Algorithm: Microcontroller based scanning transfer cavity lock for long-term laser frequency stabilization

S. Subhankar,[a] A. Restelli, Y. Wang, S. L. Rolston, and J. V. Porto
Joint Quantum Institute, National Institute of Standards and Technology and the University of Maryland, College Park, Maryland 20742 USA

(Dated: 27 November 2018)

PACS numbers: Valid PACS appear here

I. STATE MACHINE

We use an event-driven state machine to determine the arrival times of the peaks, t_M, t_S and $t_{M'}$ as shown Fig. 1. The events in the state machine labeled as **1**, **2**, and **3** (see below) are when the state machine and system timer are reset, the acquisition of a peak starts and the acquisition of the peak stops respectively as shown in Fig. 1.

The state machine is implemented using advanced hardware functionalities available in the SAM3x8E microcontroller. This section provides details about the specific hardware peripherals that have been used and the particular implementation of such features in the Cortex-M3 processing units. This particular state machine is designed to respond to a series of interrupts generated by hardware (peripherals) or software (Software Triggered Interrupts) which change the control flow of execution in the program[2,3]. A peripheral is programmed to assert an Interrupt Request (IRQ) when certain conditions are met. Upon assertion of an IRQ, the processor saves the current context of execution and suspends the execution of the current task. The processor then invokes an Interrupt Service Routine (ISR) to service the peripheral. After the ISR is executed, the processor restores the saved context and continues with the execution of the previously suspended task. In Cortex-M microcontrollers (μC) like SAM3x8E, the handling of interrupts is performed by the Nested Vector Interrupt Controller[3,4].

A. Events

In our implementation of the STCL, we employ the following event-centric series of responses.

1 On the rising edge of the Global Digital Trigger (GDT) provided by the cavity driver, the state machine is reset and the control voltages to the slave laser and the Fabry-Perot Cavity that were calculated in the previous cycle are updated. All the contents of the buffers in the μC RAM are reset to 0 and the global timer is reset. It is crucial to keep track of the timestamps of all the events in the state machine. The Cortex-M3 μC has a 24-bit system timer called SysTick timer that decrements from an initial reload value (0xFFFFFF) down to zero every processor clock cycle, which runs at 84 MHz[4]. Upon reaching zero, it loads the reload value on the next clock cycle and counts down again. In our implementation, each cavity scan takes 10 ms which is well within the time it takes for the timer to count down to 0.

2 In order to limit the data acquired for peak determination, the acquisition occurs only near the transmission peaks. To do this, the Analog to Digital Converter (ADC) which samples the peak signal is programmed to assert an IRQ when the value of the incoming peak signal is higher than the High Threshold (HT) of the programmed comparison window. When the peak signal first crosses the High Threshold, the High Threshold IRQ event is serviced by the ADC Handler. The ADC Handler first reprograms the ADC to assert an IRQ only when peak signal is lower than the Low Threshold (LT). It then tags the High Threshold IRQ event with a timestamp and engages the Direct Memory Access (DMA) functionality of the μC to start acquiring the peak signal data.

3 The Low Threshold IRQ event is triggered when the peak signal is lower than the Low Threshold. Upon this IRQ assertion, the ADC Handler reprograms the ADC to now trigger an IRQ only when peak signal is higher than High Threshold, in preparation for the next peak acquisition. It then tags this Low Threshold IRQ event with a timestamp and disengages the DMA which stops the acquisition of the peak signal data. The maximum of the transmission peak is then determined in a peak finder subroutine. Then a software triggered interrupt is invoked that engages the Digital to Analog Converter (DAC) Handler, where the control signals for the Cavity PZT and the Slave Laser PZT are calculated.

[a] Electronic mail: sarthaks@umd.edu.

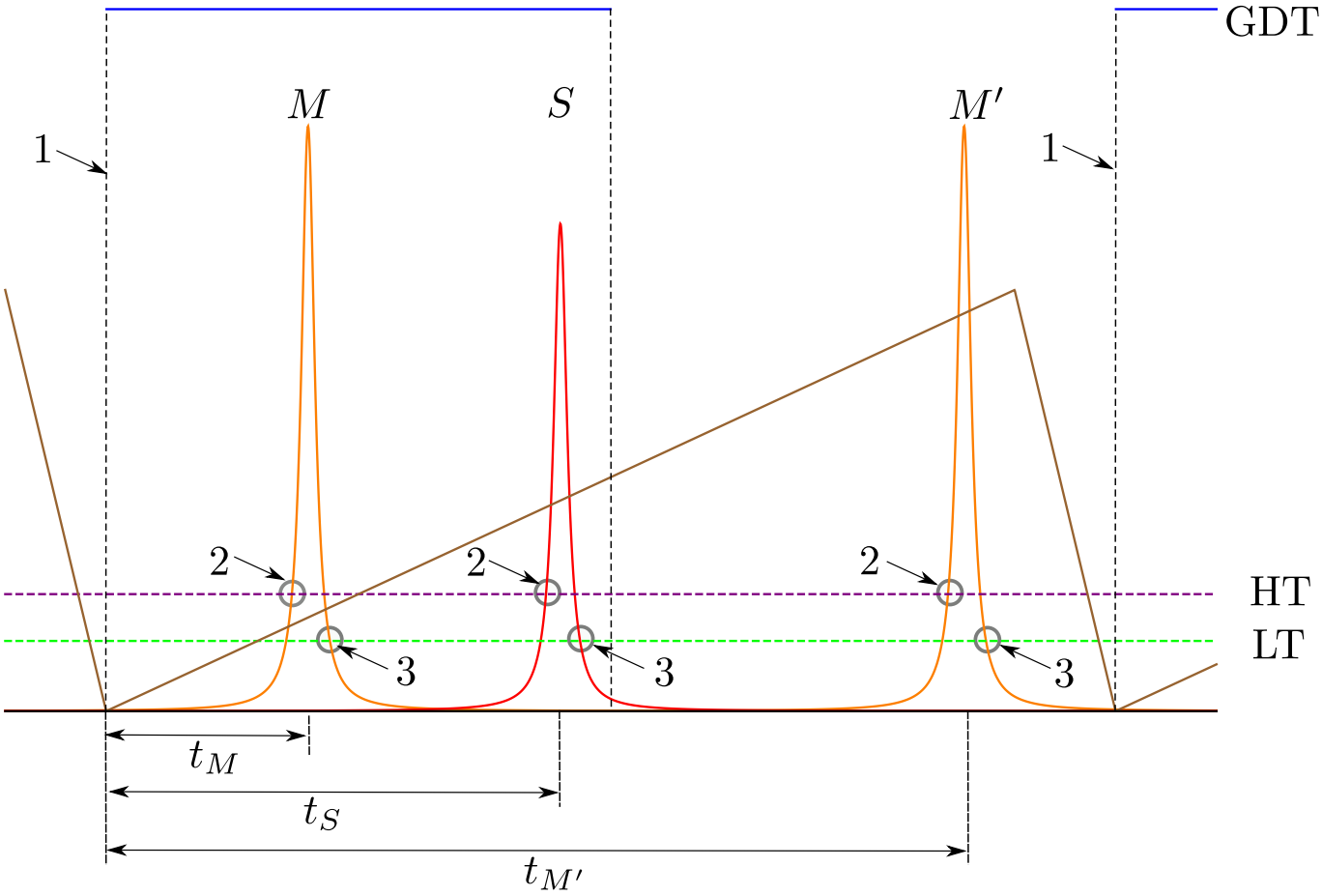

FIG. 1: Timing of the STCL events. The circles pinpoint the time and trigger levels at which events **2** and **3** are triggered. GDT is the Global Digital Trigger, HT is the High Threshold and LT is the Low Threshold. The brown trace represents the cavity PZT scan voltage. (A Photodiode Blanking circuit in the SA201-5B PZT driver[1] ensures that no peaks are measured during the falling section of the PZT scan.)

B. Data Acquisition

In order to ensure fast data acquisition and feedback, we implement the DMA functionality of the 12-bit ADC native to the SAM3x8E μC[3,4]. The ADC samples the transmission peak at 1 Megasamples per second and the DMA functionality transfers the peak signal data directly into a buffer in the μC RAM without any processor intervention. This frees the processor to perform other tasks like performing digital filters on the previously acquired data, while the ADC is engaged in sampling the current peak, thereby parallelizing the process.

In order to optimize RAM usage and reduce the need for data filtering of irrelevant data, we only sample near the peak via signal threshold based interrupts in our state machine. We found that a comparison-window based interrupt (High Threshold-Low Threshold as shown in Fig. 1)[3,4] is superior to a single-valued level-triggered interrupt, since fluctuations in the peak signal near the threshold spuriously triggered interrupts. This is resolved in the comparison-window based interrupt by setting a sufficient difference between High Threshold and Low Threshold. It also helps with easy determination of peak maximums by keeping track of the timestamps of the ADC IRQs. We typically sample 125 points for each peak.

C. Peak Detection Algorithm

The position of the maximum of the transmission peak is determined using a digital Savitzky-Golay (SG) filter[5,6]. This filter is an efficient method to smooth the acquired peak signal data without significantly distorting the signal while improving the SNR. To find the maximum of the peak, the acquired peak signal data array is convolved with a 5-point 1st derivative SG filter and the position of the zero-crossing is found. The program then calculates the time at which the maximum of the peak was sampled by linearly interpolating the time of the zero-crossing between the timestamps taken at the High Threshold IRQ (Event **2**) and Low Threshold IRQ

events (Event **3**).

The 5-point SG filter has the following form

$$Y'_j = \frac{-2y_{j-2} - y_{j-1} + y_{j+1} + 2y_{j+2}}{10} \quad (1)$$

where y_j = value of the buffer at index j and Y'_j = derivative at buffer position j. The 5-point filter can be efficiently implemented using shift operators in the program. Using the zero-crossing timestamps to tag the peaks makes the STCL robust to laser power fluctuations.

Upon finding a zero-crossing timestamp (t_M, t_S or $t_{M'}$) a software triggered IRQ is asserted for the DAC which invokes the DAC Handler that calculates the error signals and control voltages for the servo loop.

D. Servo Loop

The servo loop feeds back on the voltage to the PZT of the external cavity diode laser (ECDL) to stabilize the slave laser frequency and to the Fabry-Perot cavity PZT to stabilize the cavity length.

Inside the DAC Handler, the zero-crossing timestamps t_M, t_S and $t_{M'}$ are used to compute the error signal for the Cavity PZT and Slave Laser PZT, which are $t_{M,\,\mathrm{lock}} - t_M$ and $r_{\mathrm{lock}} - \Delta t_{MS}/\Delta t_{MM'}$ respectively. The control signal ($u(t_y)$) to the PZT at discrete time t_y is given by the velocity algorithm[7]

$$u(t_y) = u(t_{y-1}) + K_{\mathrm{P}}(e(t_y) - e(t_{y-1})) + K_{\mathrm{I}} e(t_y)\Delta t \quad (2)$$

where $e(t_y)$ is the error signal at discrete time t_y, K_{P} is the proportional gain, K_{I} is the integral gain, Δt is the time it takes to scan the cavity, and $t_y = y\Delta t$ where n is an integer. Updates to the PZT voltage are performed at the rising edge of the next Global Digital Trigger using two 12-bit DACs native to the SAM3x8E μC. Anti-windup is implemented by not updating the value of $u(t_y)$ if $u(t_y) - u(t_{y-1})$ causes the DAC output to fall outside an adjustable range of voltages.

[1] The identification of commercial products is for information only and does not imply recommendation or endorsement by the Joint Quantum Institute or the National Institute of Standards and Technology..

[2] E. White, *Making Embedded Systems: Design Patterns for Great Software*, Vol. 2011 (2011).

[3] J. Yiu, *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, 3rd Edition* (2013).

[4] *SAM3X / SAM3A Series Atmel — SMART ARM-based MCU* (2015).

[5] M. Howard and J. J. Workman, Spectroscopy **19**, 44 (2004).

[6] R. W. Schafer, IEEE Signal Processing Magazine **28**, 111 (2011).

[7] K. J. Astrom, Control Systems, IEEE **26**, 216 (2006).