Vibration Suppression R&D at University of British Columbia

Tom Mattison
University of British Columbia

Joint
T6 (Environmental Control)
T1 (Interaction Regions)
M3 (Linear Colliders)

Snowmass, Colorado
14 July 2001

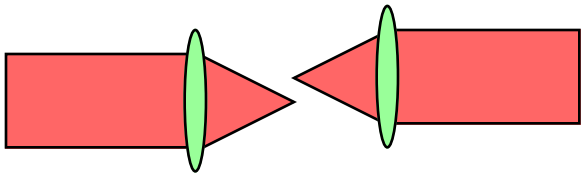
Overview

- Description of the problem and proposed solution
- Our R&D program at UBC
- Hardware
- Some results from work in progress

The Final Quad Stability Problem

Linear collider luminosity requires nanometer beam sizes in one dimension, conventionally the vertical.

Transverse motions of the final quadrupole move the focal point essentially nanometer to nanometer.



Even at a good site, e.g. Aurora IL dolomite mine, ground motion exceeds this at frequencies below about 10 Hz.

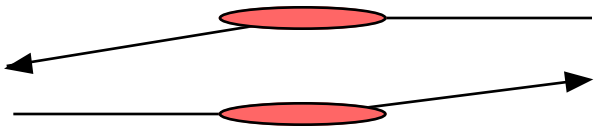
The IP quads will be about 10 meters apart, and natural ground motion correlations are large at low frequencies.

Rigid supports to bedrock for both quads would probably be good enough.

Final quadrupoles always end up inside the physics detector, usually on floppy cantilever supports, which amplify any ground motion. The detector, and even the quads themselves, typically have vibration sources (coolant flows, fans, etc).

Relief from Beam-Beam Feedback

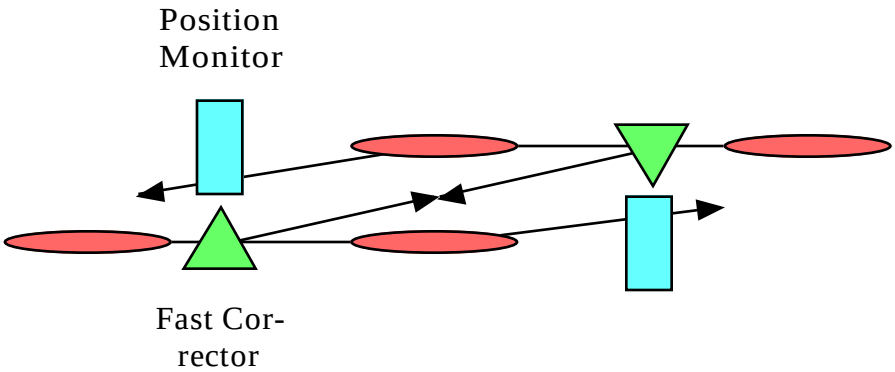
Beam-beam deflections can be used to steer the beams into collision, so the absolute alignment need not be good to nanometers.



Beam-beam-deflection feedback can keep the beams in collision, but the data rate is limited by the bunch rate

With low-latency feedback, deflection of the first bunches can be used to steer later bunches in the train

"easy" for TESLA, plausibly helpful for X-band



Even with high latency, feedback works well for frequencies much lower than the macro pulse rate.

but is it a factor of 2 lower, or 6, or 20 ?

Choices for Final Quad Position Feedback

Reference point
- “Bedrock:” fiducial outside the detector
- “Fixed stars:” inertial frame (no DC feedback)

Quadrupole Position Sensor
- Interferometer
  - measures position directly at long range
- Geophone
  - inertial detector above internal resonance
  - signal proportional to velocity
  - not compatible with detector solenoid field (?)
- Accelerometer
  - inertial detector below internal resonance
  - signal proportional to acceleration
- Capacitive displacement
  - measures position directly, but short range
  - useful as cross-check in laboratory

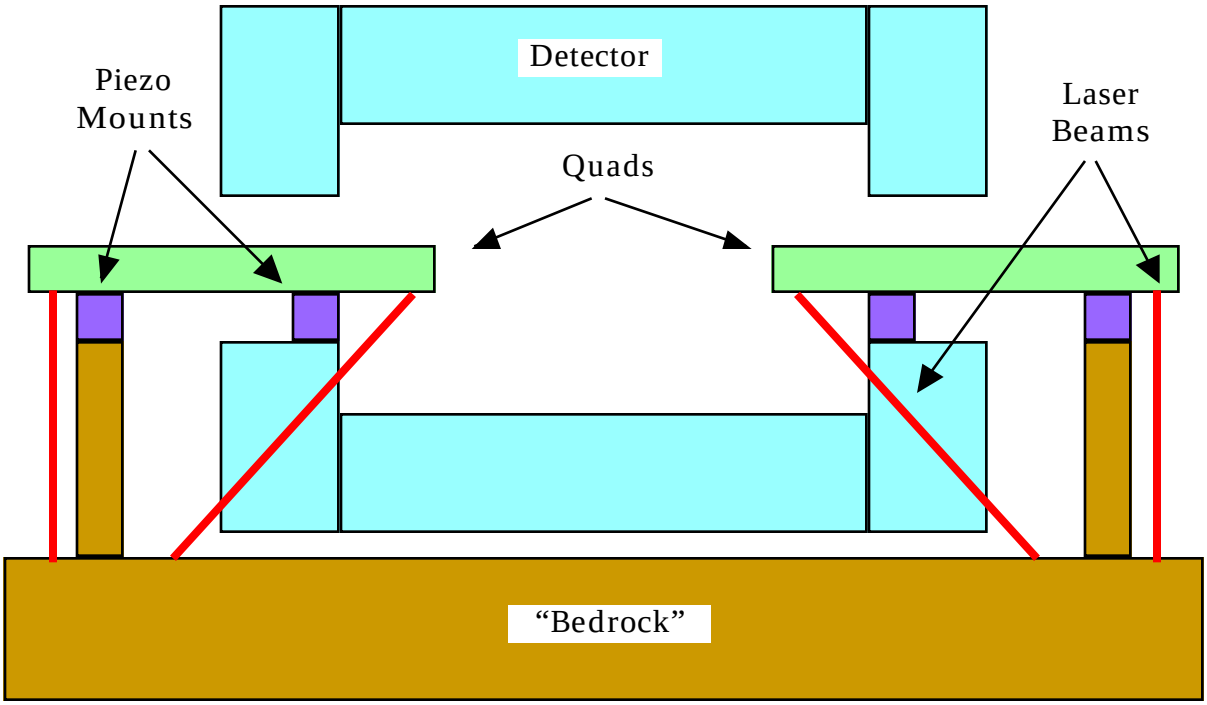
Quadrupole Position Actuator
- Piezoelectric
  - large force, short distance, stiff
- Electrostatic
  - small force, low stiffness
- Electromagnetic (voice coil)
  - large force, low stiffness
  - not compatible with detector solenoid field (?)
- Don’t move the quad, just steer the beam

Algorithm and implementation.

Optical Anchor Concept

Measure quad positions with interferometer(s)

Correct quad positions with piezoelectric(s)



Feedback artificially stiffens the quad supports to simulate a true rigid connection to bedrock

Needs light paths through detector to "bedrock"
(also some external interferometer legs not shown here).

Idea originated at SLAC, developed by Mike Woods, who demonstrated nanometer resolution over 10 meter baseline with interferometer, and piezoelectric movement of 100 kg test masses (but not actual feedback).

UBC Optical Anchor Program

Goal is to stabilize a 100 kg test mass to a nanometer in one dimension relative to a reference 10 meters away.

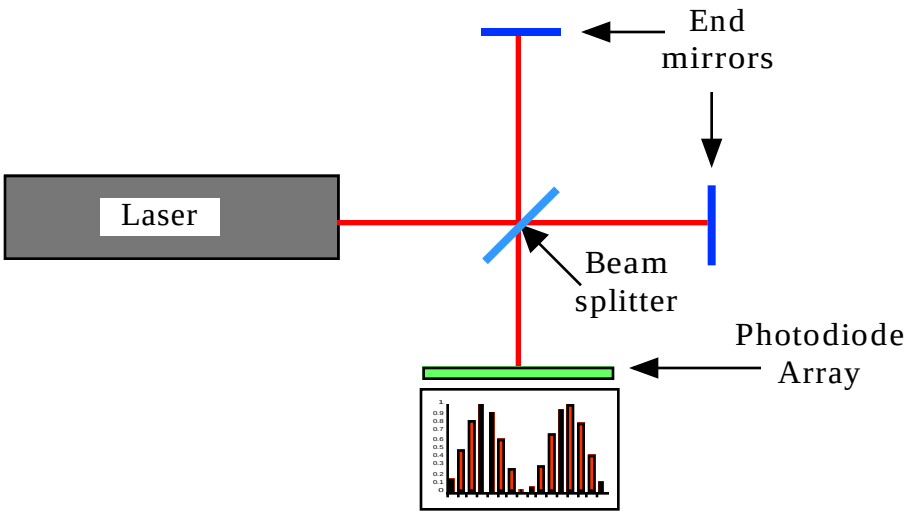
SLAC loaned 10 meter interferometer equipment, funds additional hardware. I have a lab in the UBC Physics basement with data acquisition PC, and a small NSERC grant for students.

Ken Yau (left) came with me to SLAC last spring and operated the interferometer before it was dismantled for shipping. He also set up the DAQ computer, did calculations interferometer parameters, and simulated the feedback in LabView.

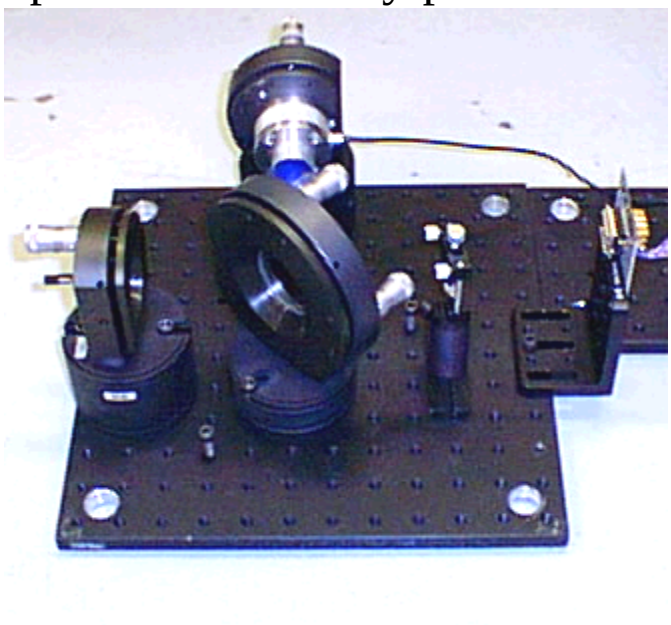
Jason Thompson (right) designed the test platform, and wrote the beginnings of a Linux kernel module for the feedback process.

Interferometer 101

Classical Michelson configuration



Change in length of either interferometer arm shifts the interference pattern. Tilt one end mirror slightly to make fringes on the photodiode array plane.



Far mirror is mounted on a piezoelectric actuator. Re-combined beams are expanded by small cylindrical lens onto photodiode array at right.

UBC Feedback Test Platform

Payload-table mounted to the floor by flexures

- bear the test-mass load in y (vertical)
- stiff in x (normal to interferometer axis)
- flex in z (along interferometer axis)
- can lock to baseplate for reference tests

Position control

- feedback-piezo between tie-rod and payload-table
- exciter-piezo between tie-rod and end anchor

Instrumentation

- one interferometer mirror tied to payload-table
- capacitive position sensor (relative to baseplate)
- can put geophone or accelerometer on table, floor

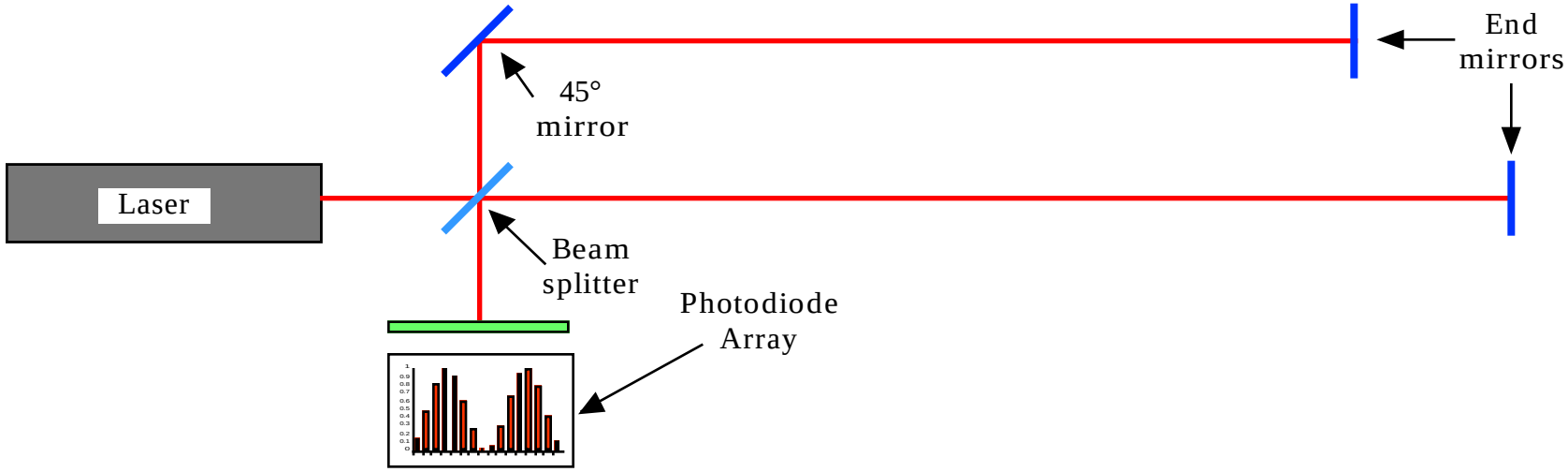
Computing hardware

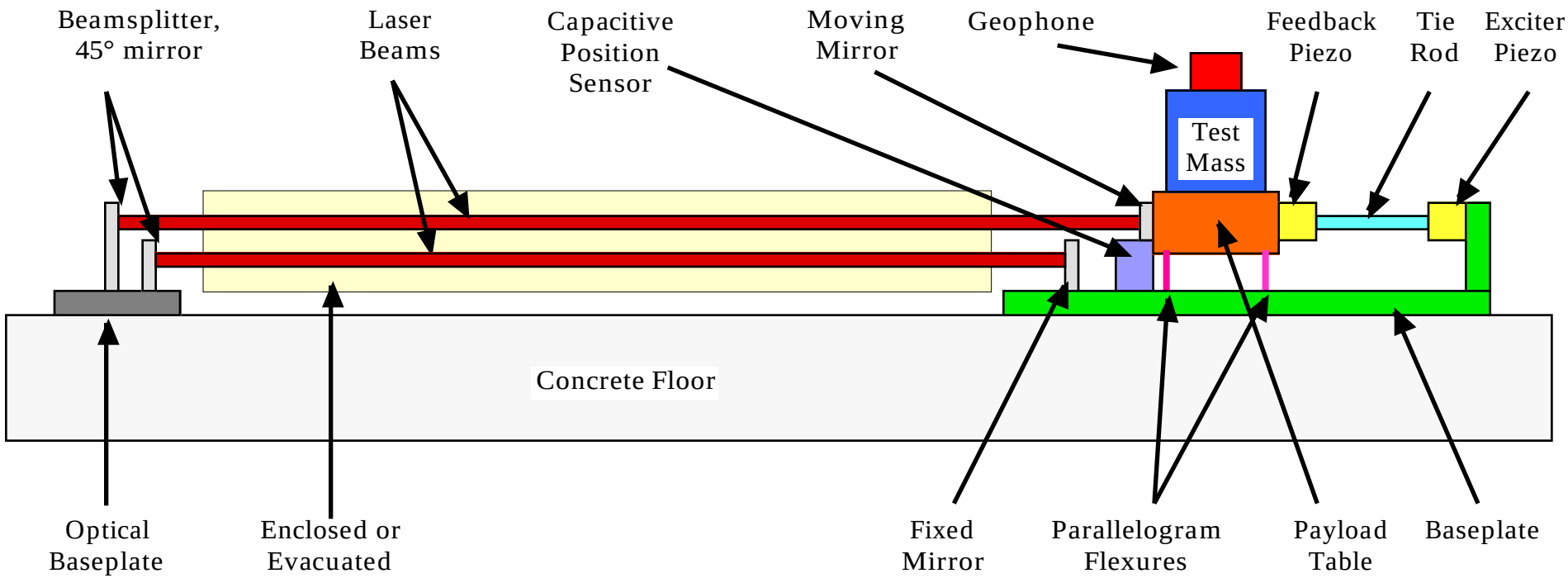
- PC with 100 kHz multiplexed ADC/DAC card
- Card can self-trigger for digitizing, and do DMA

Computing software

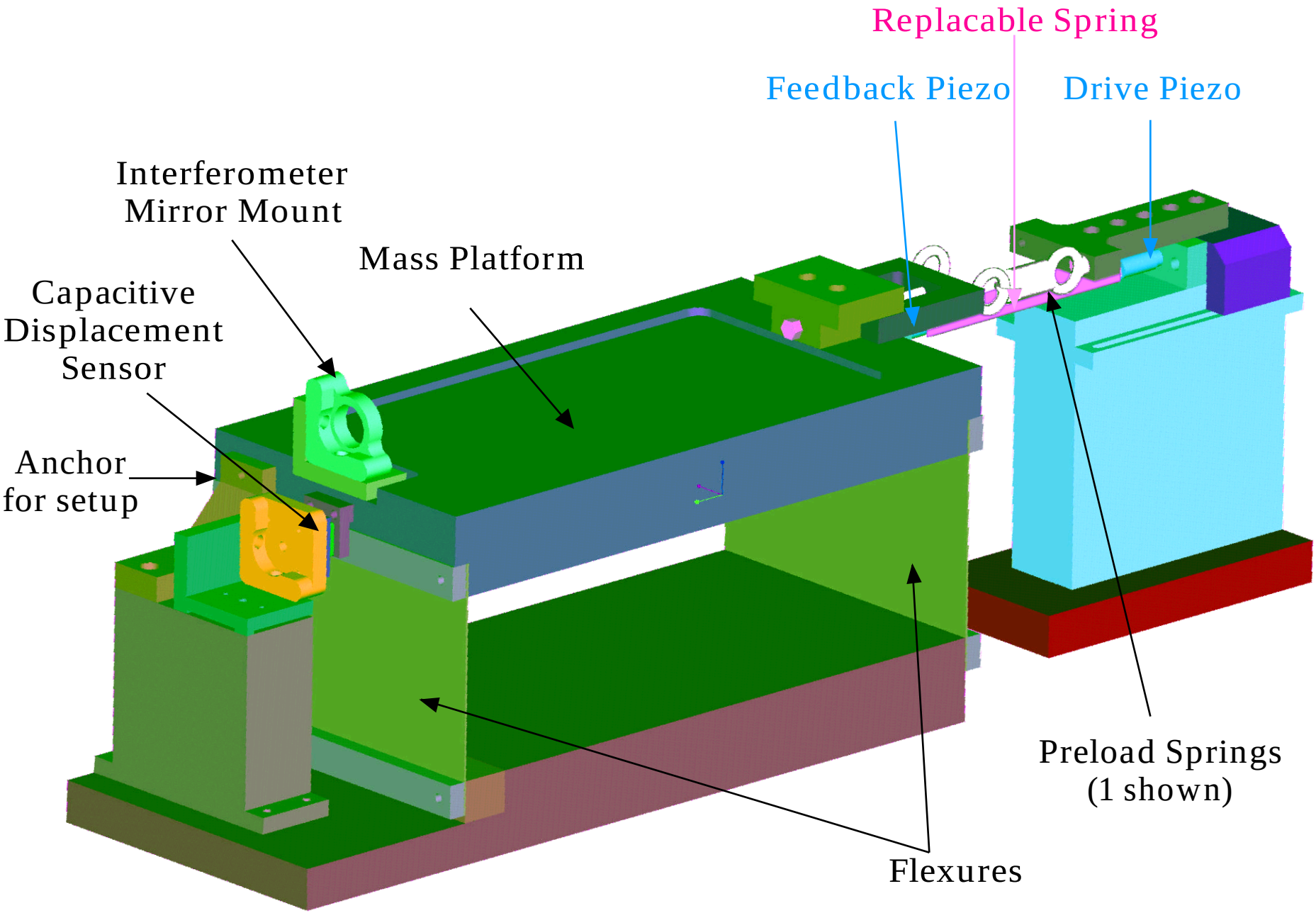
- Writing feedback code as Linux “driver”
- Feedback wakes up, takes over CPU
- Calculates position from ADC data
- Updates state feedback, writes to control DAC
- Re-sets its alarm clock and goes to sleep

Proposed Interferometer + Test-Mass Configuration at UBC

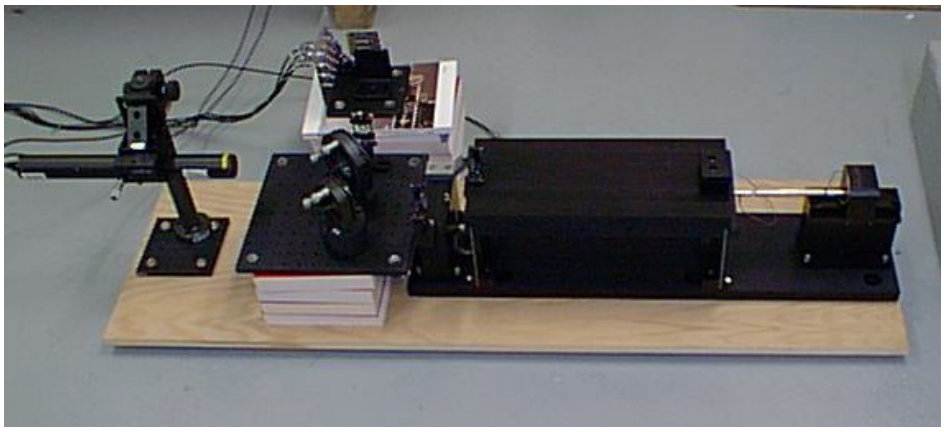


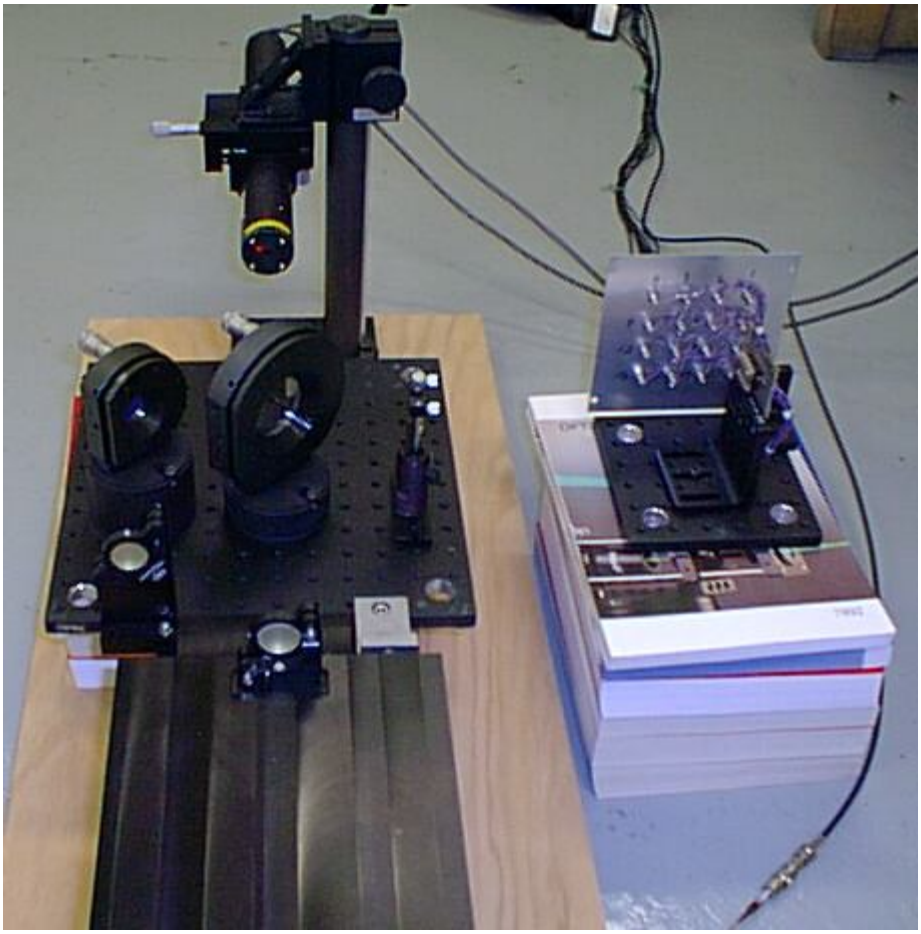


UBC Nanometer Vibration Stabilization Test Platform Design



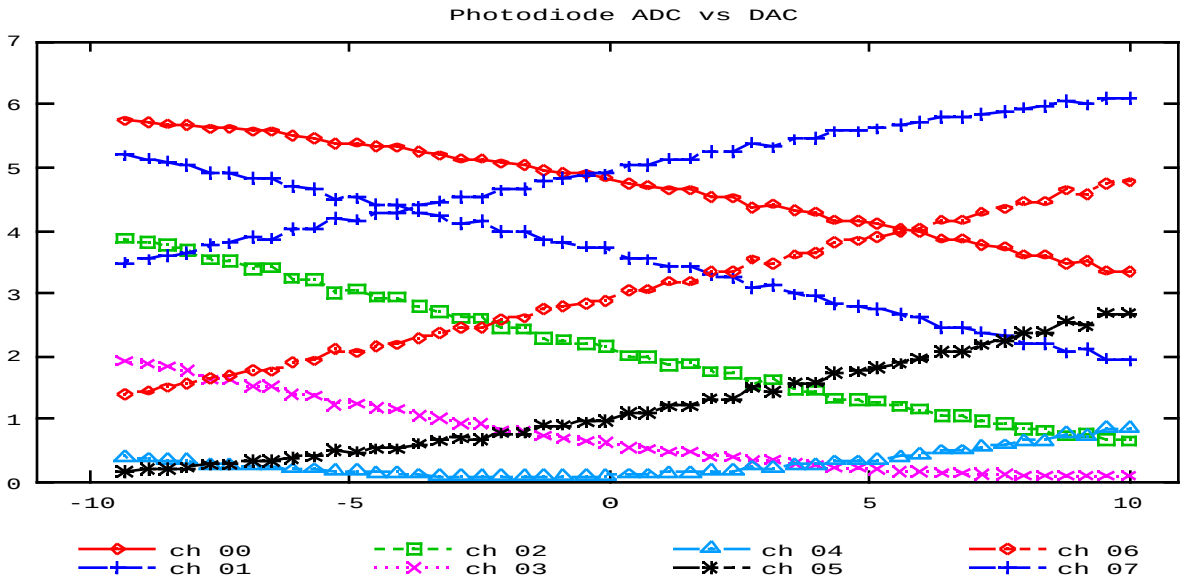
UBC Feedback Test Platform with Short Interferometer



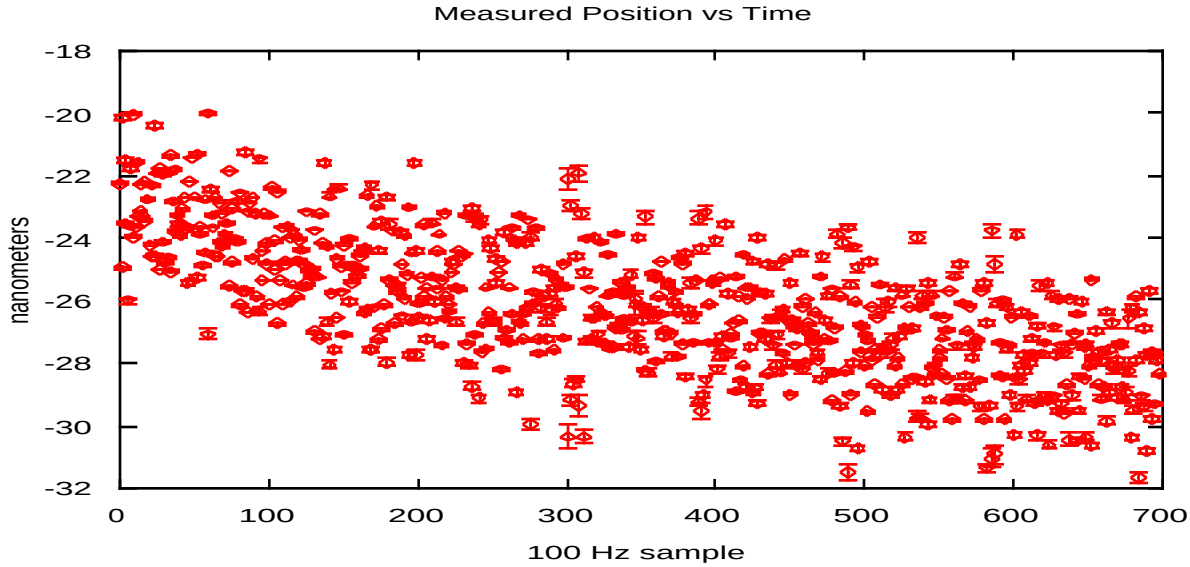


Small Test Interferometer Results

Connect signal generator to mirror on piezo, and digitize 8 photodiodes plus piezo voltage. Fit photodiode vs drive to sinusoids with common "frequency" for all 8 diodes.



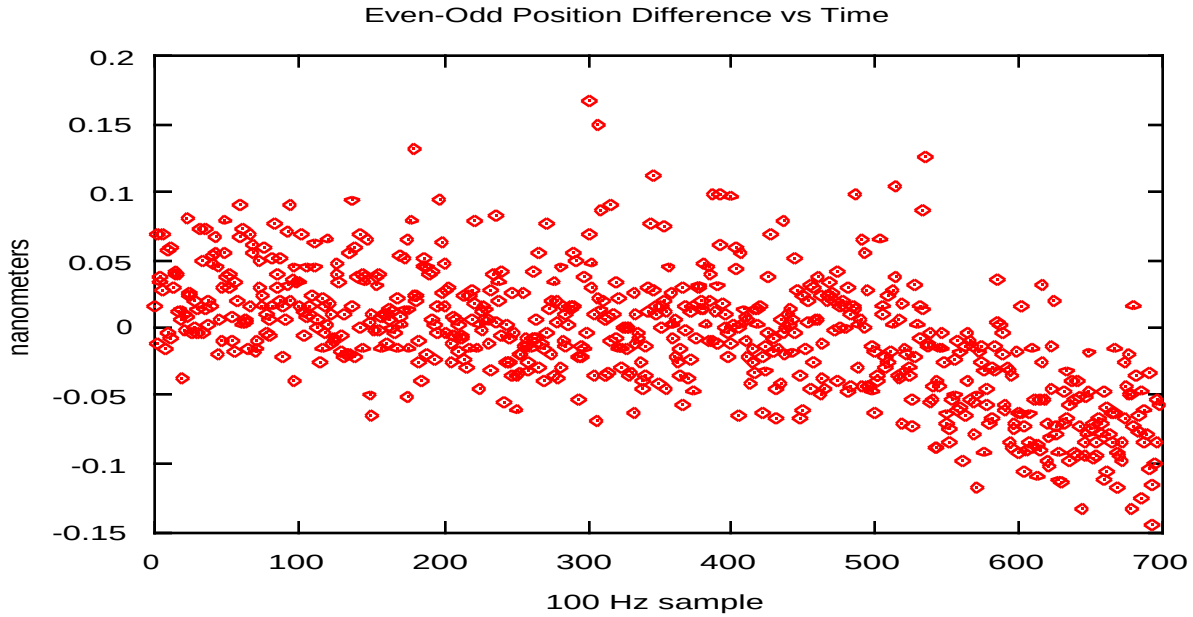
Digitize at 100 Hz with generator off. For each sample, fit 8 photodiodes with above parameters fixed but a common "phase." Mirror position is $\Delta x = (632\text{nm}) \times \phi / 2\pi$



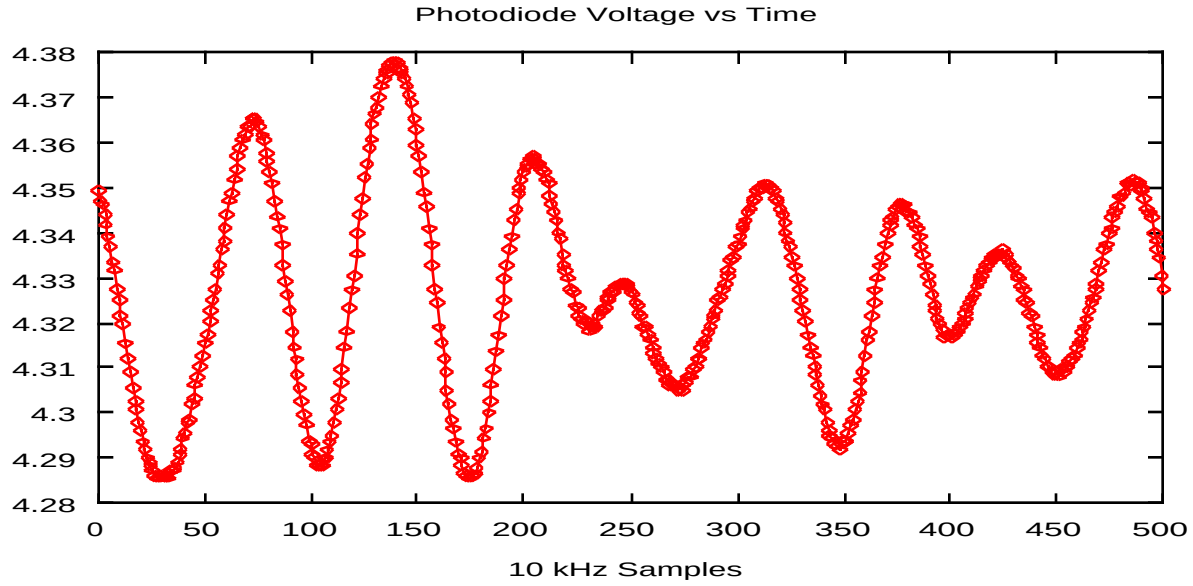
Noise is a few nanometers, so is drift over a few seconds

Small Interferometer Results (2)

It's actually much better than it looks. If we fit the even photodiodes and the odd photodiodes separately, and plot the difference vs sample, they agree to a small fraction of a nanameter

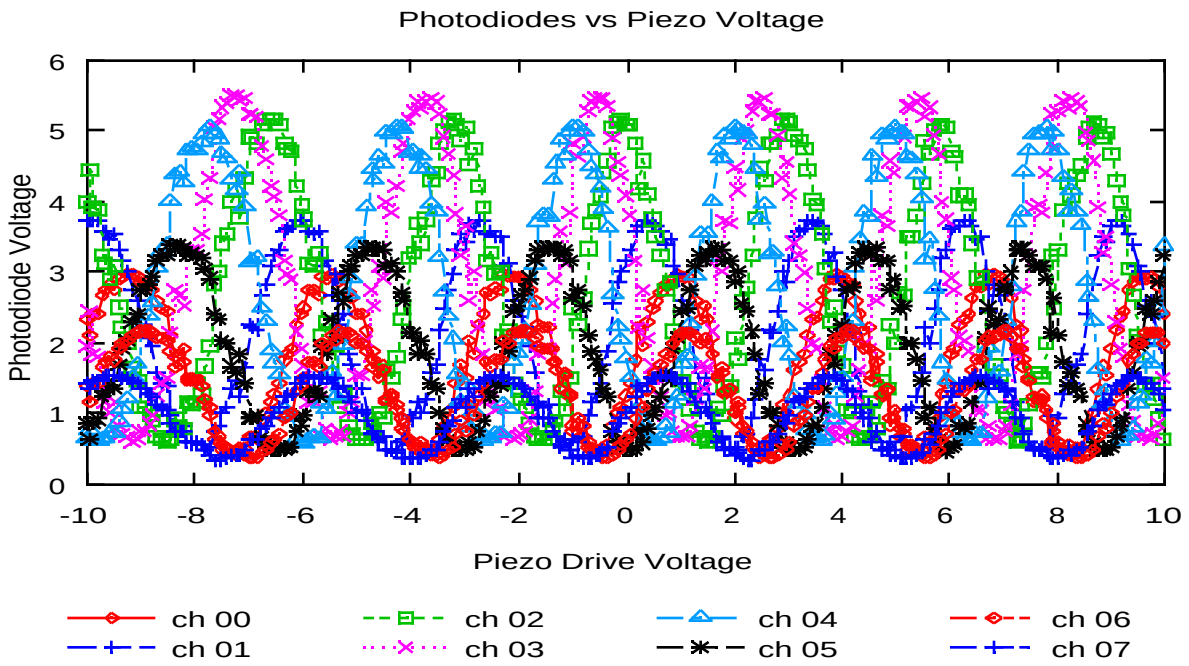


Sampling at 10 kHz shows what's really happening

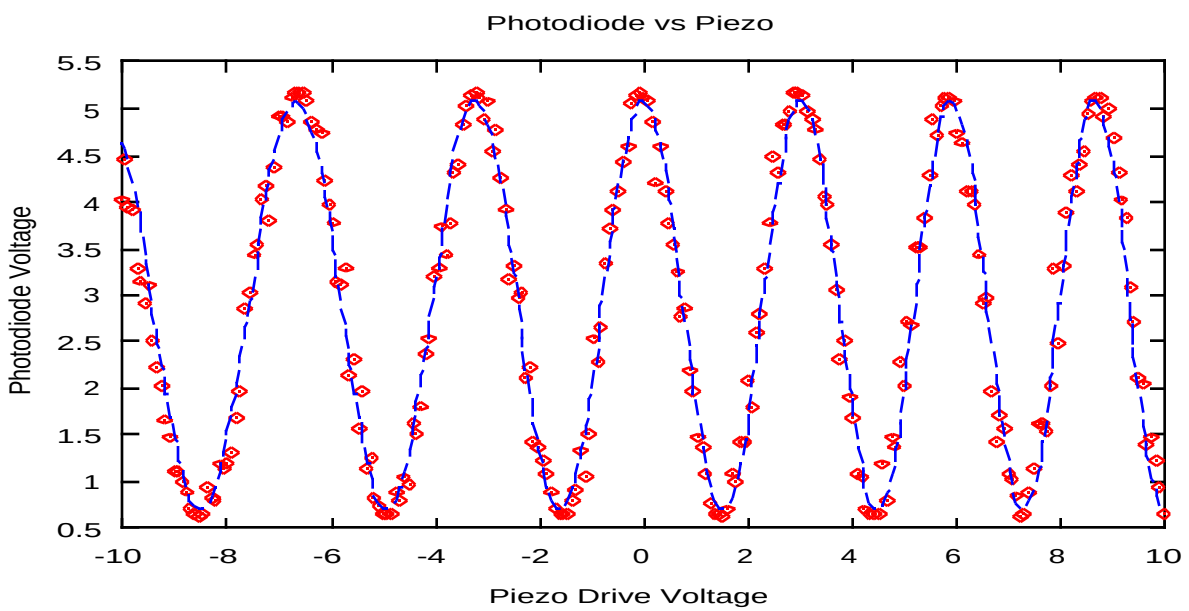


Early Platform Results

The piezo on the big platform moves more per volt than the (larger) one on the small test interferometer, so the calibrations span many cycles.

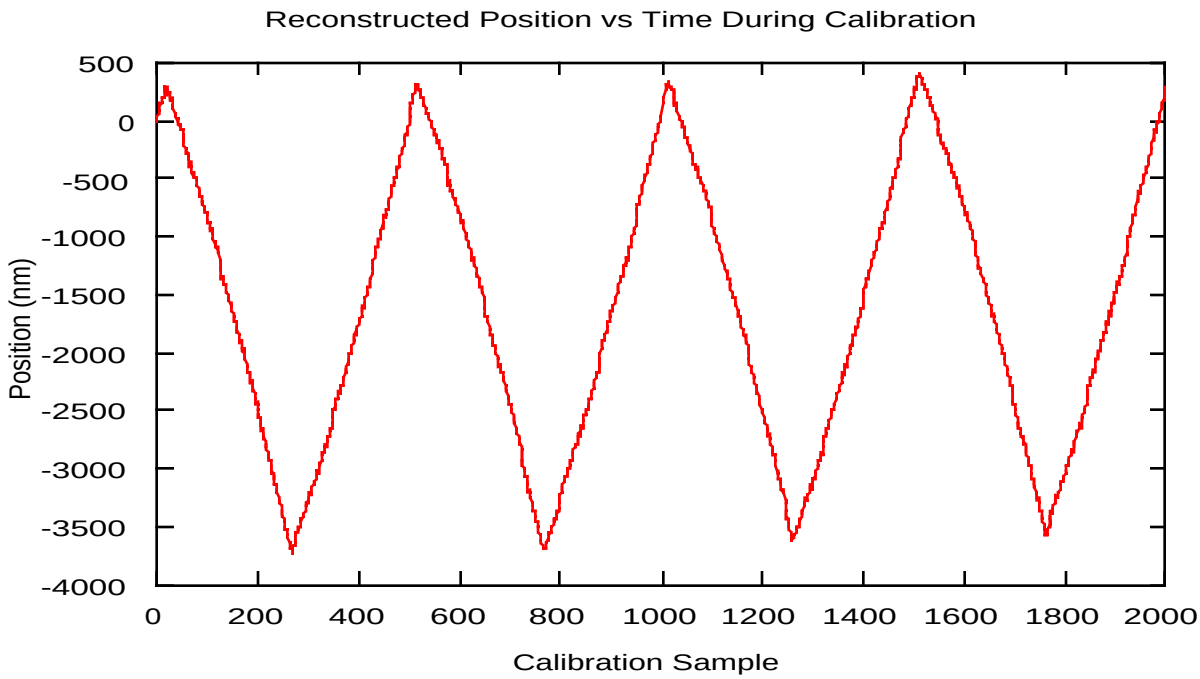


The "frequency" changes noticably, so we add a quadratic term to phase vs piezo voltage:

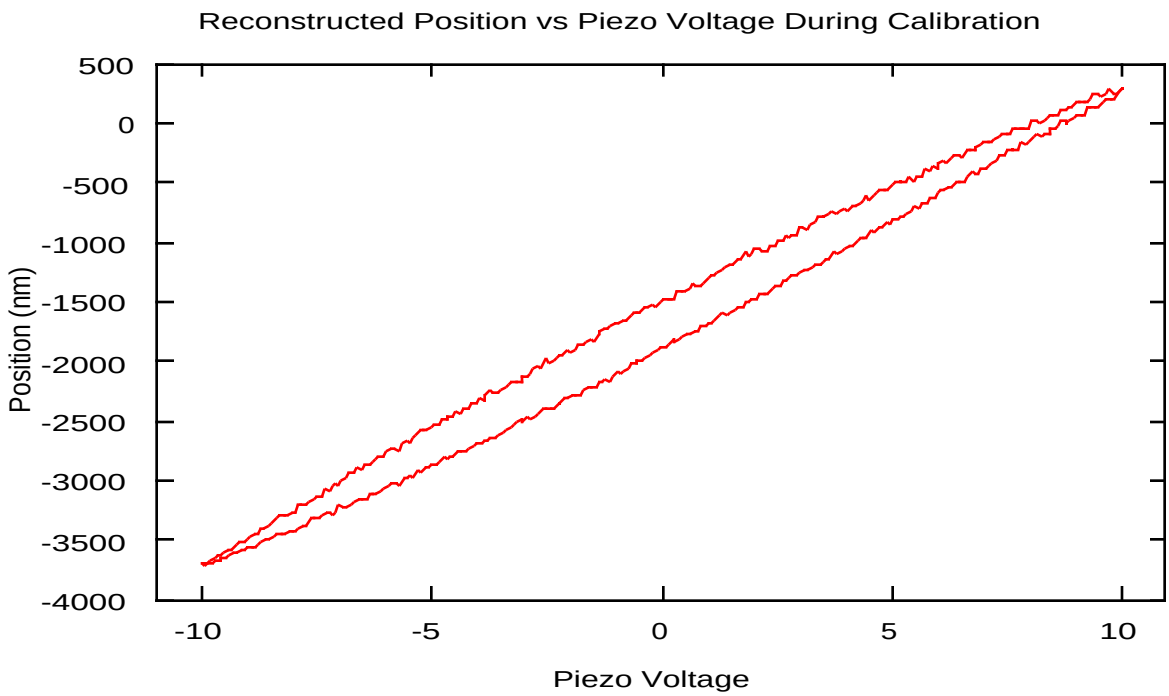


Early Platform Results (2)

If we use the calibration fit to reconstruct the motion during calibration, we can see the nonlinearity.

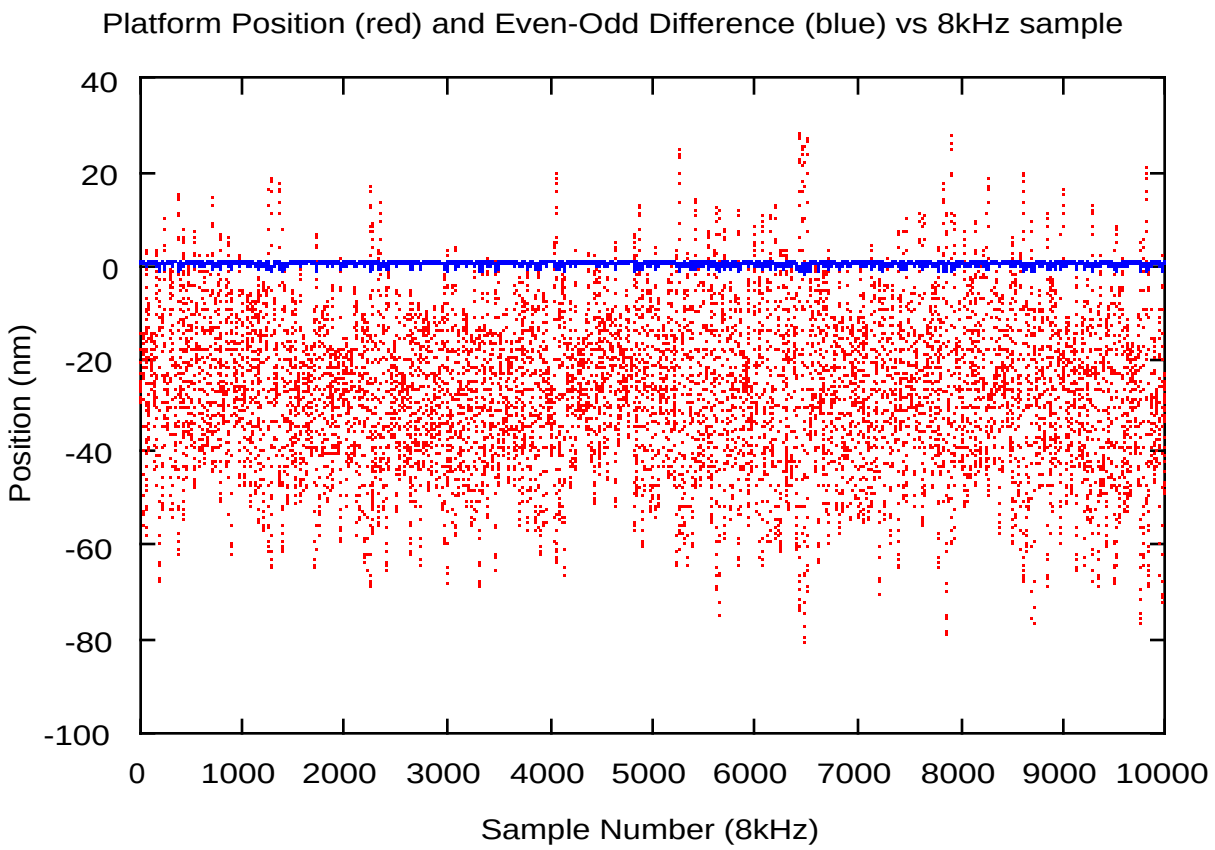


If we plot position vs piezo voltage, we see hysteresis:



Early Test Platform Results (3)

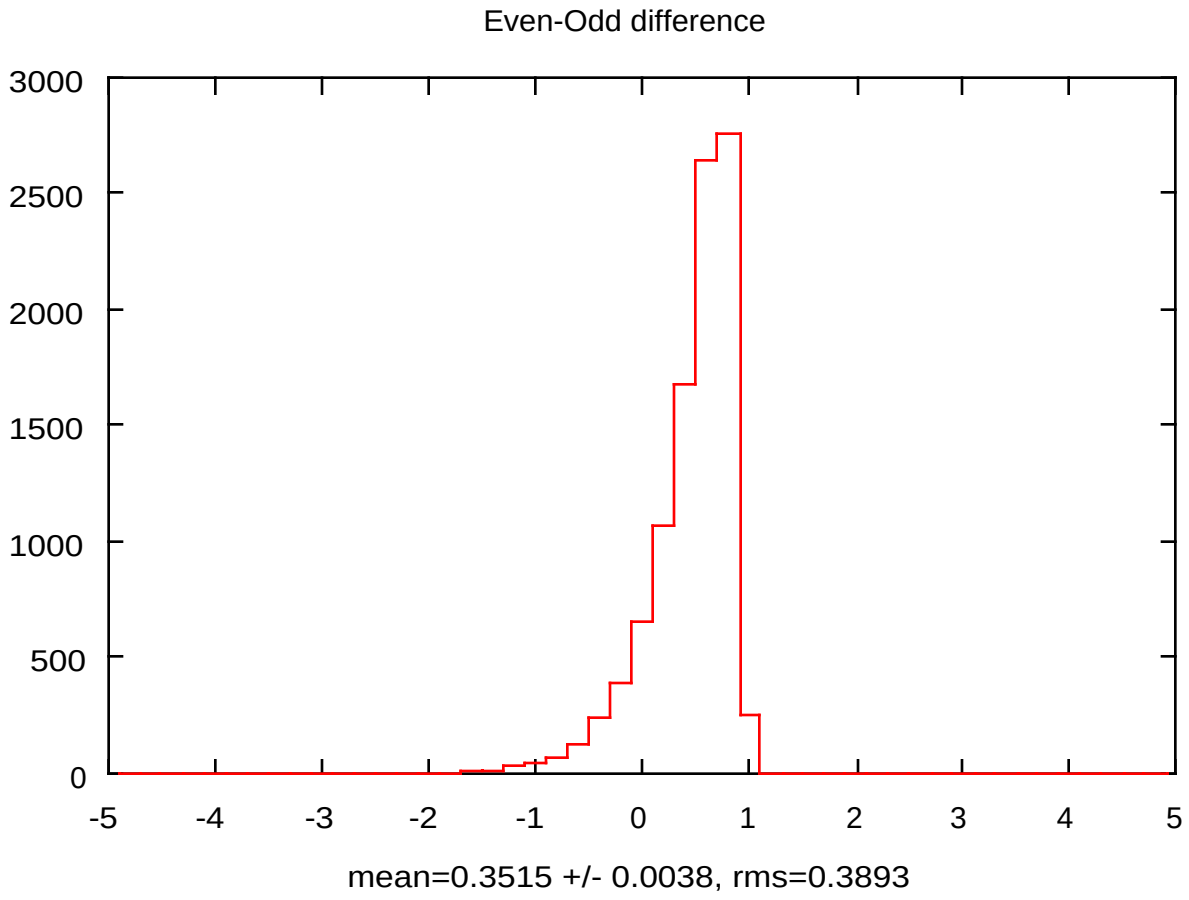
Using the same calibration on higher frequency data at fixed piezo voltage, we can see the platform motion, and also calculate the difference in the platform position using the odd and even photodiodes.

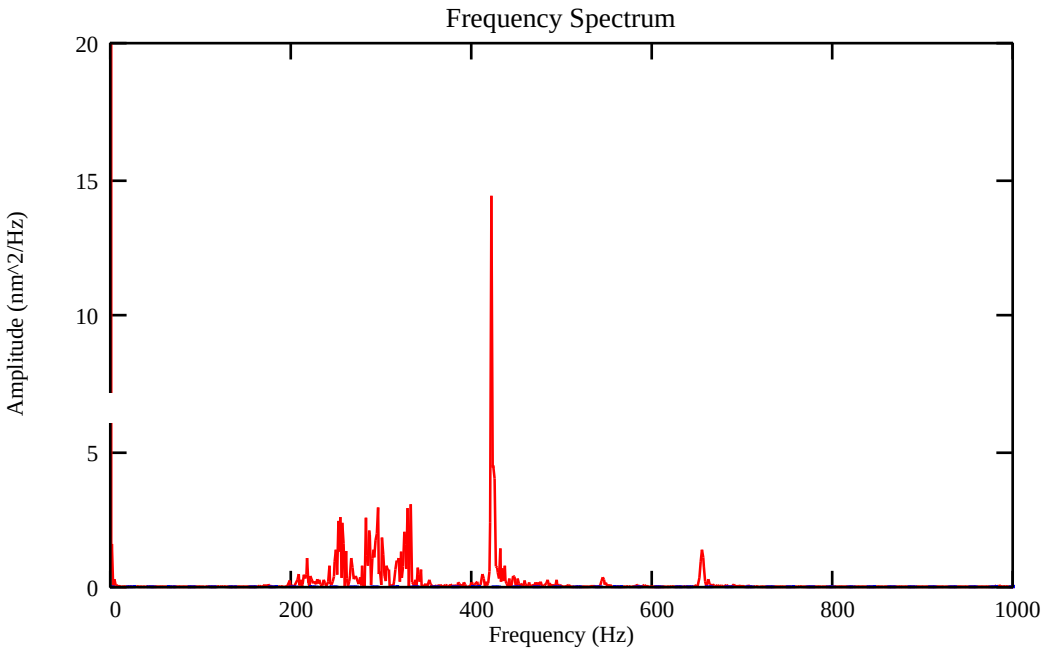


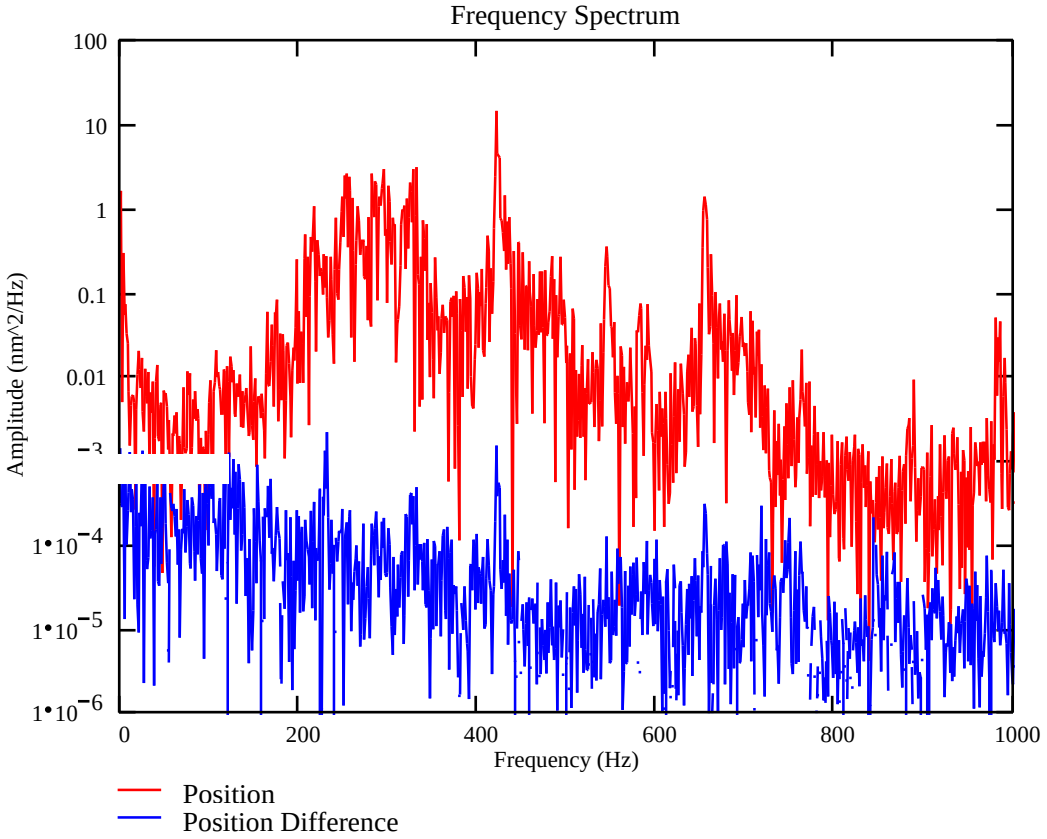
Clearly the even-odd difference is much less than the motion of the platform.

Early Test Platform Results (4)

The RMS of the even-odd difference is less than half a nanometer. The distribution isn't Gaussian.




Frequency Spectrum
Amplitude (nm^2/Hz)
20
15
10
5
0
0
200
400
600
800
1000
Frequency (Hz)


Frequency Spectrum
Amplitude (nm^2/Hz)
100
10
1
0.1
0.01
1•10−4
1•10−5
1•10−6
0
200
400
600
800
1000
Frequency (Hz)
Position
Position Difference

Computing Issues

Feedback needs to run at a few kHz (vibrations at higher frequencies are expected to be small).

Algorithm is implemented in software rather than hardware: required rate is low enough, and sophisticated algorithms are easier to implement and tune in software.

Feedback process must run continuously, and not be swapped out by the operating system. So we can't use Windows, or a normal Linux application.

We looked into using DOS, but it's hard to find DOS drivers and compilers for modern hardware.

A real-time operating system like VxWorks would be OK, but expensive (license, VME crate and processor, different ADC/DAC hardware, separate DSP processor). VxWorks users at TRIUMF encouraged us to stick with Linux unless we really need much higher performance.

Present approach is to implement the feedback as a Linux kernel module on a 500 MHz Pentium-3 PC, with a Keithley KPCI-3108 card (100 kHz ADC multiplexed between 16 channels, plus DAC, counter/timers, and digital I/O).

Linux Kernel Hacking Experiences

There's lots of documentation, and of course you can read the headers and source. But the kernel changes faster than the documentation, and backward-compatibility is often sacrificed for efficiency or new features like multiple platform or multi-processor support.

Writing kernel modules at the "Hello, world!" level is as easy as writing normal Linux text-based user programs. Just buy the right books and follow the directions. Modules can be loaded and unloaded dynamically, with no rebooting required.

The interface to user-level programs by reading and writing byte streams in the **`/dev`** directory is easy to add (what to do with the byte streams is still up to you).

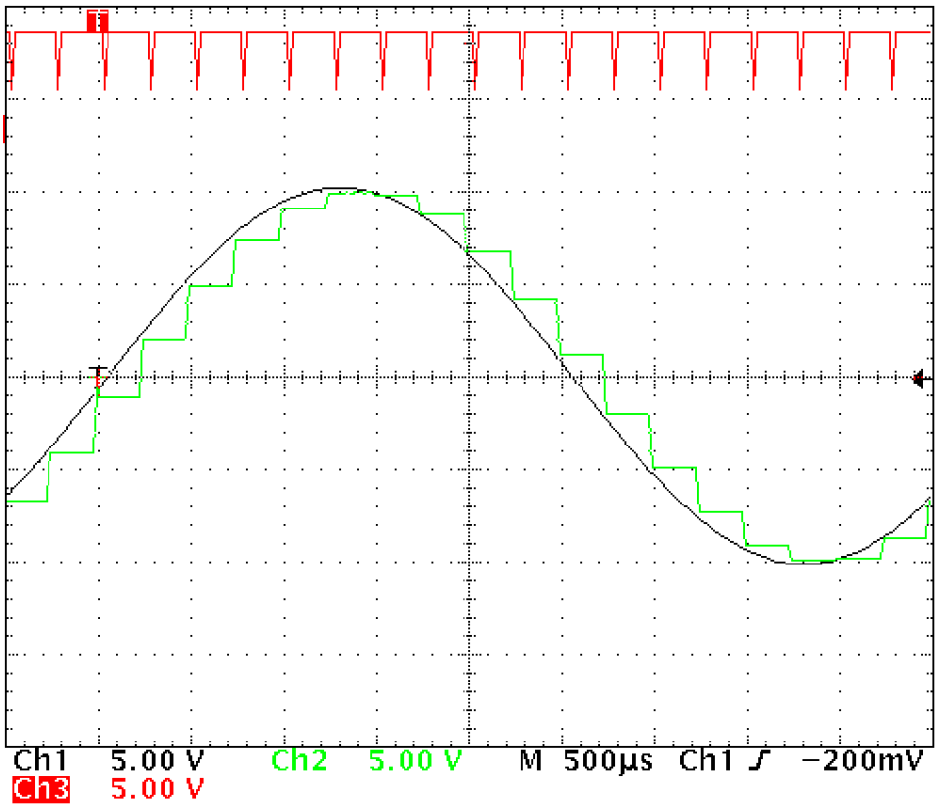
Talking to the hardware through the PCI bus to take single ADC readings, or writing single DAC values, is also easy.

Using hardware interrupts is less easy, and most mistakes will freeze or crash the machine. The ADC card documentation was particularly poor on interrupts. But Linux interrupt overhead is low enough to run at 100 kHz.

The ADC card can write into main memory via DMA without interrupting the CPU (this is a good way to experience a Linux kernel panic firsthand).

Linux Kernel Hacking Status

We have written a module that programs the ADC to read all 16 channels at 10 MHz, DMA the data, and interrupt the CPU, which then writes the last ADC reading to the DAC. The cycle repeats at 4 kHz.



Black: ADC input signal (200 Hz sine wave),
Green: DAC output signal, Red: 4 kHz cycle clock

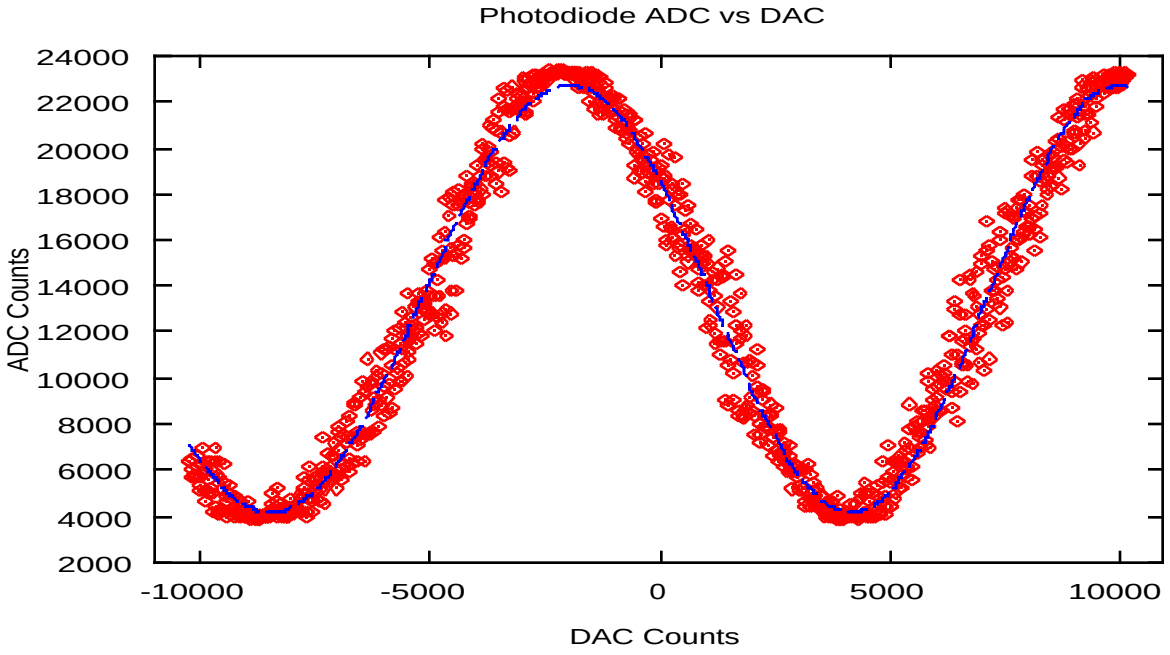
There is no perceptible slowdown on the PC, and interrupt latency jitter is only microseconds (once we told the IDE disk driver to allow itself to be interrupted).

The process still dies randomly after a few hours, and we put in an auto-restart watchdog.

Linux Kernel Hacking Status (2)

The DMA data acquisition has been integrated into a standard Linux “character device” driver, and has been used to acquire interferometer data (although the results I just showed were acquired with LabView in Windows).

The calibration procedure (measuring photodiodes while moving the piezo) has been integrated into the driver, and can be activated by a user program, which waits then requests the data, fits it, and downloads the calibration constants.



The next step is to move into the interrupt handler the much simpler fit that uses the calibration constants to turn a photodiode snapshot into a position. (I almost got it done before coming here, and might even get it done before leaving).

Future Plans

Feedback engineering

- optimization of feedback parameters
- investigate algorithms beyond PID

Interferometer non-invasiveness engineering

- corner reflector to minimize remote adjustments
- two laser colors for air density correction
- sealed / inflated gas-volume along light paths

Stabilize more degrees of freedom

- extended platform and two beams

Simulate internal resonances of support structure

- second platform and spring in series with first

It is possible to use an interferometer for “absolute” position measurements by frequency scanning, which would be useful for in-situ micron-alignment checks.

Participate in future IR stabilization exercises