Field Control Coil Current Regulation with Raspberry Pi Pico Microcontrollers

M. Epstein, C. Hansen, C. Paz-Soldan

Columbia Plasma Physics Laboratory

Background and Introduction

- Current in the toroidal and poloidal field control coils on the CUTE reactor is driven using rapidly switching power amplifiers (SPAs)
- Inductive load on these coils is somewhat unpredictable due to effects from interactions between coils and with the plasma itself
- For this reason a feedback loop between the SPA driver and coil current is required to maintain a target current within the coil
- A digital SPA control system is under development for the CUTE reactor, making use of the capabilities of the new Raspberry Pi Pico microcontroller board

Raspberry Pi Pico Architecture

- The Raspberry Pi Pico is a low cost development platform for the RP2040 microcontroller
- Relevant features of the RP2040 include:
  - Dual-core cortex M0+ at up to 133MHz (On-chip PLL allows variable core frequency)
  - 12-bit 500ksps Analogue to Digital Converter (ADC)
  - 2 Programmable IO (PIO) blocks with 8 state machines total that offer user-programmable high speed I/O

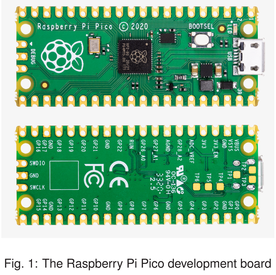


Fig. 1: The Raspberry Pi Pico development board

- Together, these features allow the pico to continuously modulate its output based upon an external analog signal on a 20µs cycle

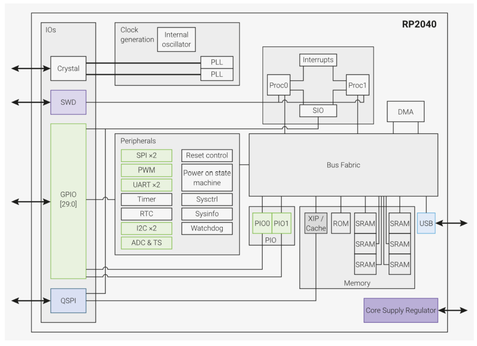


Fig. 2: A system overview of the RP2040 chip courtesy of Raspberry Pi

- The two PIO blocks can be programmed directly to manipulate GPIOs and transfer data with precise timings

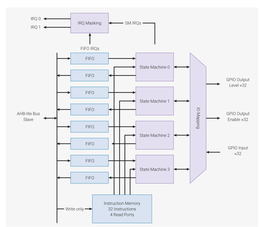


Fig. 3: PIO block level diagram courtesy of Raspberry Pi

- They are highly specialized for I/O, allowing integration with custom high speed fixed function hardware
- In the CUTE system, this allows a custom interface to be created specifically for SPA control, following the strict timing sequences necessitated by the SPA circuit

Current Stabilization Control Outline

- The SPAs in the CUTE reactor deliver power from several large capacitor banks to the field control coils
- Internally, each SPA is in an H-Bridge configuration where:
  - Power can be delivered one way (negative potential) to the coil by closing only switches S1 and S4 or the other way (positive potential) by closing only switches S3 and S2
  - The coil can be brought to an unpowered, "freewheeling" state by closing only switches S2 and S4
  - Closing S1 and S2 or S3 and S4 at the same time is prevented internally as that would short the capacitor bank to ground
- Each SPA is controlled by a single Raspberry Pi Pico through fiber optic signals produced by a driver board attached to the pico

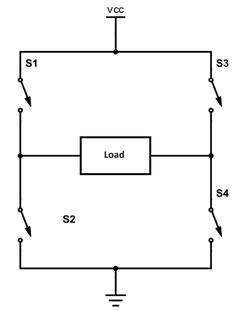


Fig. 4: An H-Bridge Configuration

- The pico outputs an on/off signal on 4 GPIO pins, corresponding to the 4 switches on the SPA H-bridge

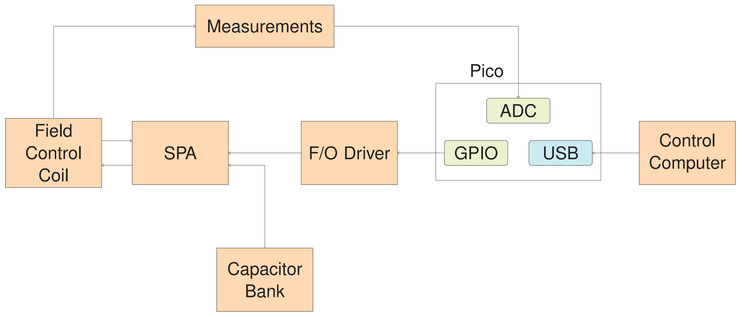


Fig. 5: The Current Stabilization Control Layout

- By alternating switch signals at a high frequency (pulse width modulation), a chosen current can be targeted as a percentage of the maximum capacitor discharge current
- Based upon analog measurements of the coil current, the pico modifies the target current every switching cycle using a proportional, integral, derivative (PID) controller algorithm to correct for coil load instability

SPA Controller Program

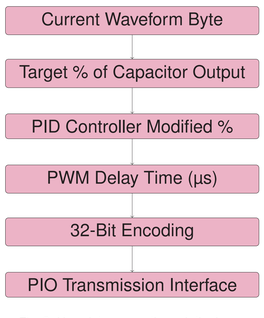


Fig. 6: How data moves through the loop

- The pico recieves the target current waveform as a list of bytes from the control computer
- Every 20µs, a byte is read in order from the list and converted to a positive or negative percentage of the capacitor output
- That percentage is fed to the PID algorythm, the output of which is converted into a delay value from 0 to 20µs
- The delay value along with the switches corresponding to the sign of the target current are encoded into a 32-bit "word" that is passed to the PIO FIFO buffer
- The "word" is then read by a custom PIO program that adds a hard-coded 640ns stagger between switches to further ensure that the capacitor output is not shorted to ground

SPA Controller Program Cont.

- Precise timing is made possible by the separate devices on the RP2040 chip running in parallel
- Every time the PIO completes a cycle (a single PWM "pulse"), it reads the next pulse from its FIFO buffer regardless of whether the value therein has been changed, causing a repeated pulse rather than a missed one if there is an unexpected delay from the PWM module
- Similarly, if the core processor running the PID algorithm is delayed, the PWM block will also default to repeating a target current as to not fall out of sync
- The inverse problem (the processor runs too fast and gets out of sync) is prevented by a flag bit that facilitates communication between the PWM and processor modules

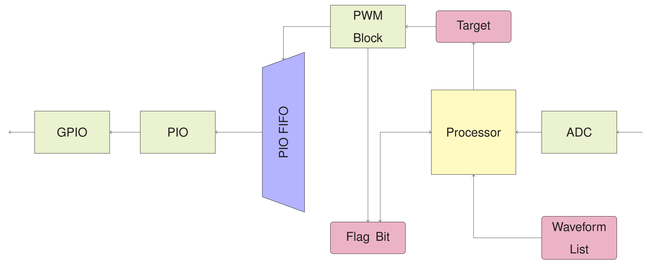
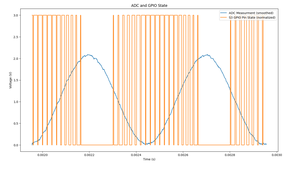


Fig. 7: Simplified SPA Controller Program Main Loop Block Diagram

- As seen below, the SPA controller behaves as expected when coil current measurements are emulated using a function generator and target current is set to 0



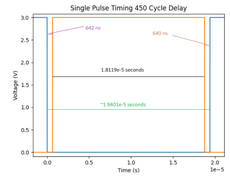


Fig. 8: GPIO Output from External Sine Wave ADC Input (Open Loop)

Fig. 9: S1 and S2 States over one Cycle with Longest Possible Delay

- In this state, because there is no feedback between the PID algorithm and ADC measurement (an open loop), the PWM output trails the input function to the ADC

Future Work

- A trial test setup will be constructed with a simple induction coil and low voltage SPA power supply to ensure that the system works as intended in a closed loop situation
- A communication protocol between the control computer and multiple picos will be implimented to allow driving of multiple SPAs

Acknowledgments

This work was supported in part by the Columbia University Summer@SEAS grant program.