drive circuits for the high side and low side fets and the use of charge pumps.

An easier means, by far, for boosting the output of a servo controller (whether analog or digital) is a monolithic power H-Bridge ASIC like the MC33887DH. Since this IC incorporates not only the low Rds(on) H-Bridge, but also the gate drive, charge pump, and input logic circuitry, it is a simple and robust solution for boosting the output of a servo controller IC. The MC33887DH allows the system engineer to select motors requiring up to six amps steady-state at up to 30V. This means motors up to 180W may be utilized, thus providing both high torque and high speed via low-ratio gear trains.

Software vs. Hardware

The latest generation of microcontrollers makes it a very easy task to do closed loop servo control, especially when the micro's brains are teamed with the nerves and muscle of power ASIC's like the MC33887, MC33886, MC33486, MC33186, and MC33880. Utilizing the MCU's A/D inputs to read the position sensors, and using the PWM or parallel/serial outputs of the MCU to communicate with the power ASIC, makes servo control more of a programming exercise than a hardware design exercise.

A Design Example for your Battle'Bot or Robot Warrior

Towards the end of keeping this a totally hardware-oriented article, this design example takes a sans-micro approach to tackle the task of converting a standard radio control (RC) pulse width coded signal into a high speed *and* high torque servo response. Refering to Figure 1, High Torque Servo Motion Control Schematic, note that there are five functional areas to the design. In the following paragraphs and figures we will focus on the function and operation of each area delineated on the schematic, beginning with the "SERVO AMP".

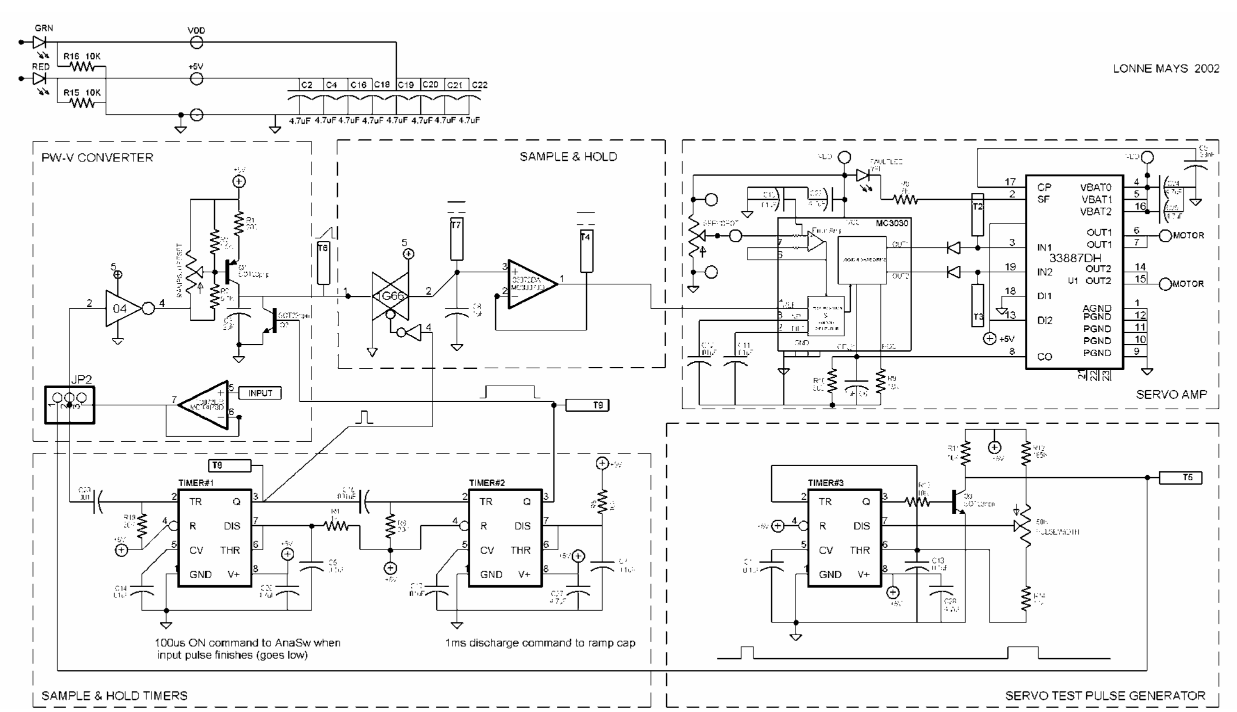

Figure 1. High Torque Servo Motion Control Schematic

Servo Amp

The upper right hand area of the schematic labeled “SERVO AMP” represents the control and power functions we have discussed so far. Note that it contains only two IC’s: the MC3030 servo IC is the “brains”, and the MC33887DH is the “nerves and muscle”. The MC33887DH is the large IC in the center of the printed circuit board shown in Photo 1.

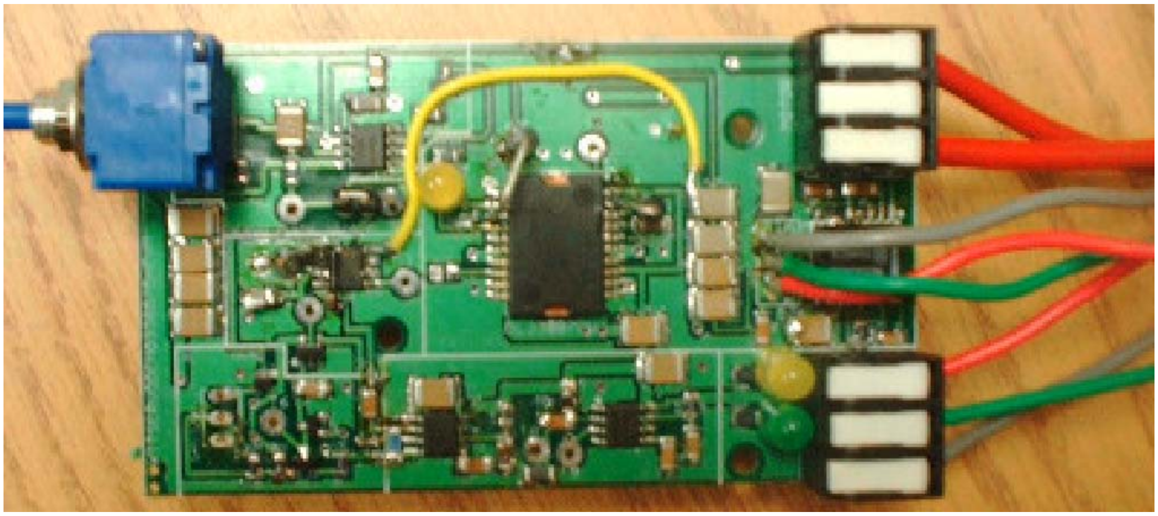

Photo 1. Printed Circuit Board

In the schematic we can see that the two outputs of the MC3030 (which would ordinarily go to a small motor) are interfaced to the two inputs of the MC33887 via two small diodes. Any small signal diodes will do in this case, as their function is only to prevent the MC3030 outputs from overdriving the MC33887’s inputs. (Note: the MC33887 has CMOS/TTL compatible 5V logic inputs with internal 80uA current source pullups.)

The stall detect and over current shutdown feature of the MC3030 has been preserved by utilizing the current feedback output of the MC33887. The MC33887 uses the lossless technique of current-mirroring to sense the motor load current. The technique provides a ratioed sample of the load current (1/375 in this case) which is easily converted into any desired voltage via a single resistor. Applying this resistor to the CDLY input of the MC3030 enables the IC to detect a motor stall/over-current condition and shut off the drive signals. The drive signals will remain shut off until a direction reversal is commanded via the error amp or reference input. The particular stall current threshold is set by the value of the feedback resistor (R10 on the schematic). Capacitor C8 is added to filter out current spikes which may be present due to capacitance in the load. (Don’t forget it’s often necessary to place small capacitors across the motor brushes to reduce EMI/RFI.)

PW-V Converter

The upper left hand area if the schematic shows a circuit I’ve dubbed a “PW-V Converter” (pulse-width-to-voltage-converter). The input to this circuit is the servo pulse supplied from an RC receiver. The width of this pulse corresponds to the servo position commanded from the RC transmitter. E.g., a pulse width of 1.5 ms corresponds to a requested servo position of “center”, a pulse width of 2 ms corresponds to a requested servo position of “fully clockwise”, and a pulse width of 1 ms corresponds to a requested servo position of “fully counterclockwise”. (Or vice-versa, depending on which side of the servo shaft you’re standing.) The output of this circuit is a linear voltage ramp which peaks at a voltage corresponding to the requested servo position. (This voltage will be sampled, held, buffered, and applied to the servo IC as the Reference Voltage. But we’ll get to all that after the next paragraph.) An MC33078 op amp is used to buffer the input pulse, which is then inverted by an MC74HC1G04. Q1, C3, and the associated resistors form a linear ramp circuit which produces a ramp proportional to the width of the input pulse. The slope of the ramp can be varied by adjusting the ratio of R2 to R7, or by the “rampslopeset” potentiometer (use only one or the other in the physical circuit, not both). Photos 2, 3, and 4 show the input pulse and the resulting ramp produced for three different pulse widths.

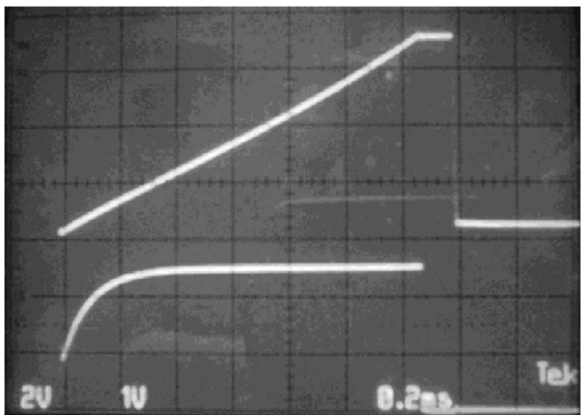

Photo 2. PW-V Conversion “Center Setting”

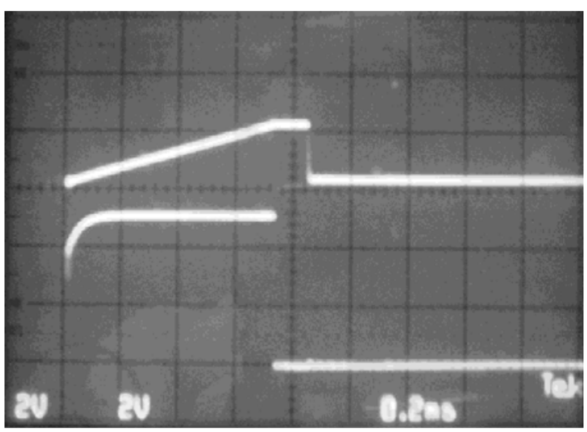

Photo 3. PW-V Conversion “CCW”

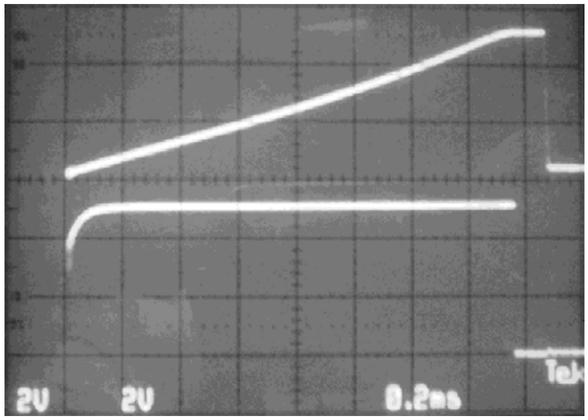

Photo 3. PW-V Conversion “CW”

Sample & Hold Timers

MC1455 timer IC’s were used in this portion of the circuitry (see lower left portion of schematic). Timer#1 is triggered by the falling edge of the input pulse, courtesy of the differentiator that C29 and R19 comprise. This timer generates a 100us command to the Sample & Hold circuit. Photo 5 shows the input pulse and the timer pulse, note that the timer pulse begins on the falling edge of the input pulse. The falling edge of Timer#1’s pulse is fed through a differentiator (C15, R6) to trigger Timer#2. Timer#2 provides a ~1 ms pulse to the base of Q2, causing the rapid discharge of the ramp capacitor, C3. Note how this 1 ms pulse begins at the falling edge of the 100us pulse (see Photo 6).

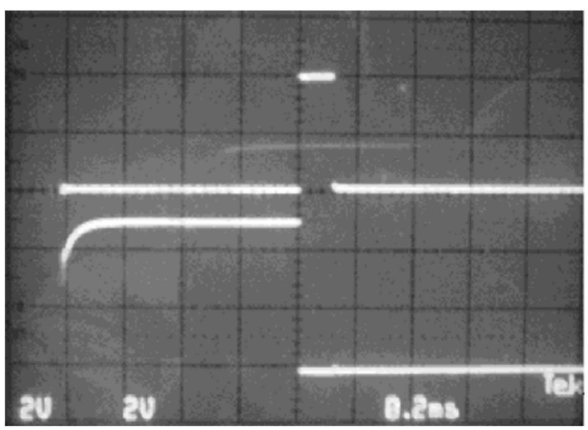

Photo 5. Sample & Hold Timer#1

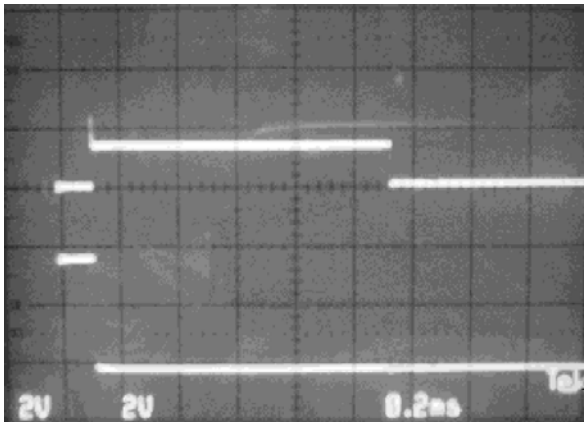

Photo 6. Sample and Hold Timer#2

Sample & Hold

The heart of this little circuit is the MC74VHC1G66 analog switch. Upon receiving the pulse from Timer#1, the analog switch transfers the voltage present on C3 (PW-V Converter) to C8. Recall that Timer#1's pulse occurs at the falling edge of the input pulse, thus the voltage on C3 will have ramped up to a peak corresponding to the width of the input pulse. Recall also that C3 is discharged by Timer#2, which generates its pulse after the completion of the sample & hold activation pulse. Photo 7 shows the output of the sample & hold circuit versus the input pulse. (Note that the input pulse repeats at about 15 ms intervals. The pulse repetition interval is not critical and does not contain any information. It can vary from 10 ms to 30 ms, depending on the brand of RC transmitter.) The voltage stored ("held") on C8 is buffered by another MC33078 op amp (the other half of this dual op-amp chip, actually), and applied to the Position Reference input of the MC3030.

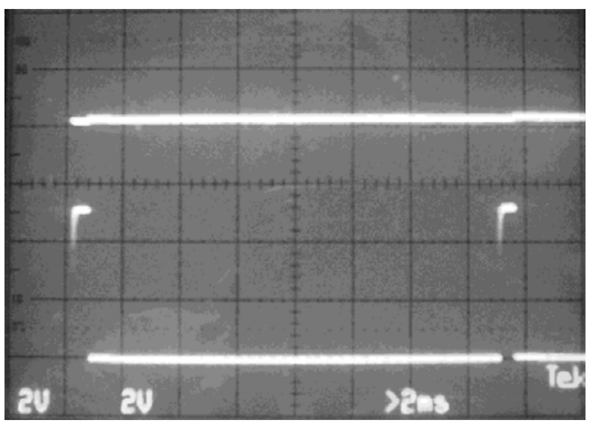

Photo 7. Input Pulse vs. S&H Output

Servo Test Pulse Generator

This circuit (lower right hand portion of the schematic) is simply a test pulse generator added to the board so that it can be tested without having to have an RC transmitter and receiver present. Jumper JP2 is provided so that the PW-V Converter can receive a pulse from an external receiver, or the on-board test pulse generator.

Motor, Gear & Potentionmeter

Photo 8 shows the mechanical assembly used in testing the circuitry. The motor shaft runs a worm gear – worm wheel arangement. The position feedback potentiometer is mounted to the worm wheel and provides 120 degrees of rotation.

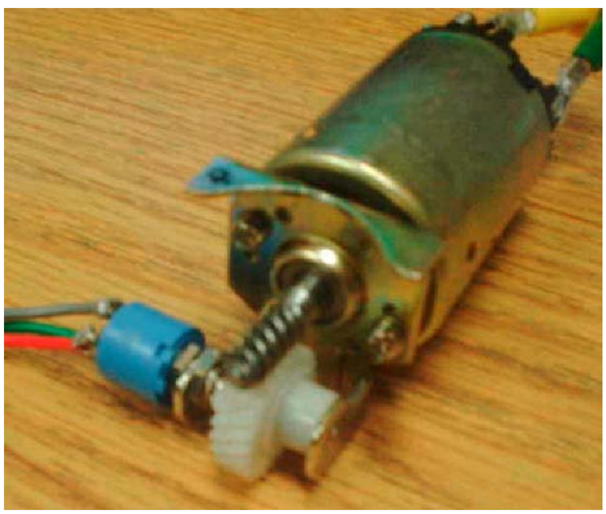

Photo 8. Motor, Gears, and Position Feedback Pot

Summary

I acknowledge that this could be done with one HC08 MCU and one MC33887DH, and about 2K worth of code. But wasn't this more fun (sticking strictly to a hardware solution)?