Locking cavities in the digital domain

Sean Leavey

Institute for Gravitational Research

October 9, 2015

How cavities are locked

Typically with analogue electronics, using various techniques:

Pound-Drever-Hall locking

Beat between reflected cavity light and RF sidebands

Dither-and-lock

Amplitude modulation using intentionally applied audio signal

Tilt locking

Beat between misalignment modes and carrier

Analogue control

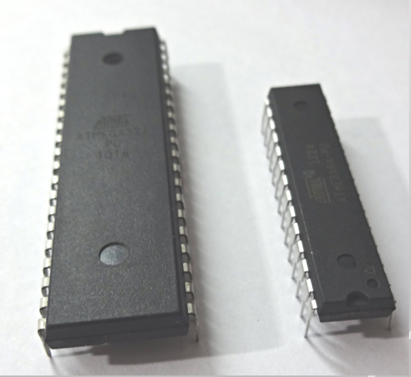
These techniques typically involve relatively simple analogue electronics, for various reasons:

- Fast
- Stable, linear components
- Wide bandwidth
- Automatically “real time” (c.f. computer interrupts)
- It’s what we know

But can digital electronics perform as well as traditional analogue electronics?

Microcontrollers

- Available in all shapes and sizes
- Programmable in C, Matlab, LabVIEW...
- Low cost



Microcontrollers

Q: *But microcontrollers can't perform RF demodulation!*
A: True, but could they be "good enough" for certain applications?

REVIEW OF SCIENTIFIC INSTRUMENTS **85**, 123112 (2014)

Microcontroller-based locking in optics experiments

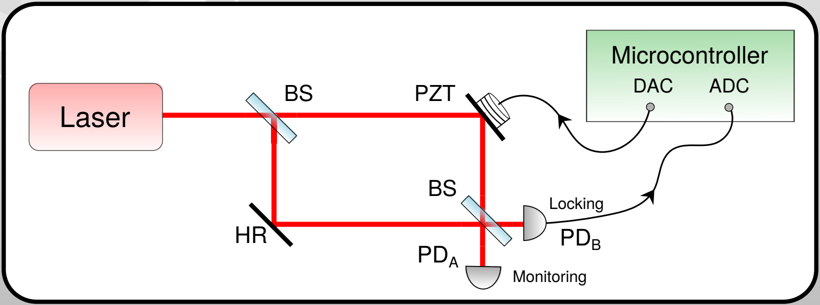
K. Huang,[1,2] H. Le Jeannic,[1] J. Ruaudel,[1] O. Morin,[1] and J. Laurat[1,a)]
[1]*Laboratoire Kastler Brossel, UPMC-Sorbonne Universités, CNRS, ENS-PSL Research University, Collège de France, 4 place Jussieu, 75005 Paris, France*
[2]*State Key Laboratory of Precision Spectroscopy, East China Normal University, Shanghai 200062, China*

(Received 12 September 2014; accepted 29 November 2014; published online 18 December 2014)

Optics experiments critically require the stable and accurate locking of relative phases between light beams or the stabilization of Fabry-Perot cavity lengths. Here, we present a simple and inexpensive technique based on a stand-alone microcontroller unit to perform such tasks. Easily programmed in C language, this reconf gurable digital locking system also enables automatic relocking and sequential functioning. Different algorithms are detailed and applied to fringe locking and to low- and high-f nesse optical cavity stabilization, without the need of external modulations or error signals. This technique can readily replace a number of analog locking systems advantageously in a variety of optical experiments. © *2014 AIP Publishing LLC*. [http://dx.doi.org/10.1063/1.4903869]

Mach-Zehnder

Consider a table-top Mach-Zehnder interferometer



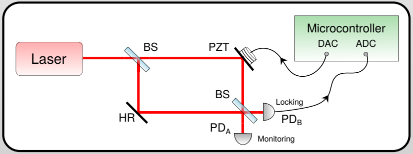
Searching algorithm is implemented in the microcontroller to lock the interferometer.

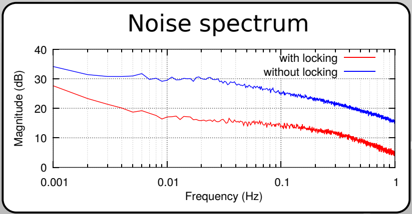
Algorithm

In a loop, run:

1. Set $X_n = X_{n-1} + k\Delta X$
2. If $Y_n < Y_{n-1}$, set $k = -k$
3. Set $Y_{n-1} = Y_n$

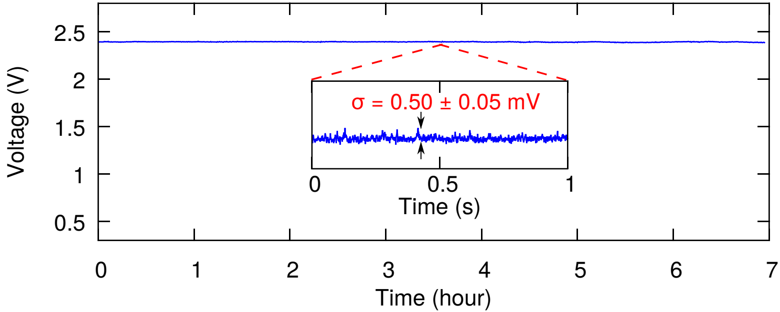
where X_n is the controller output and Y_n is the controller input at step n, k is a flag (1 or -1) and ΔX is the step size to use on the PZT.





Advantage: Re-centring of range

With a microcontroller, the actuator range can be re-centred before it reaches its limit. In the Mach-Zehnder experiment, this was required roughly once per hour, and allowed it to be locked for 7 hours:

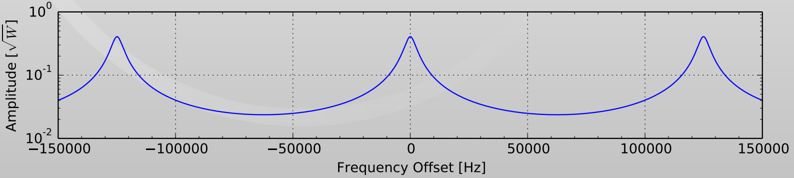


Optical cavities

Locking a Mach-Zehnder is all very well, but does it work for optical cavities?

Optical cavities

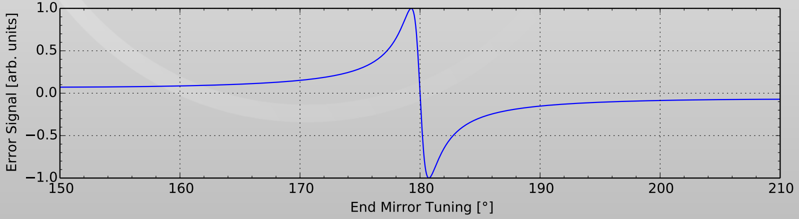
In an optical cavity we want to sit at the top of the resonant peak. This can be found by searching for the maximum signal magnitude, and somehow controlling the cavity to stay there.



Optical cavities

Imagine the cavity starts to move away from the resonant peak. **How do we know which way it's moving in order to correct it?**

In our field we typically use the famous Pound-Drever-Hall signal:



Digital bipolar error signal

In the microcontroller, it is possible to differentiate the DC signal in order to obtain information about the cavity's direction of travel (and thus feedback a compensating impulse):

Finite difference method

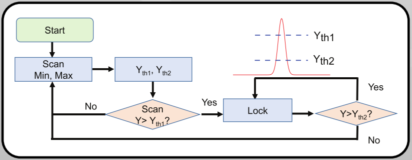
$$f'(n) = \lim_{\Delta t \to 0} \frac{f(n) - f(n-1)}{\Delta t}$$

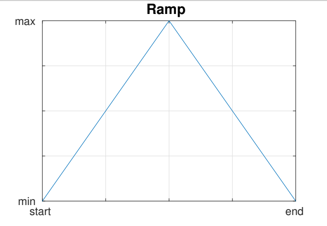
Finesse 100 cavity

Let's try a 4 cm cavity with a non-linear crystal of finesse 100.

The algorithm needs to be modified from the Mach-Zehnder case with a simple ramp routine to scan for the peak signal:

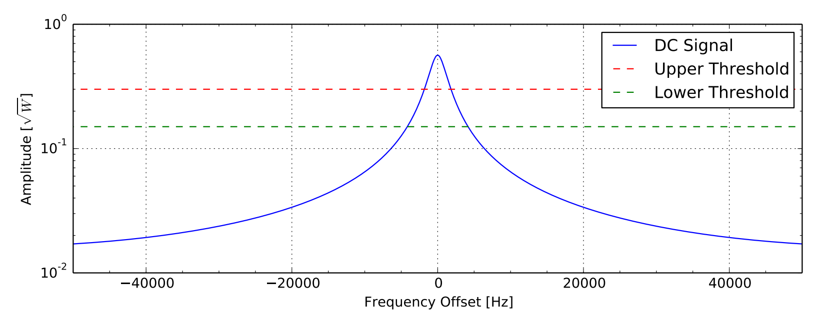
1. **Scan through range to find maximum signal**



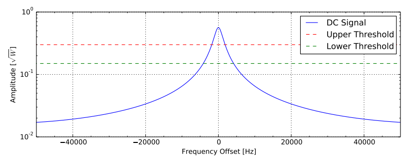


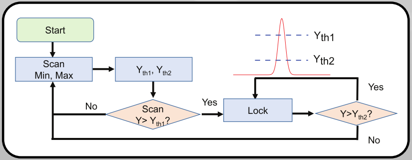
Finesse 100 cavity

1. Scan through range to find maximum signal
2. **Set Y_{th1} and Y_{th2}**



Finesse 100 cavity





1. Scan through range to find minimum and maximum signals
2. Set Y_{th1} and Y_{th2}
3. **Scan again. If $Y_n > Y_{th1}$, enter same locking loop as Mach-Zehnder; else start again at (1). During loop, if $Y_n < Y_{th2}$, start again at (1).**

Finesse 100 cavity

The locking algorithm from before is still essentially the same (just with the addition of a peak finding routine):

Algorithm

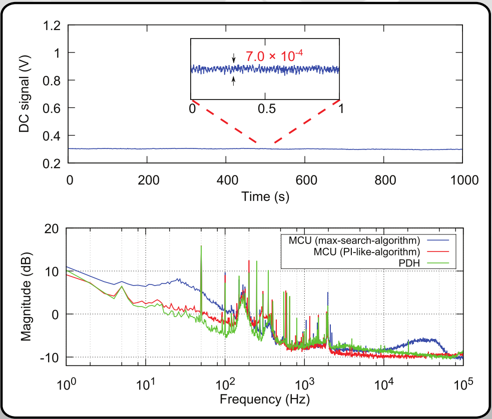
In a loop, run:

1. Set $X_n = X_{n-1} + k\Delta X$
2. If $Y_n < Y_{n-1}$, set $k = -k$
3. Set $Y_{n-1} = Y_n$

where X_n is the controller output and Y_n is the controller input at step n, k is a flag (1 or -1) and ΔX is the step size to use on the PZT.

What's the phase stability like with these step sizes?

Finesse 100 cavity

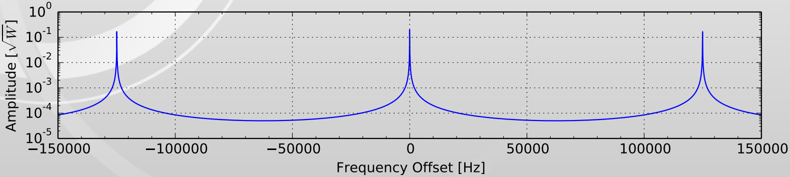


Cavity locked for 1000 s with similar phase stability to PDH on the same cavity ($= 6.6 \times 10^{-4}$ V).

Control matches that of PDH at frequencies above 200 Hz when it is given some gain and integration filters.

Finesse 1000 cavity

What about a finesse 1000 cavity?

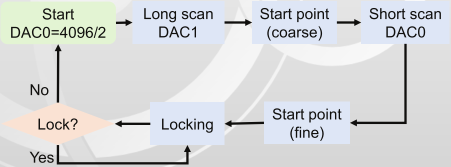


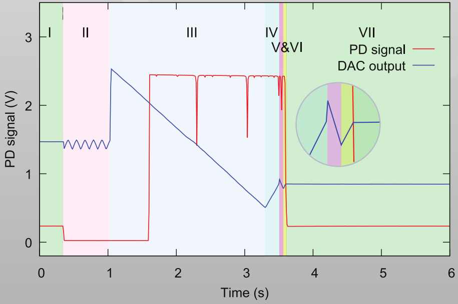
The DACs on typical microcontrollers have 12 bits, meaning there are $2^{12} = 4096$ possible levels to cover the whole control range.

That means only 4 points cover the peak.

To overcome this problem, the authors combined two 12-bit ADCs to obtain the effective resolution of one 24-bit ADC.

Finesse 1000 cavity





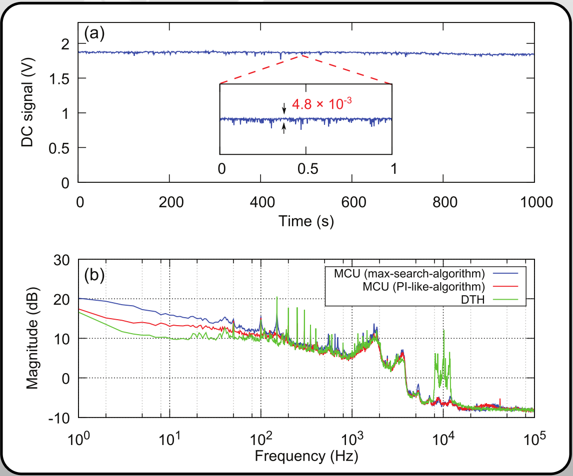
The idea is similar to the finesse 100 cavity controller.

The peak is first found with a "rough" scan using the first DAC.

Once this peak is found, the second DAC performs a fine scan across the peak to find the maximum.

Finesse 1000 cavity

Performance matches a dither-and-lock system



But...

These results are dominated by technical noise!

- Seismic noise in the table-top experiment surely dominates here
- If seismic were suppressed then frequency noise would then dominate
- Traditional PDH locking avoids frequency noise by going to RF **so it would still outperform other techniques** for us

Other downsides from our point of view:

- Microcontrollers probably couldn't run our complicated filter models
- Timing is non-deterministic

Can we, the people that care about noise in the audio band, get the best of both worlds: RF locking, but in a real-time digital servo?

FPGAs

Sparkes *et al.* implemented Pound-Drever-Hall locking in the digital domain with Field-Programmable Gate Arrays (FPGAs):

REVIEW OF SCIENTIFIC INSTRUMENTS **82**, 075113 (2011)

A scalable, self-analyzing digital locking system for use on quantum optics experiments

B. M. Sparkes, H. M. Chrzanowski, D. P. Parrain, B. C. Buchler, P. K. Lam, and T. Symul
Centre for Quantum Computation and Communication Technology, Department of Quantum Science, Research School of Physics and Engineering, The Australian National University, Canberra, ACT 0200, Australia

(Received 19 May 2011; accepted 20 June 2011; published online 27 July 2011)

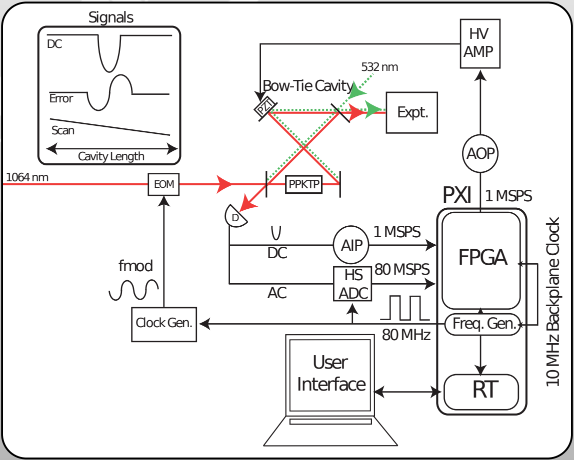
Digital control of optics experiments has many advantages over analog control systems, specif cally in terms of the scalability, cost, f exibility, and the integration of system information into one location. We present a digital control system, freely available for download online, specif cally designed for quantum optics experiments that allows for automatic and sequential re-locking of optical components. We show how the inbuilt locking analysis tools, including a white-noise network analyzer, can be used to help optimize individual locks, and verify the long term stability of the digital system. Finally, we present an example of the benef ts of digital locking for quantum optics by applying the code to a specif c experiment used to characterize optical Schrödinger cat states. © *2011 American Institute of Physics*. [doi:10.1063/1.3610455]

What are FPGAs?

- Programmable circuits
- Typically much faster than standard CPUs at certain tasks
- Varying levels of abstraction: C programmable, LabVIEW interface, etc.
- Fun fact: once used for bitcoin mining (now replaced by ASICs)

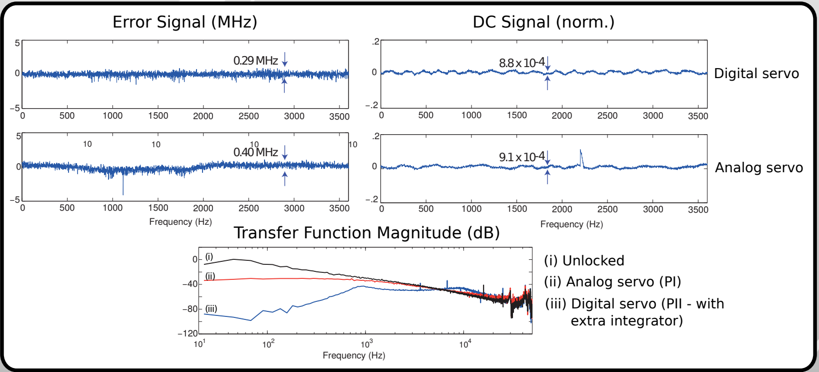
FPGAs for quantum optics

Generation of EOM sidebands for quantum optics experiments:



FPGAs for quantum optics

Stability over 1 hour



FPGA ADC noise

Offers an ENOB of 12.8. For reference, CDS has 13.1.

AD9460

AC SPECIFICATIONS

AVDD1 = 3.3 V, AVDD2 = 5.0 V, DRVDD = 3.3 V, LVDS mode, specified minimum sample rate, 3.4 V p-p differential input, internal trimmed reference (1.7 V mode), A_{IN} = −1.0 dBFS, DCS = AGND (on), SFDR = AGND, unless otherwise noted.

Table 2.

		AD9460BSVZ-80			AD9460BSVZ-105			
Parameter	**Temp**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
SIGNAL-TO-NOISE RATIO (SNR)								
f_{IN} = 10 MHz	25°C	77.6	78.4		77.2	78.1		dB
	Full	77.4			76.9			
f_{IN} = 170 MHz	25°C	76.1	76.8		75.0	76.2		dB
	Full	75.0			74.5			
f_{IN} = 225 MHz	25°C		75.7			75.2		dB
SIGNAL-TO-NOISE AND DISTORTION (SINAD)								
f_{IN} = 10 MHz	25°C	76.1	78.0		75.2	77.4		dB
	Full	74.4			74.5			
f_{IN} = 170 MHz	25°C	74.0	76.1		72.0	75.1		dB
	Full	72.1			71.2			
f_{IN} = 225 MHz	25°C		74.6			73.6		dB
EFFECTIVE NUMBER OF BITS (ENOB)								
f_{IN} = 10 MHz	25°C		12.8			12.7		bits
f_{IN} = 170 MHz	25°C		12.5			12.4		bits
f = 225 MHz	25°C		12.3			12.1		bits

Cost

One benefit of microcontrollers is the reduced cost. FPGAs that work for RF are expensive, so why move away from analogue electronics?

But there are potential benefits to the use of FPGAs:

- Adjustable modulation frequency
- Changing phase of local oscillator as you sweep through resonance
- Re-centring of actuator ranges
- CDS-like capabilities (on-the-fly reprogramming, transfer functions, etc.)
- Possibly similar noise performance to CDS (I'm not certain)

Conclusion

- Microcontrollers seem pretty convenient for table-top experiments
- Cheap and easily (re)programmable
- Ideal for student projects and quick optical tests without expensive equipment
- Definitely not good enough for the speedmeter's main DOF

Possible discussion:

- Could microcontrollers be useful for auxiliary DOFs in suspended experiments (e.g. mode cleaning with tilt-locking)?
- Could FPGAs offer similar or better performance to analogue RF electronics?
- Could we implement re-centring of actuator ranges in CDS?