Firmware and Control System

The control system for the power converter runs on a dsPIC33f microprocessor, which includes 12-bit ADC's, a 16bit fixed point DSP for filtering and PID loops, and a range of peripherals useful in this application such as a motor control PWM generator that generates complementary PWM signals with a programmable dead-time to limit shoot-through, a CAN controller for communications, and an DMA controller that frees the CPU/DSP from spending time doing data transfer to/from the various peripherals.

The firmware for this project is entirely interrupt driven, with nothing happening in "main()". Three asynchronous but periodic interrupts run continuously, and a variety of interrupts occur due to various events.

Periodic Interrupts:

- ADC conversions are initiated at 128kHz, scanning through four channels such that every channel is sampled at 32kHz (with some skew between channels). The DMA is set up to collect 32 samples on each channel before initiating a DMA interrupt. Therefore, at 1kHz a ISR is called and presented with the 32 samples on each channel taken over the past 1 ms. This ISR filters these using a FIR running on the DSP, resulting in a filtered signal at 1kHz. This same ISR then calls the control code which generates control signals (PWM signals for the bridge and load-dump transistor). The processor can run considerably faster than this update rate, but so far this bandwidth has been sufficient.
- A 604Hz interrupt occurs for computing RPM of the generator. This ISR looks at the change in encoder position and computes RPM based upon the interrupt period. The frequency of this interrupt is a tradeoff between maximum rotor speed and resolution, and has been selected to be able to measure speed up to 10000RPM, with a resolution of 1RPM.
- A 25Hz interrupt also occurs, this ISR further filters some of the ADC channels, packages the data into CANbus frames, and initiates the appropriate CAN transfers

Asynchronous Interrupts:

- Encoder position change. An interrupt occurs at every encoder transition, and when the index pulse is seen. If the phases that the PWM signal is applied to depends upon rotor position (such as in motor mode), the PWM setting is overridden for the appropriate phases in this interrupt.
- CAN data available, this interrupt occurs when a CAN packet arrives that has passed the CAN ID mask.

The control loops for battery charging are shown in the attached figures, control of power as a function of RPM is achieved with an inner and outer loop, the outer loop sets a target voltage based on the power error, under the assumption that raising the voltage raises the power. A threshold is applied to this target voltage to constrain it to an allowable range and the amount of thresholding provides an error signal for the load dump control. In this way the load dump is only active when the power control loop is not able to raise the bus voltage high enough to achieve the power target. e.g. If the bus voltage

has been raised to the battery system float voltage, and the batteries are still not drawing enough current, the load dump becomes active. Anti-windup is included in every case where a control signal can saturate. There is potential for interaction between the control loops but in testing so far these interactions have been stable.

Documentation Included:

- Control Loop Diagrams
- Firmware Documentation
- CANbus packet structure