Optical Interferometer Vibration Control

Tom Mattison
University of British Columbia

ILC Physics, Detector & Accelerator Workshop
Snowmass, Colorado
24 August 2005

Outline

Overview

Interferometry Progress

Canonical and State-Vector Feedback

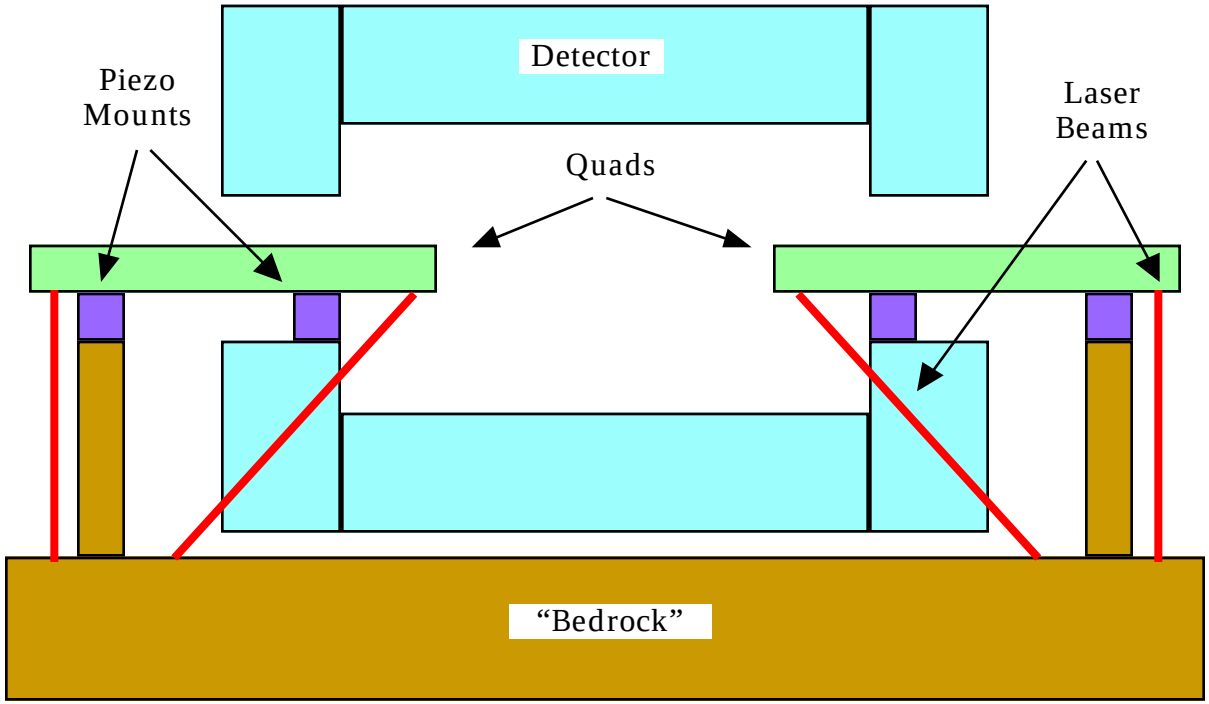
Using State-Vector Feedback

Conclusions

Canonical Optical Anchor Concept

Linear collider must focus beams to nanometers for adequate luminosity,
so the final quads must be stable to sub-nanometer accuracy, despite awkward location.

Measure quad positions with interferometer(s) referenced outside detector
Correct quad positions with piezoelectric(s)



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

Needs light paths through the detector to “bedrock” or other external references
(and other external interferometer arms not shown here).

Optical Anchor R&D at UBC

The original goal at UBC was to demonstrate feedback control of a 100 kg test mass to sub-nanometer precision over a 10 meter baseline in one dimension (horizontal).

Borrowed Mike Woods' HeNe laser, beam-splitter, mirrors, photodiode array, electronics.

Bought PC with ADC/DAC/DIO card, wrote Linux driver for near-real-time response, and ADC/DAC/DSP box for true real-time response but less compute power.
Bought piezos, piezo drivers, capacitive position sensor, and electronics

Built test platform with 10 kg aluminum block on flexures (can add more mass), variable spring constant, piezo actuator, variable preload, interferometry mirrors, cap-sensor.

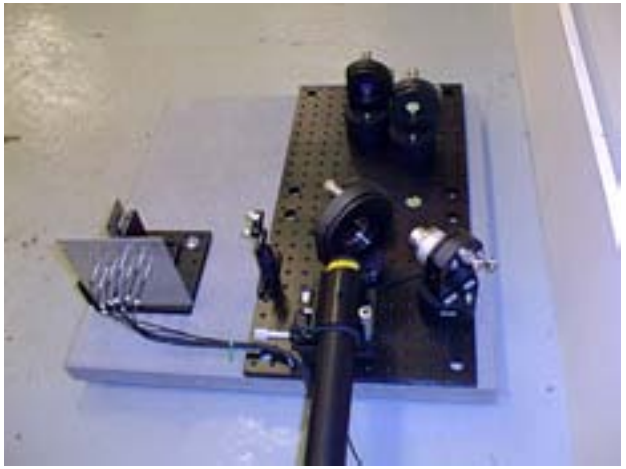
Learned how to calibrate Michelson interferometer (decoupled from test platform), and do reconstruction inside an interrupt handler at 10 kHz with sub-nanometer precision.

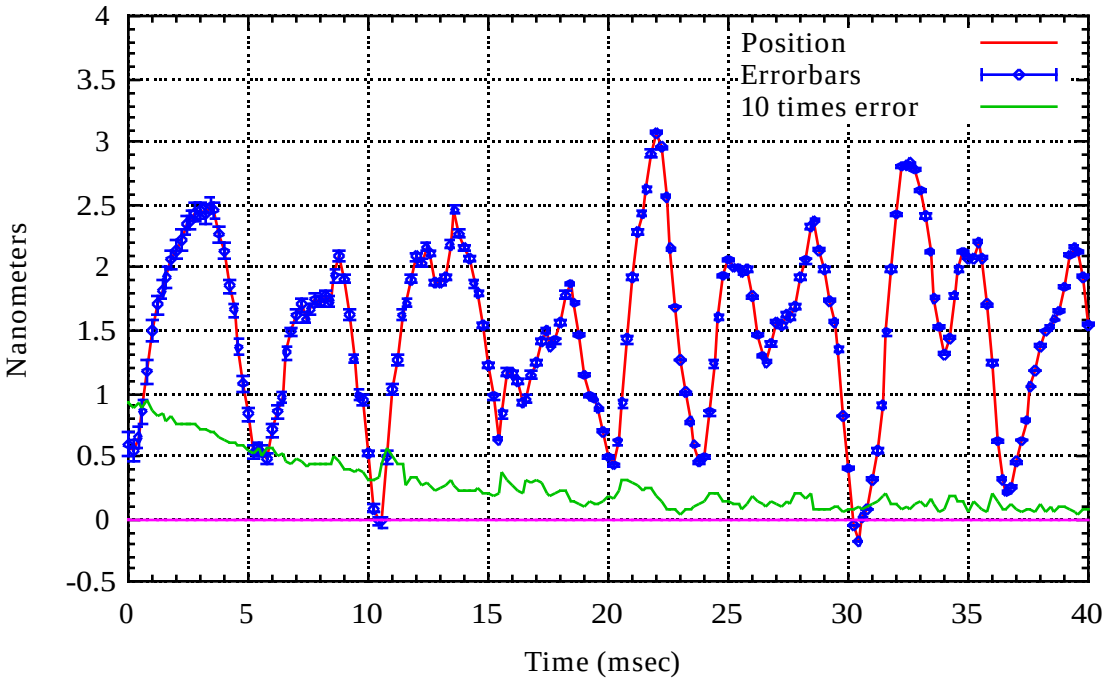
Made rather elaborate PID plus tunable-filters control algorithm for DSP box for the 10 kg test platform (using the capacitive position sensor rather than the interferometer). Feed-back worked, but not as well as we had hoped.

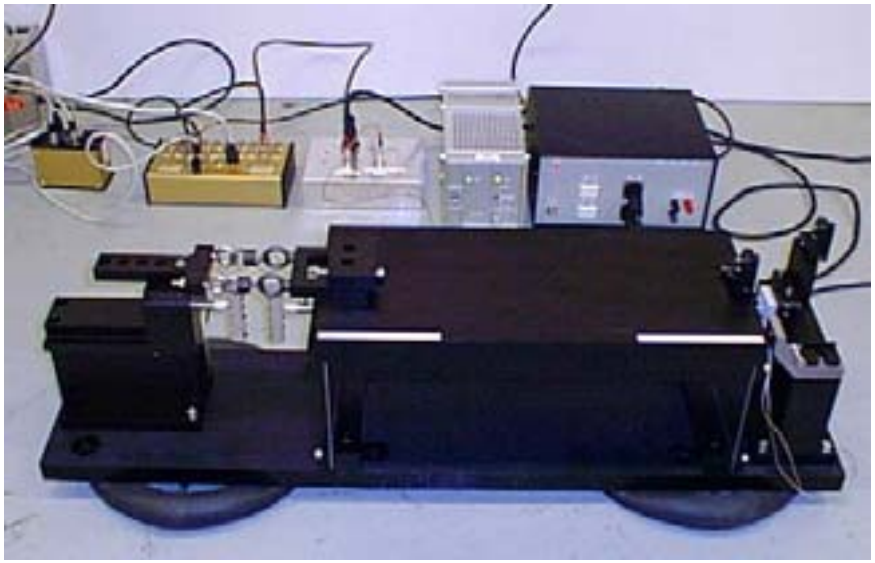
Also did PID feedback of the stand-alone interferometer with mirror on a piezo.

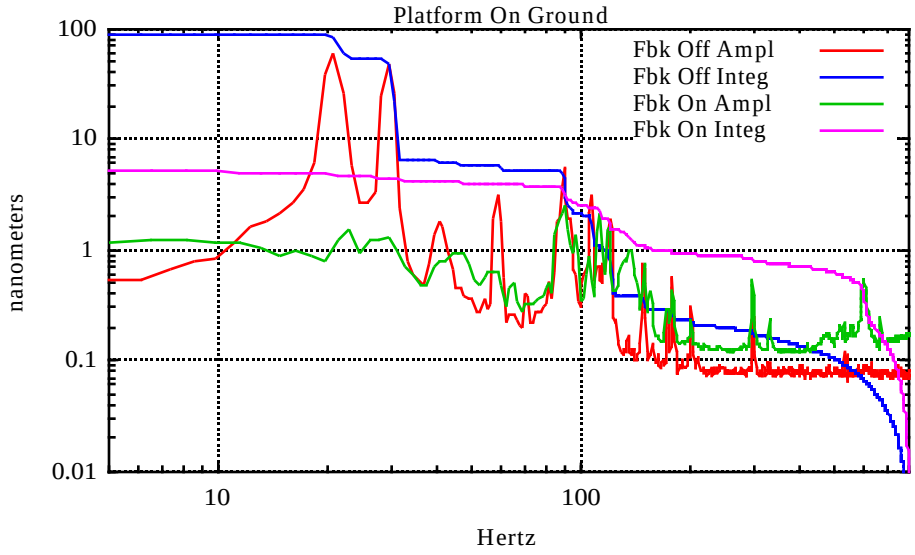
Have previously reported (Nanobeam 2002) **interferometer resolution** of **0.01 nm (1/10 angstrom)**, feedback **control** of interferometer **mirror** to **0.06 - 0.35 nm**, **control** of 10 kg **platform** from **90 -> 5 nm** (on ground), and **4.5 -> 1.5 nm** (isolated).

Hardware and Past Results







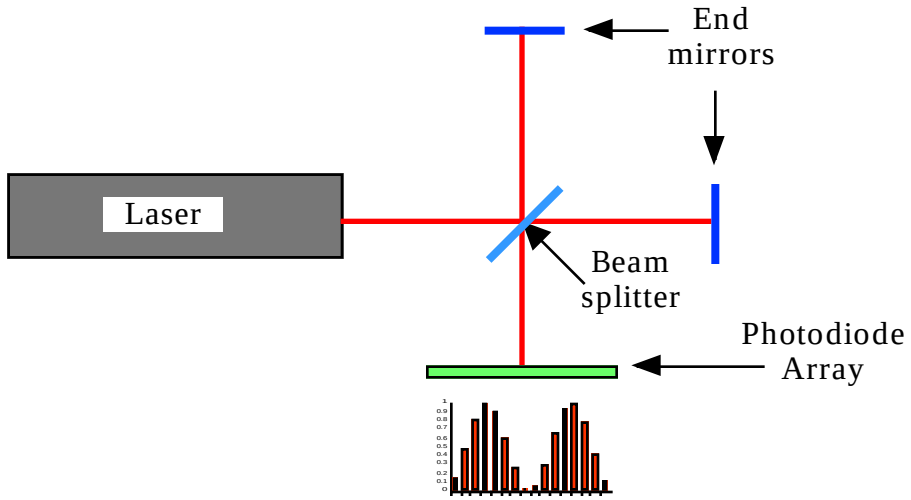


Interferometer Reconstruction

Michelson interferometer, with beams at slight angle at photodiode array to make fringes on photodiode array.

Moving end mirror moves fringe pattern, fit pattern to measure motion to a tiny fraction of a wavelength.

632 nm HeNe is ≈100 nm / radian, but path length changes by twice the mirror motion, so it's 50 nm / radian.



Voltage at photodiode k is modelled as $V_k = (1 + C + kC')\left[A_k + B_k \sin(\psi_k + \phi + k\phi')\right]$

ϕ is the phase that contains the arm-length-difference information.
A and B are the average intensity and interference amplitude at a given photodiode
ψ is the relative phase of the interference at a given photodiode

C and C' are relative laser intensity and intensity-gradient across photodiode array.
ϕ' is the phase-gradient across the photodiode array (relative to calibration).
C, C', ϕ' are all nominally zero right after a calibration.

Fit each set of photodiode measurements for ϕ and ϕ', with slow update of C, C', and A_k.
Measurement error calculated from residuals on the fit.

Interferometer Calibration

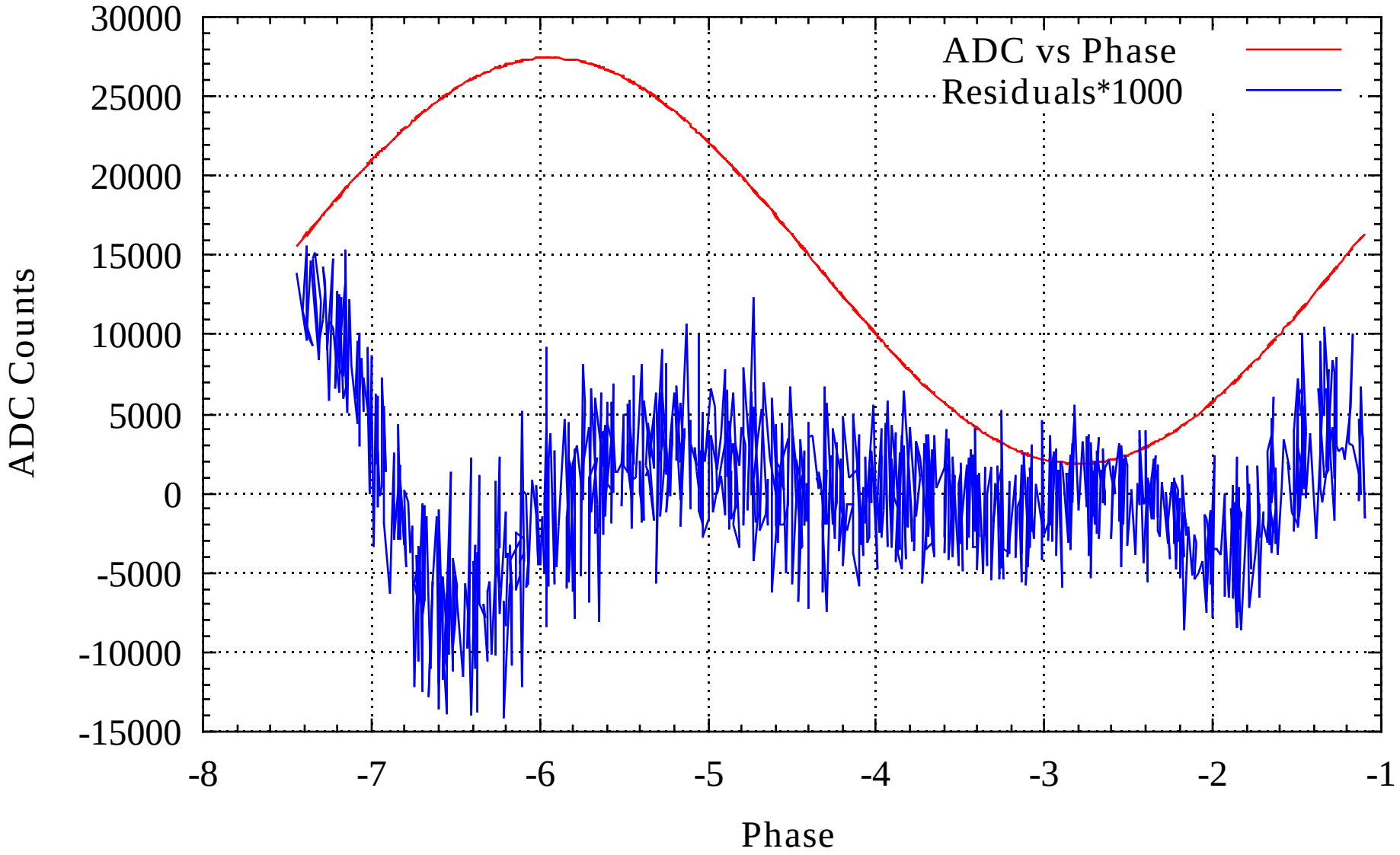
One mirror of the interferometer is on a piezo for calibration.

Originally we calibrated by assuming ϕ is a polynomial in the piezo voltage and fitting each photodiode for A, B, and the polynomial coefficients, assuming the other parameters were zero. That method breaks down when the platform is moving a significant fraction of a wavelength due to disturbances.

The next method was to guess A and B from the maxes and mins of the piezo scan,
then guess ψ from $\psi_k = \psi_{k-1} + \left\langle \sin^{-1} \frac{V_k - A_k}{B_k} \right\rangle - \left\langle \sin^{-1} \frac{V_{k-1} - A_{k-1}}{B_{k-1}} \right\rangle$ with $\psi_0 = 0$,
then fit each piezo scan setting for ϕ using the above A, B, ψ ,
then use those ϕ values to fit for A, B, ψ for each photodiode, and iterate a few times.

This produced calibration residuals of a few ADC counts for signals of >10000 counts.

Old Interferometer Calibration Data



Improved Interferometer Calibration

Recently, we made a new piezo-mirror that moves many wavelengths, instead of a bit less than one wavelength. The residuals from these calibrations were clearly periodic in phase. (The effect was visible in the old calibration residuals too, but we didn't know what caused it). So we did some simulation studies.

These demonstrated that alternating between fitting for ϕ for each "row" of calibration data with fixed A, B, ψ, then fitting "columns" of calibration data for A, B, ψ with fixed ϕ didn't actually converge to the right answer in any reasonable number of iterations!

The procedure can give A, B, ψ values that give ϕ values that oscillate harmonically around the true ϕ value as ϕ varies, and these wrong ϕ values prevent the fit from recognizing that A, B, ψ are wrong.

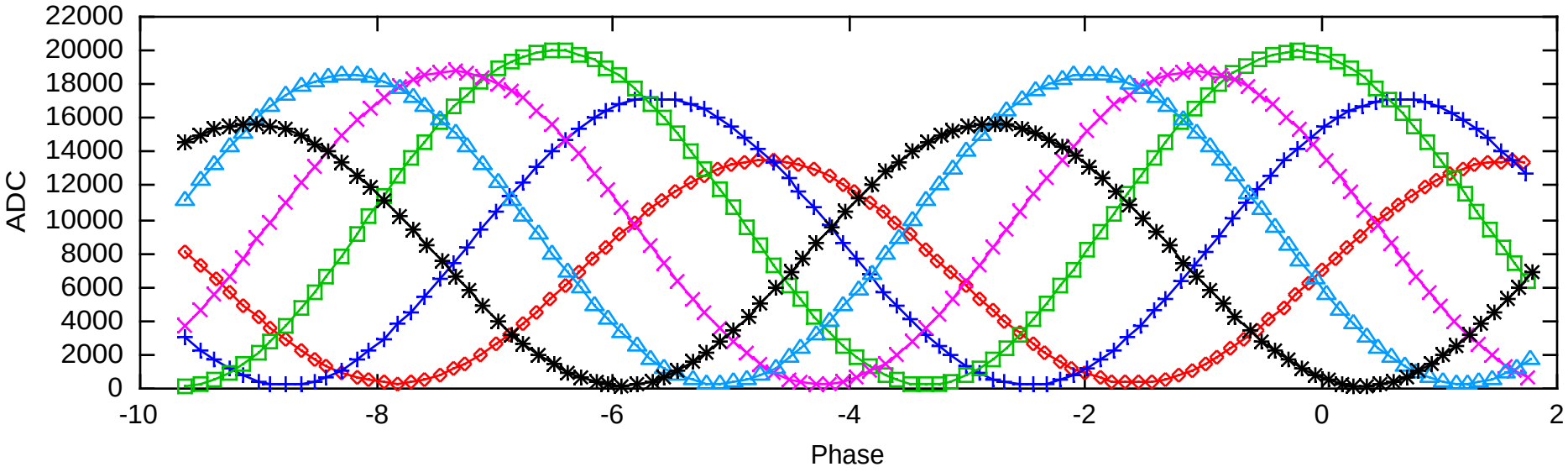
So we also introduced some extra parameters into the calibration fit

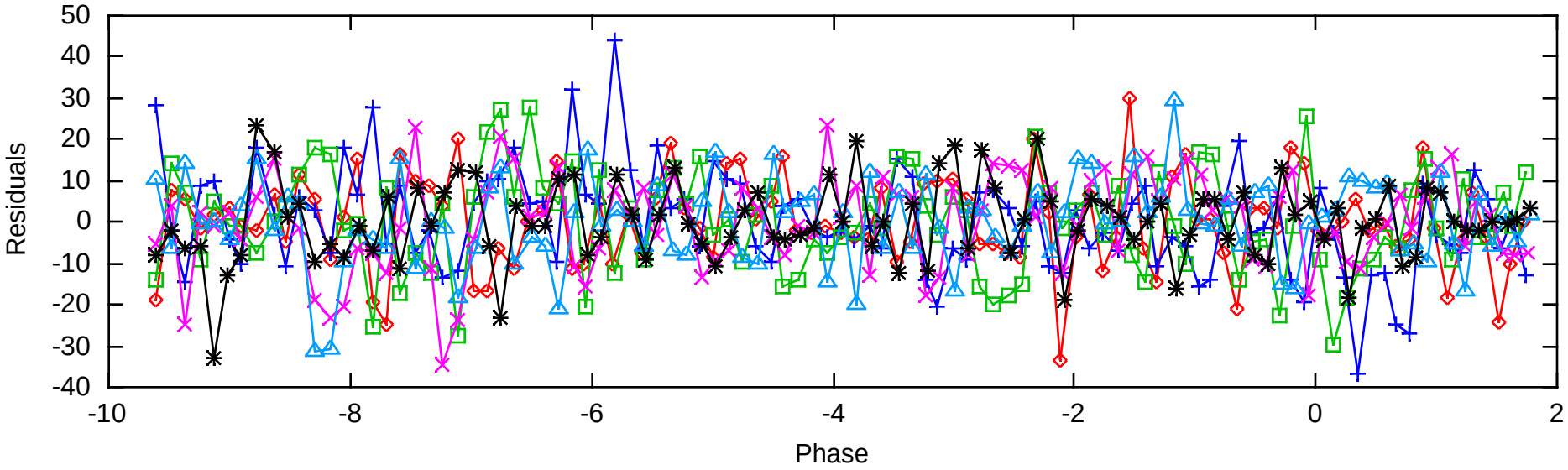
$$V_k = A_k + B_k \sin\left(\psi_k + \phi + k\phi' + P_k \sin\phi + Q_k \cos\phi\right)$$

The P and Q parameters let the fit recognize the harmonic distortion, and the resulting A and B are more correct. After the first iteration, P and Q are nearly consistent with zero. The kϕ' term allows for the fact that the mirror tilts slightly as the piezo moves it.

Ideally we should fit both the rows and the columns of the calibration data simultaneously, to get rid of harmonic distortions of all orders. Since there may be hundreds or thousands of rows (piezo settings), this is a big fit. I'm working on the software to do this.

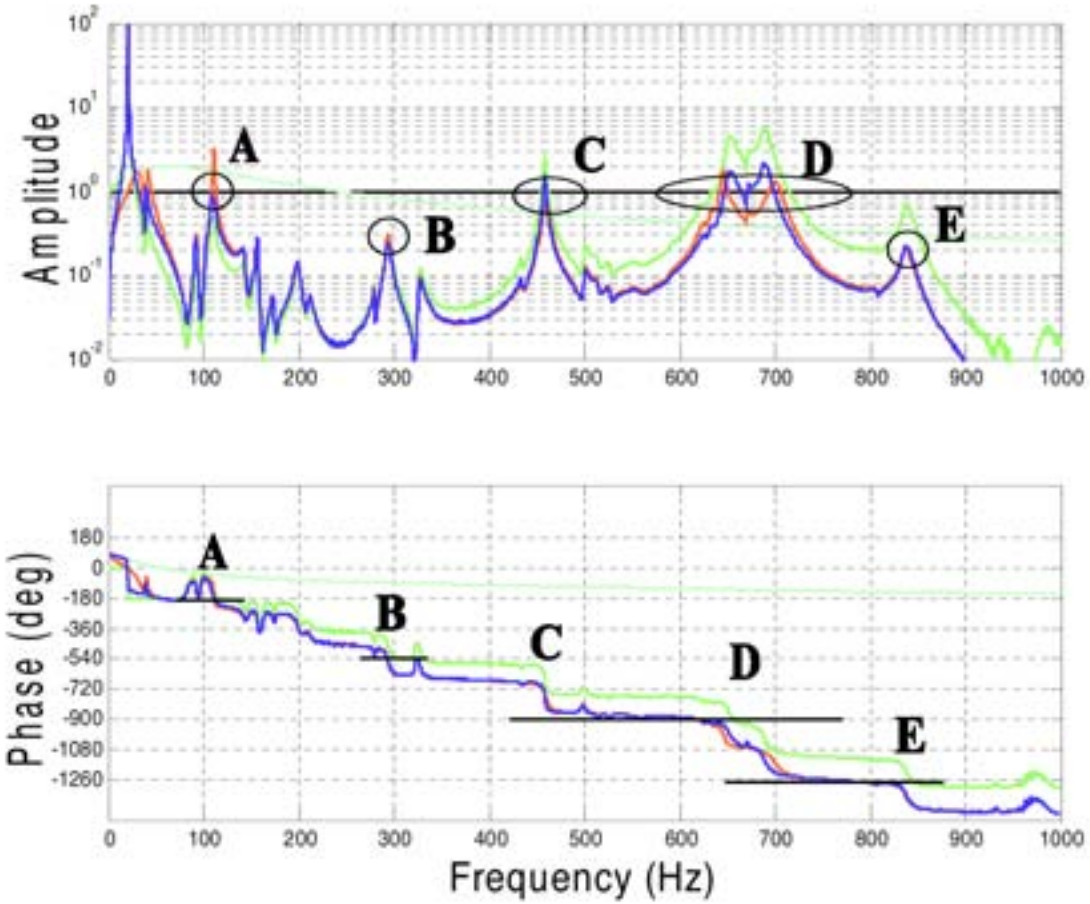
New Interferometer Calibration Fits & Residuals





Why Simple Feedback was Disappointing with Test Platform

If the "transfer function" (complex ratio of output to input) of the "plant" (platform motion over piezo motio) times the transfer function of the "controller" (piezo motion requested by combination of integral, proportional, derivative gains including any filtering, divided by platform motion) has magnitude > 1 at any frequency where the phase is $180^\circ + N \times 360^\circ$, the system will undergo growing oscillations at that frequency.



Blue is measured platform transfer function

Pale cyan is controller function at instability threshold

Red is product

Black bars are dangerous phases

A, D are borderline
B, C, and E are safe

Green is calculated closed-loop function

Why Simple Feedback was Disappointing with Test Platform (2)

Simple PID control of the platform was disappointing because the feedback phase that is appropriate to damp the fundamental resonance ends up anti-damping another resonance.

A low-pass-filter in the controller guarantees stability, but defeats the purpose.

Notch filters to reduce the amplitude of the controller function at the plant resonances help, and we used them. But such filters always introduce new phase shifts, and it becomes a balancing act to keep things stable.

What you want is an automated procedure that calculates the best controller given your measured plant response.

There is a method using “state vectors” that the textbooks claim to be “optimal.”

It pretty much requires a computer in the loop, although the programming isn’t hard.

State-vector control requires a very detailed description of the system. Getting this information is called “system identification,” and it’s acknowledged as something of a black art. There is commercial software (Matlab toolbox) that claims to do it, but it’s pretty simplistic.

The calculation of the optimal gains is straightforward but non-obvious and tedious. There is commercial software for that too (Matlab is the market leader).

State Vector Model

The state vector x_i is the set of numbers it takes to describe the system at time t_i.

The free evolution matrix A describes the free evolution during a single time step.

The control variable u_i is the command you give to the system at time t_i

The drive vector B describes the state-change induced by the control u_i.

The state at time t_{i+1} can be computed from the state at time t_i by

$$x_{i+1} = Ax_i + Bu_i$$

The observation vector C describes how the measurement y_i is related to the state x_i .

The measurement at time t_i can be computed from the state by

$$y_i = Cx_i$$

Note that we don't actually measure all the elements of the state x_i. We have to estimate them from measurements of y_i at different times.

Then we need to compute the control variable u_i from the estimated state x_i so the state is driven toward the desired state in an optimal way.

Estimator Gain and Control Gain

The measurement y_i has variance S. We can fit a sequence of measurements y_i to estimate the full state x_i. That estimated state has a covariance matrix, which gets smaller as more measurements are added.

The state vector also has a disturbance covariance matrix T which describes how random environmental effects change the state, relative to free evolution plus control of the state.

The estimator gain K describes how to update the estimate of the state x_i given the difference between the actual measurement y_i and the predicted measurement Cx_i.

$$\Delta x_i = K(y_i - Cx_i)$$

The famous Kalman Filter is simply the above equation using the optimal estimator gains, which are identical to what standard fit and error propagation formulas would give.

The control gain G is the vector to calculate the control variable from the state vector

$$u_i = Gx_i$$

We define a state penalty matrix Q and a control penalty R.

"Optimal" control gain means we minimize $V = \sum_i \left(x_i^T Q x_i + R u_i^2 \right)$

Using State Vector Control in Theory

We need to define the state vector x_i. For N masses coupled by springs in 1 dimension, the state vector could be the N positions and N velocities. The state vector could also be the sine and cosine amplitudes of the N normal modes.

We can put things like the setpoint, variables needed to compute filtered signals, or non-random environmental disturbances into the state vector too.

The measurement y_i and the control u_i are usually pretty obvious.

A, B, and C are constant vectors and matrices that you have to determine from a model of your system, or from actual data.

Your choose the state penalty matrix Q to define what you want to optimize. You can set the control penalty R to zero, or use it to compromise between saturating the actual control and quality of control.

Normally you know the measurement variance S. You have to determine the disturbance covariance T from data.

The optimal control gain G depends only on A, B, Q, and R. The optimal estimator gain K depends only on A, C, S, and T. There are well-known ways to calculate them, which take a bit of CPU time. Most people use Matlab, I've written my own.

The calculations inside the feedback loop are quite straightforward.

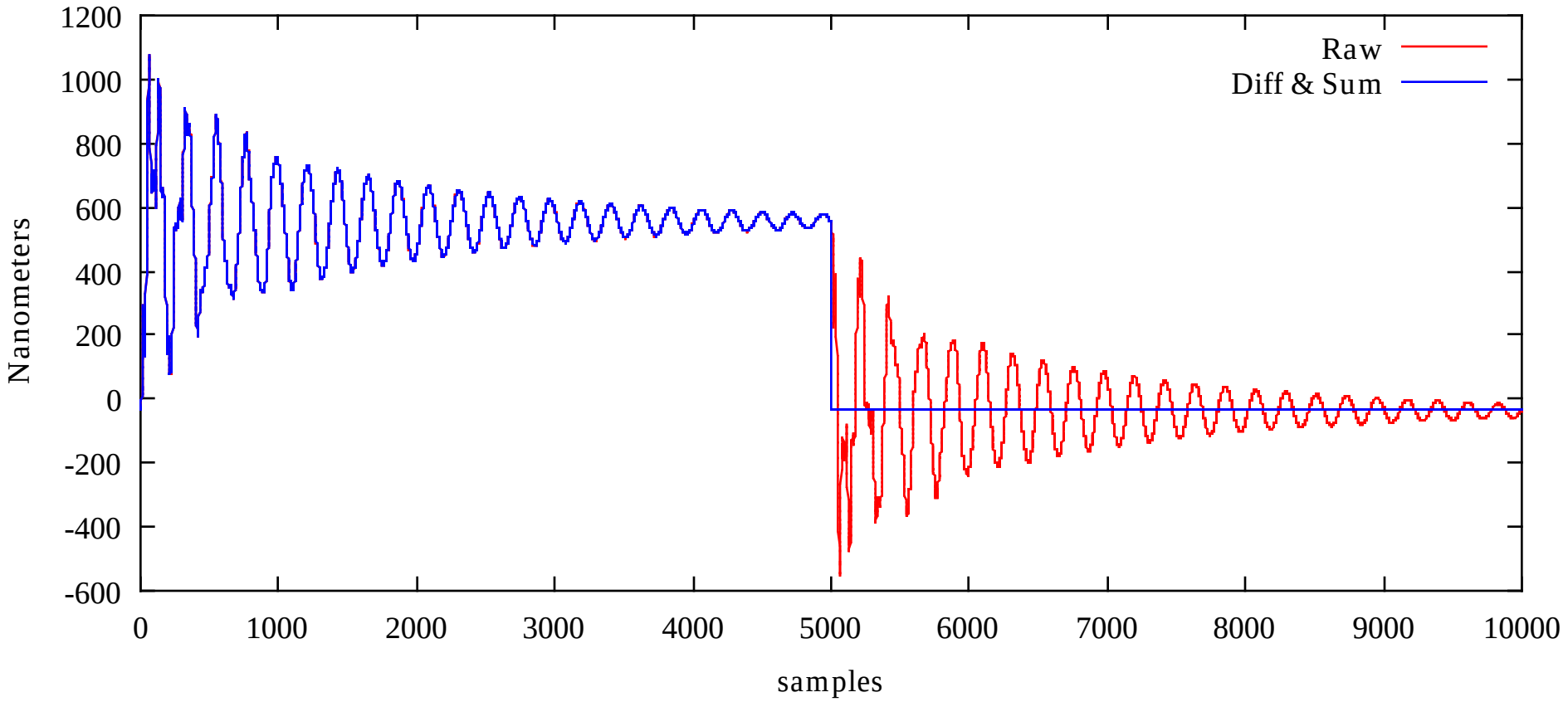
Using State Vector Control in Practice

0. Align and calibrate the interferometer
1. Determine internal modes of system (for A matrix and B vector)
   - A. Cycle the piezo with square wave, time-average the data to minimize disturbances
   - B. Fit to square-wave plus exponential piezo-creep terms
   - C. Analyze residuals with combination of FFT and time-domain fits to get mode info: frequency, damping, amplitude, phase
2. Determine disturbances to the system (for T matrix, may also augment A matrix)
   - A. Acquire interferometer data with piezo not moving
   - B. Define additional "modes" for external disturbances, append to A
   - C. Calculate disturbance-excitation of all modes, internal and external
3. Calculate optimal gains
   - A. Control gain G from A and B, plus user-input for Q and R
   - B. Estimator gain K from A, C, and T, plus user-input for measurement-error S.
   - C. Simulate measured system with measured disturbances and calculated gains
4. Do experiment on real system
   - A. Download system matrices and gains
   - B. Run estimator without control for 1000-10000 cycles as reference
   - C. Turn on control for another 1000-10000 cycles

All these steps are implemented on-line with graphical interface for the important parameters and graphical output for monitoring. The steps communicate via text-files that can be edited manually for things that are not well adapted to a graphical interface.

Piezo Square Wave (1A)

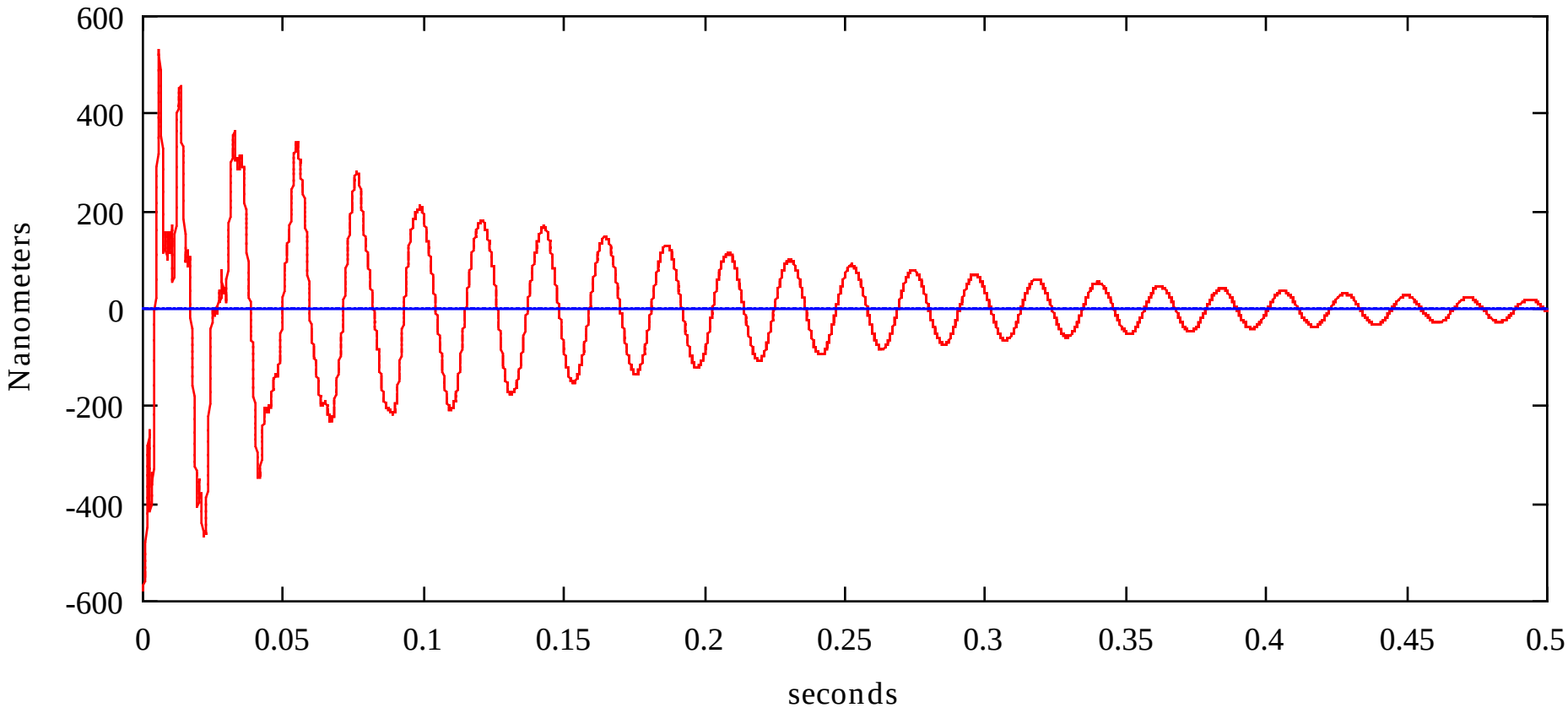
Drive the piezo with a square-wave with 1 Hz period, measure position at 10 kHz, and average thousands of cycles.



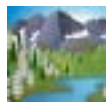
The platform overshoots, then rings at the fundamental damped resonance, with some other modes added. The second half-cycle is exactly the negative of the first, as expected.

Piezo Square Wave Fit Residuals (1B)

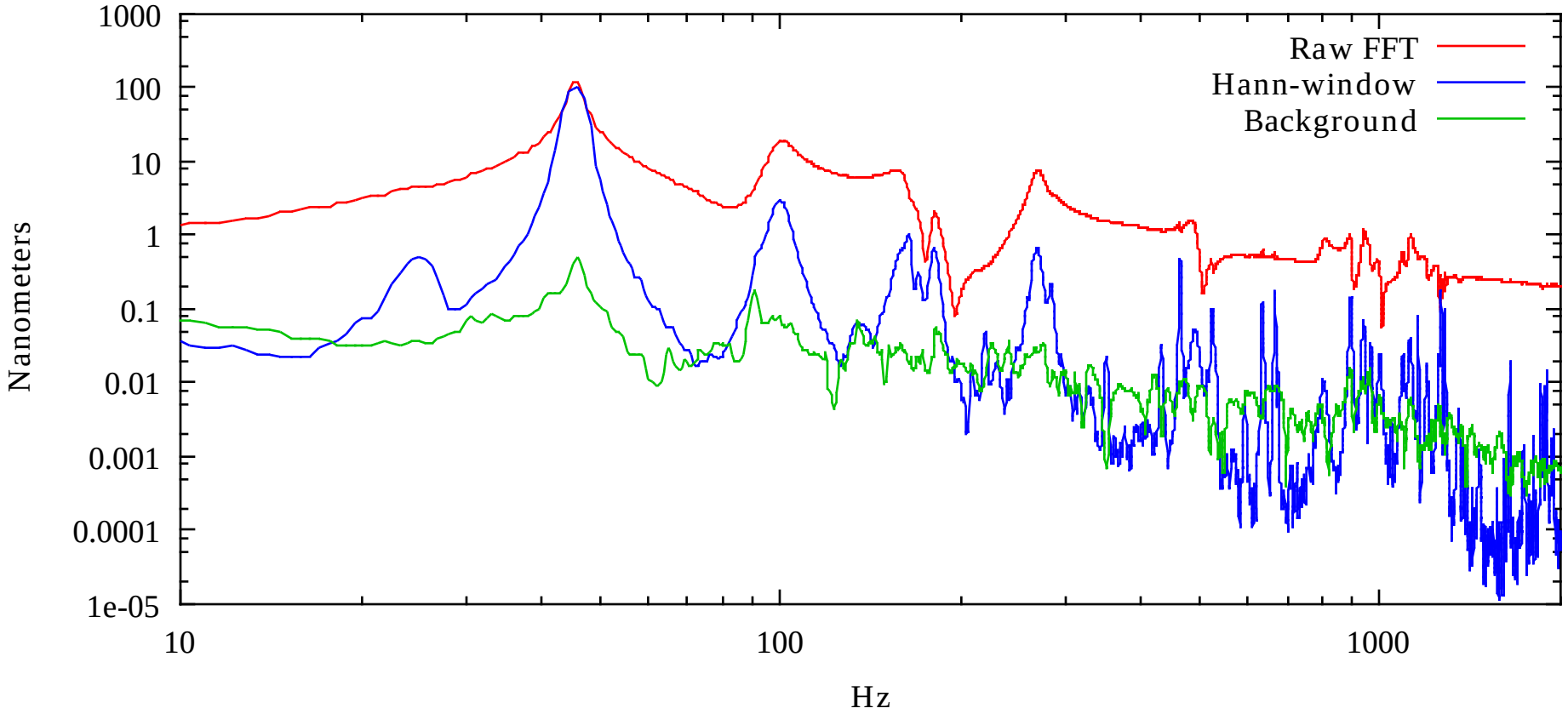
Fit data to offset square wave (plus decaying exponential for piezo creep)



The residuals show the long-lived fundamental mass-spring resonance, plus higher modes that damp quickly.

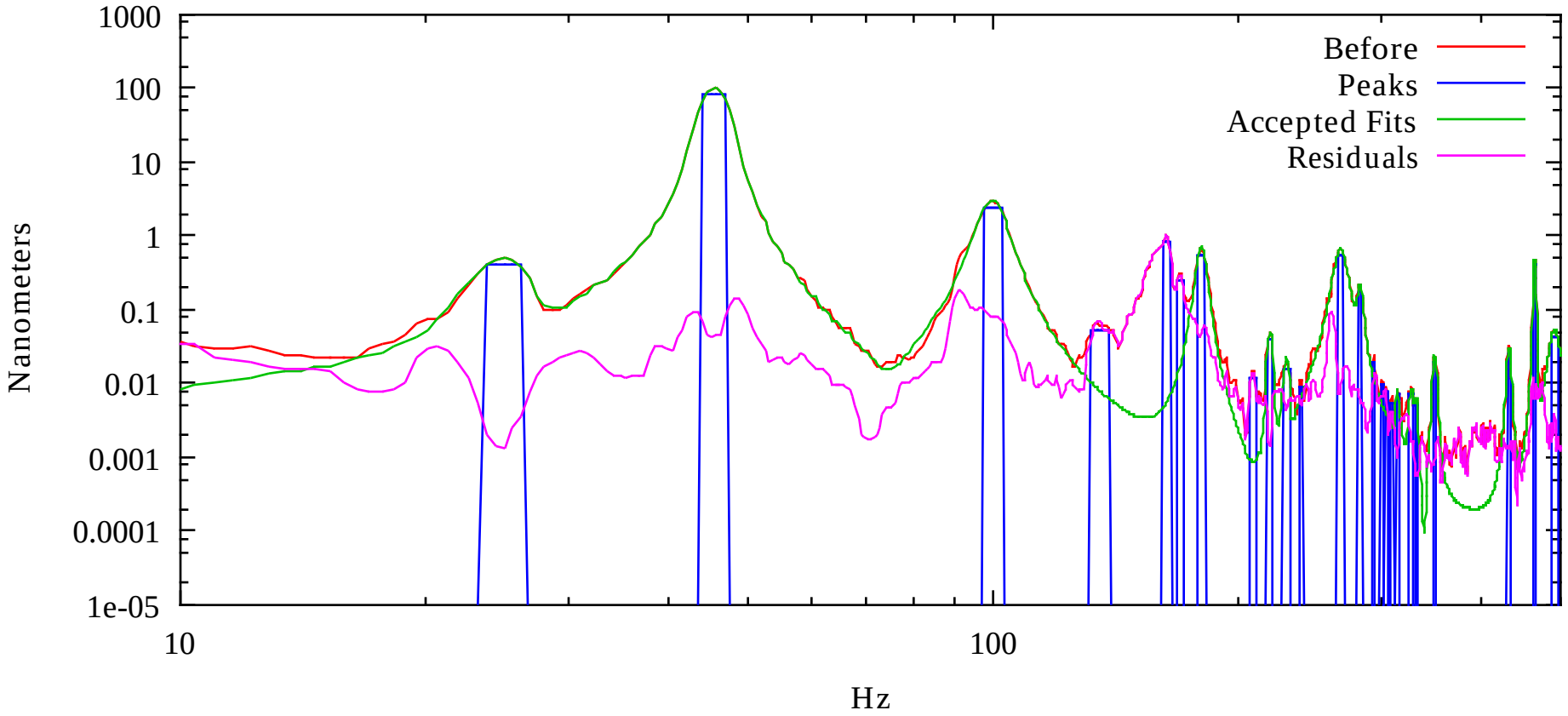


Fourier Transform of Residuals (1C)



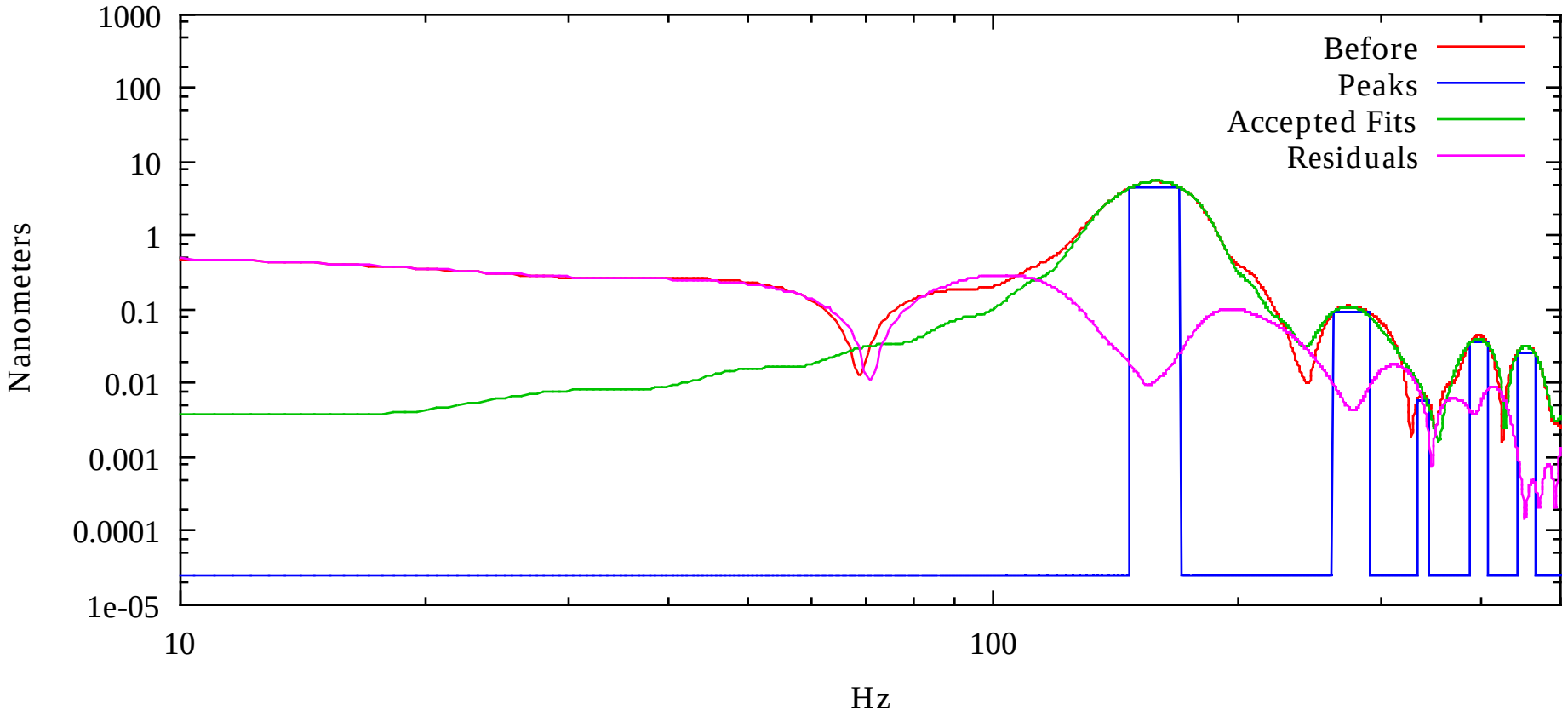
The data is zero-padded by a factor of 4 to improve frequency resolution, and local minima of the FFT have been replaced by the average of surrounding bins in 4 passes. The other resonances become much more clear if the data is multiplied by a Hann window function $1-\cos(2\pi t/T)$ before the FFT. The FFT of the background (sum of the first and second half-cycle) shows tiny traces of the fundamental and some other resonances, presumably excited by ground motion.

Find Peaks, Fit Modes (1C)



Find local maxima in the FFT, sort into amplitude order, fit the time-domain data with weights matching the Hann window function and fixed frequency and phase to get amplitude and exponential damping, refit with floating phase, floating frequency. A mode fit is rejected if the fitted frequency is outside original peak, or the amplitude of the FFT of the sum of all fits disagrees with the input FFT at the peak (which happened near 150 Hz for this data).

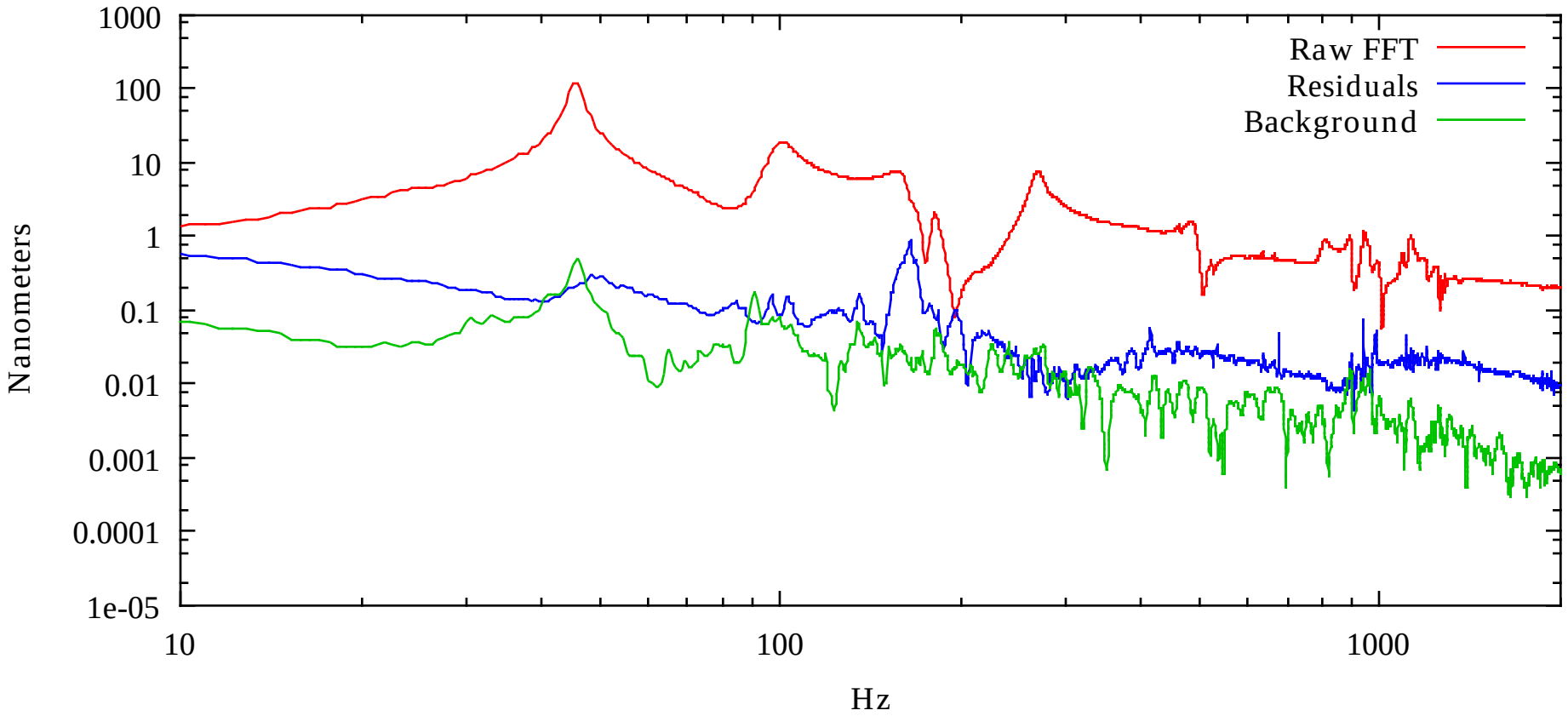
Zoom in on Early Times (1C)



Modes that decay rapidly are easier to find if only the early part of the (residual) data is used, with a narrow Hann window function. The FFT frequency resolution is worse, but a mode is found and fit nicely at 150 Hz, and some peaks that weren't even visible with the wide window are found and fit.

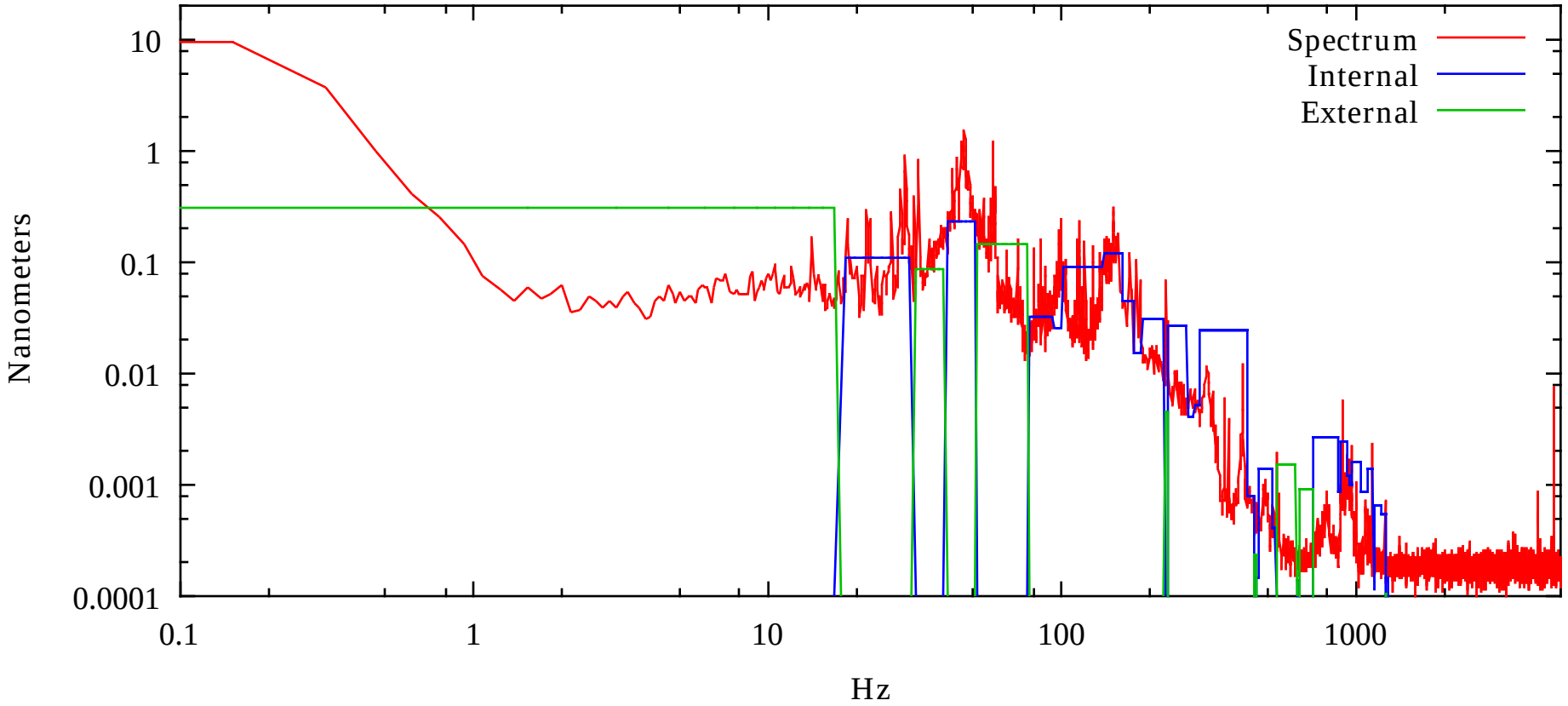
Results of Mode Finding (1C)

The fundamental mode is reduced by 3 orders of magnitude (down to background!), and nearly all modes are found.



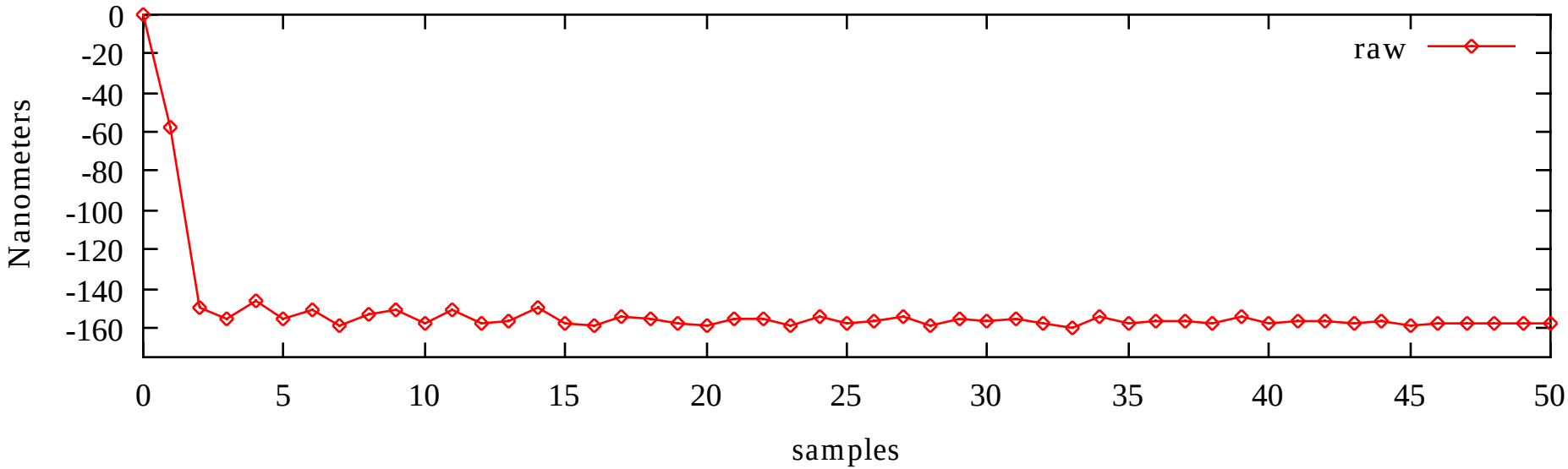
The largest problem is around 150 Hz. My guess is that there are really two modes, one short-lived (which was found and fit with the second, narrow window) and one long-lived (which was found with the wide window but the fit was rejected, probably due to confusion with the short-lived peak). Repeating the full-window peak find and fit on the final residuals would probably keep the long-lived mode that was rejected the first time.

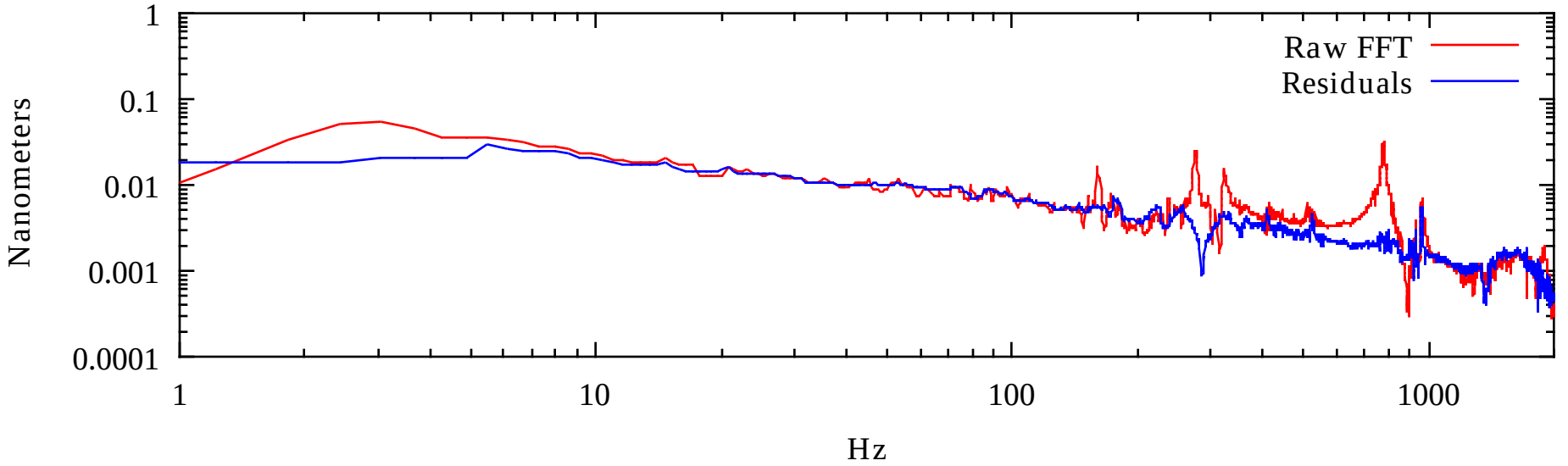
Disturbance Spectrum and Covariance (2)



Take long spectrum with piezo not driven, and FFT with Hann window. Divide into regions according to frequency and width of 25 internal modes with largest initial-amplitude / decay-time. If internal modes don't overlap, or if user adds band-division to file, add "external-mode" region. Find power-weighted mean frequency and RMS width of each region. Disturbance-covariance is integrated power times frequency-width for each region. Plot shows square-root disturbance covariance of internal and external mode regions. The largest disturbance is at the resonance, but many peaks are external.

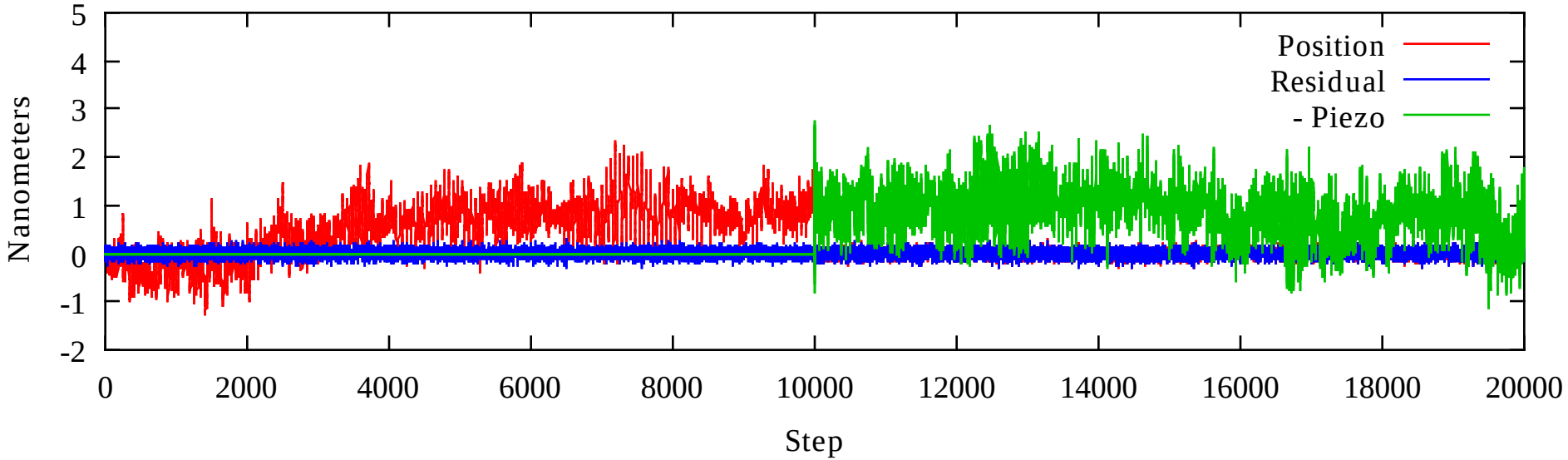
Mode Finding for Piezo-Mirror

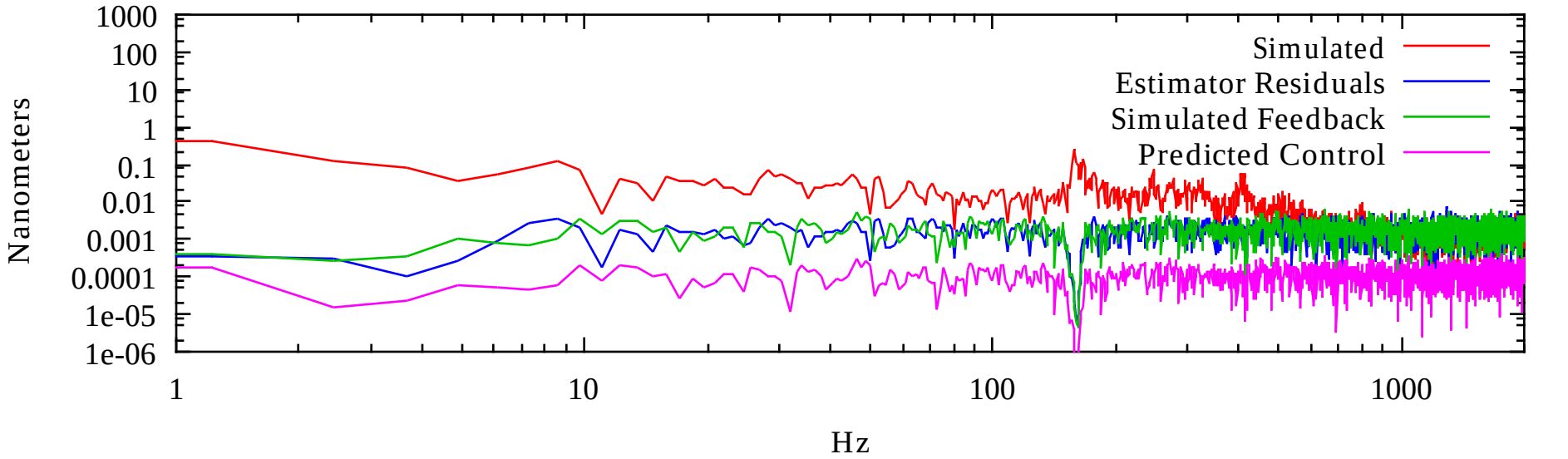




For this system, the major resonances are at much higher frequency, and are much lower in amplitude. The residuals of the first few steps are kept explicitly as part of A matrix.

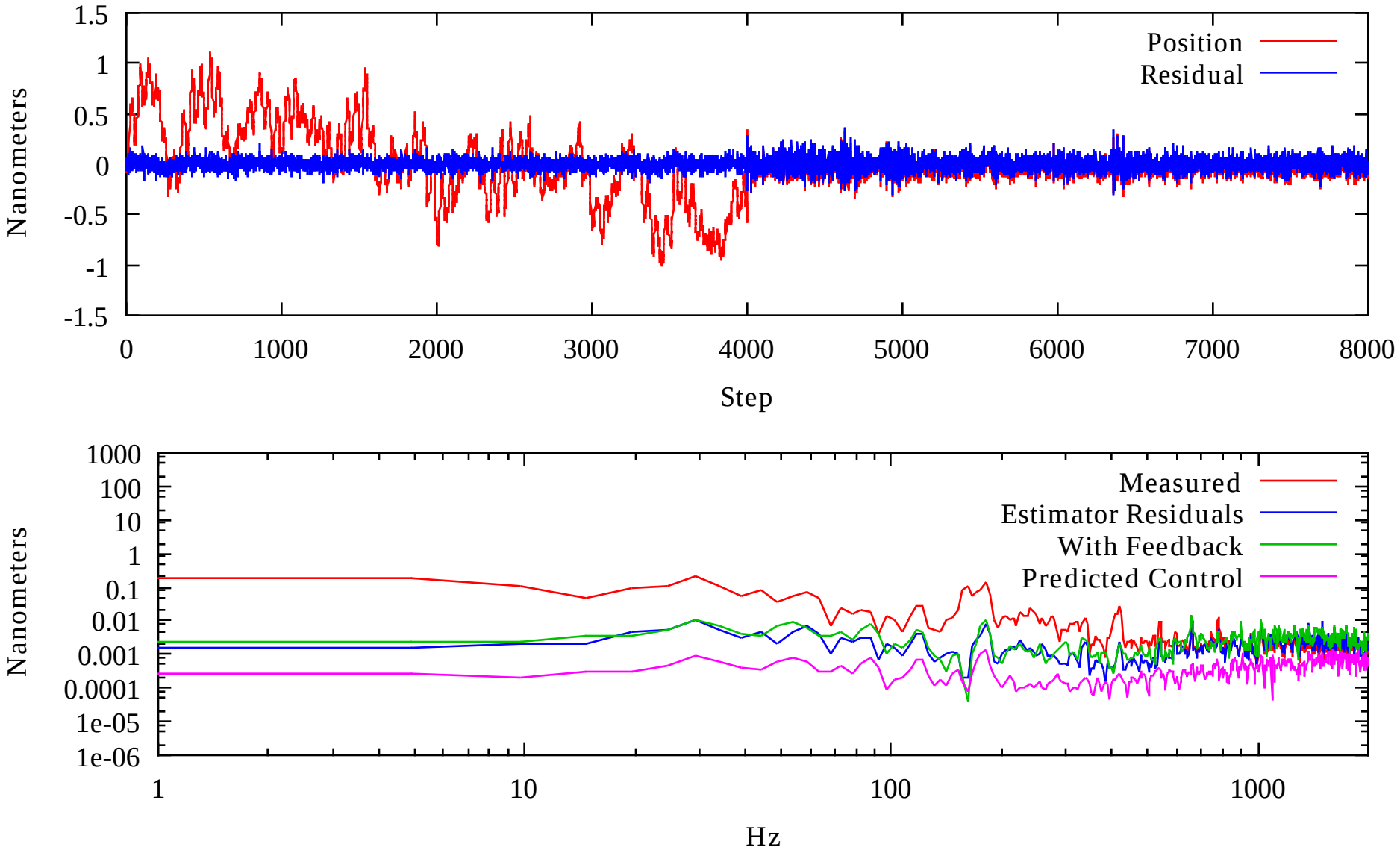
Feedback Simulation for Piezo-Mirror





The "predicted control" (as if estimated state were true state, and no disturbance or measurement error next step) is spectacular! The simulated feedback performance is as good as the estimator residuals.

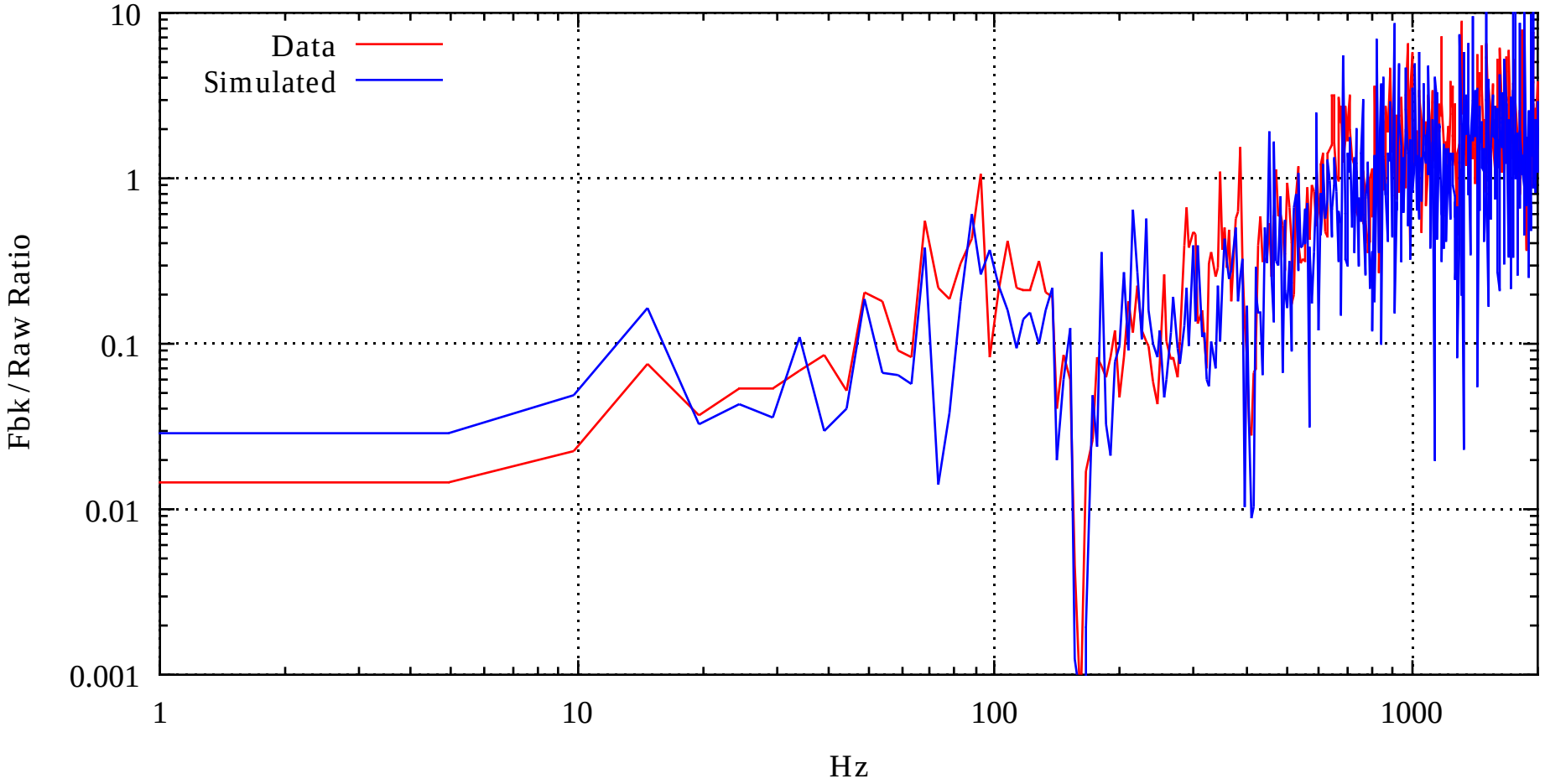
Feedback Data for Piezo-Mirror



Using the same gains with the real system, similar performance is achieved. It's not hard to tune PID feedback by hand to perform about as well. But this was an automated procedure of applying the theory to measurements of a real system.

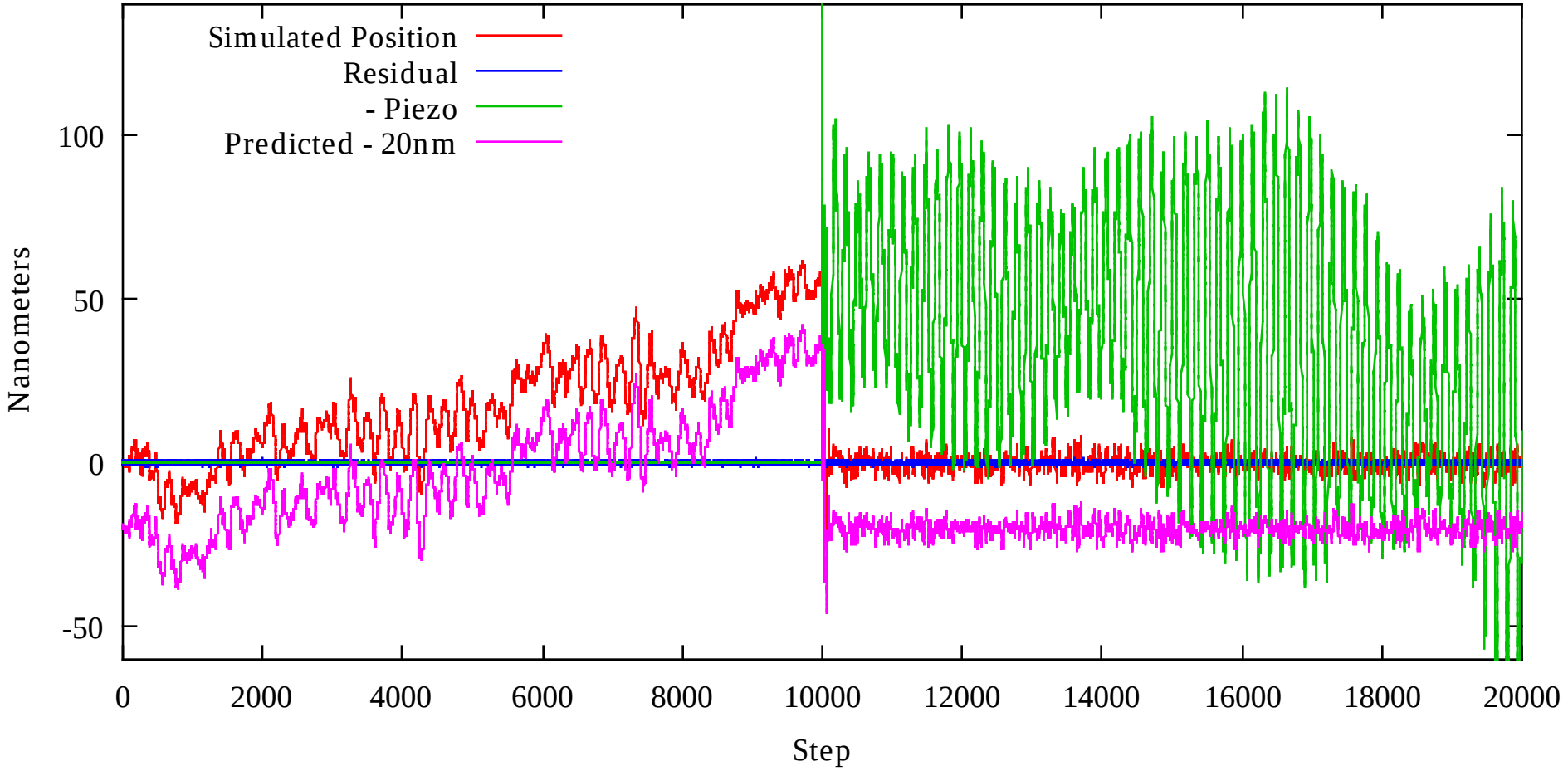
Piezo-Mirror Feedback Data vs Simulation

Data: RMS was 0.070 nm with feedback, 0.430 nm without, factor of 6.14 reduction
Simulation: RMS was 0.080 nm with feedback, 0.580 nm without, factor of 7.25 reduction



The agreement between data and simulation for the improvement due to feedback is pretty good. Even the small features are reproduced fairly well.

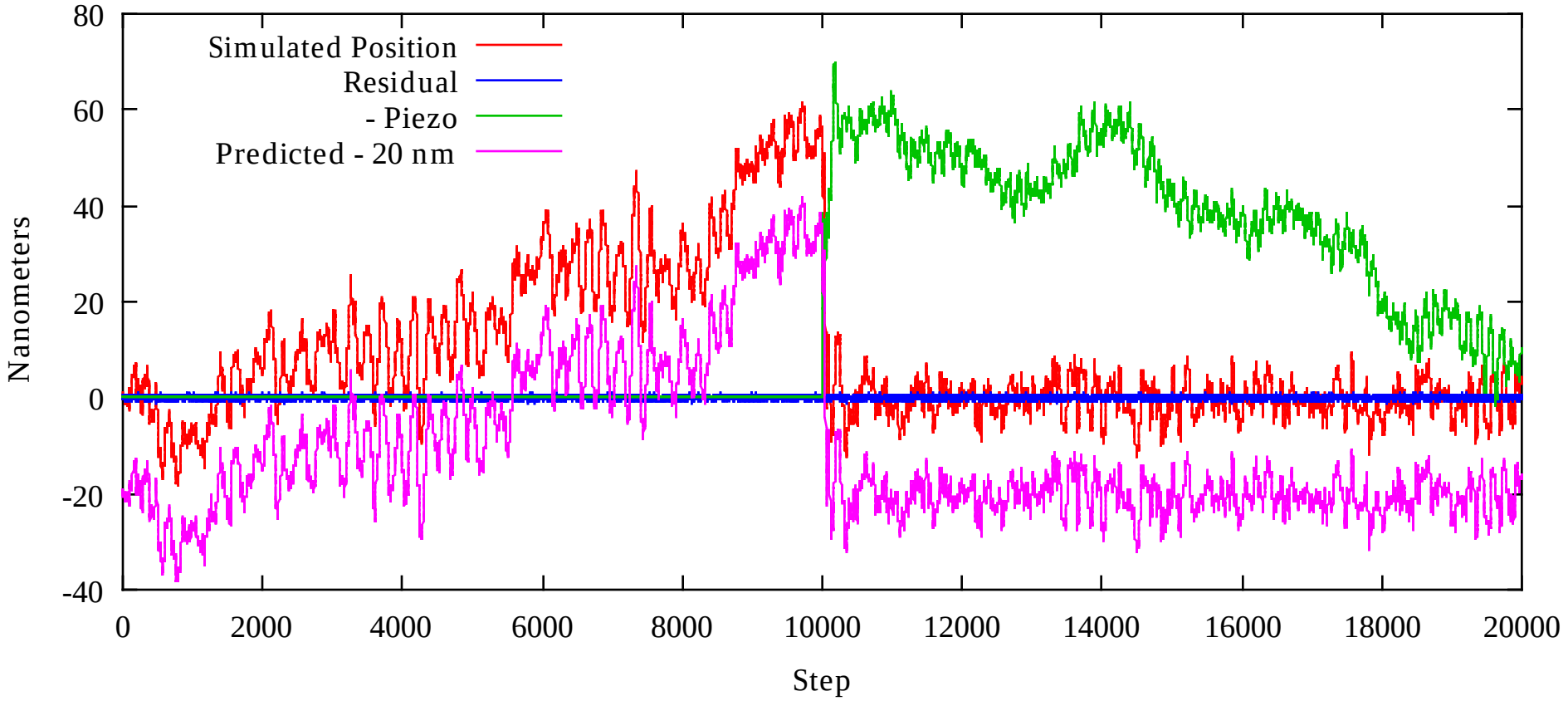
Feedback Simulation for 10 kg Platform



Residuals are OK (they look tiny because disturbance is larger than for piezo-mirror). And feedback is predicted to greatly improve things. But the piezo works very hard to achieve fairly unimpressive control. The predicted control is much worse than the estimator residuals, unlike the piezo-mirror case. Somehow the 10 kg platform system is just intrinsically more difficult to control than the piezo-mirror, even in simulation!

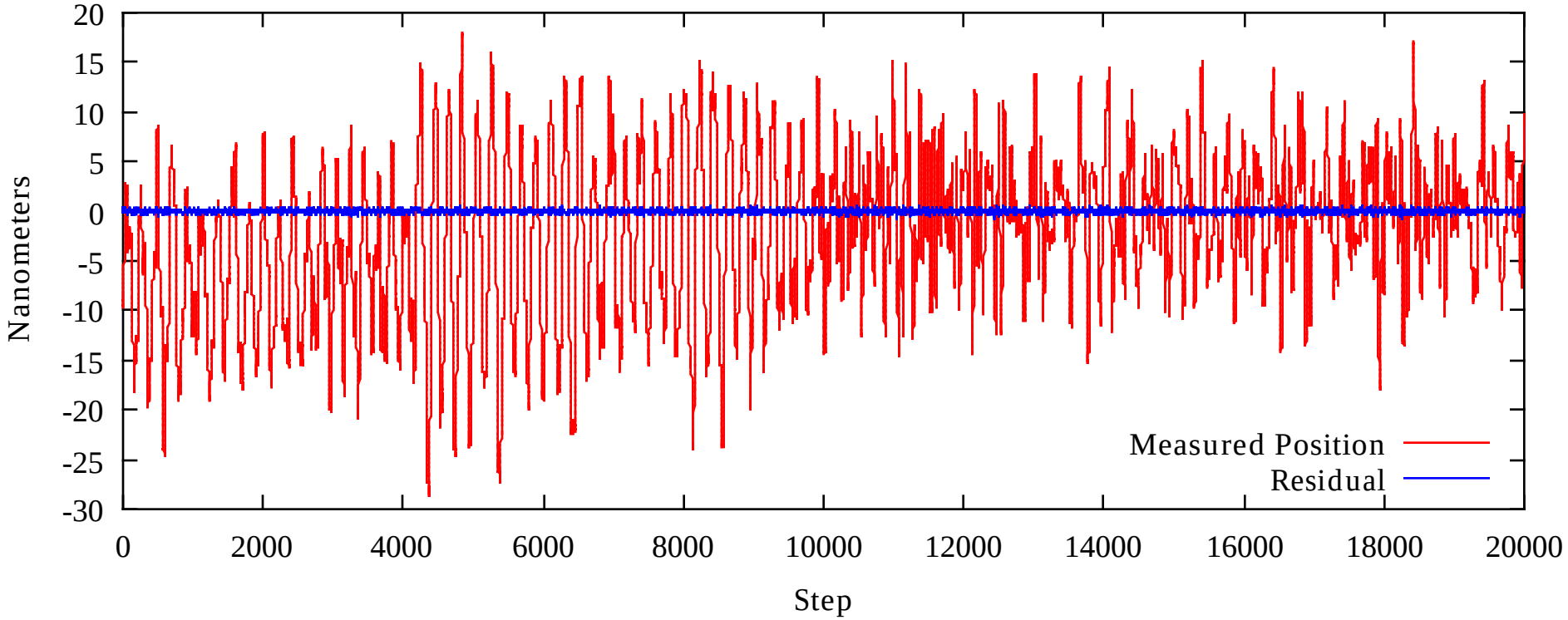
Vary Q and R to Rationalize Gains

Q was constructed so we were minimizing the total interferometer signal (the sum of all modes), but with only an infinitesimal penalty to ensure that the individual modes of the system were stable, and the control penalty R was zero. The gains (which were barely stable in simulation) were unstable for the real platform, where the model used in the gain calculation doesn't perfectly agree with reality. If we set R=5, and make the out-of-phase elements of Q equal 1, the simulated piezo is sane, with similar control quality (not great).



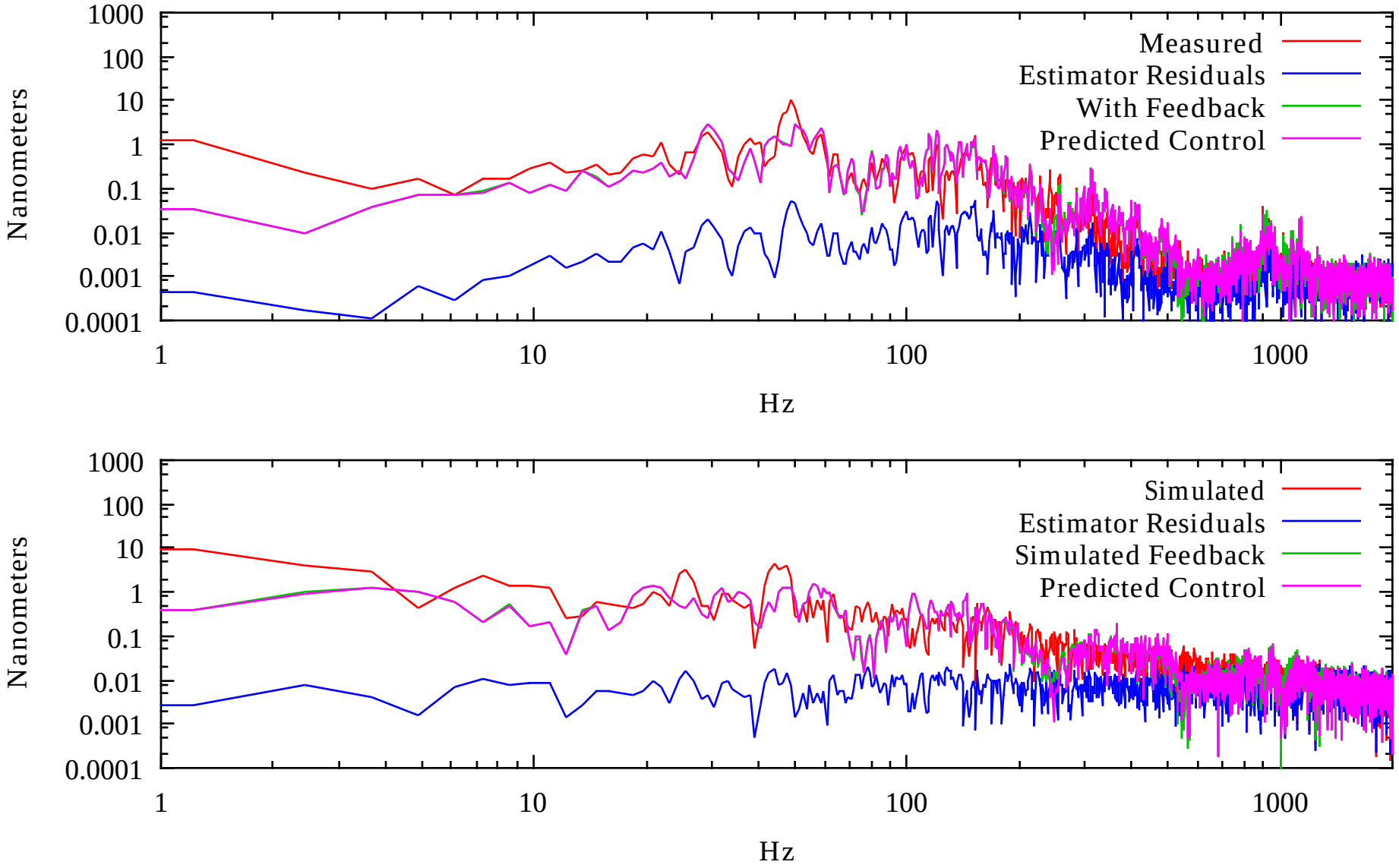
Feedback on 10 kg Platform

These gains are stable on the real platform. The estimator residuals are good: 0.150 nm ! The feedback clearly improves things somewhat, though it's not impressive to the eye...



But actually, the RMS with theoretically-optimized feedback is 5.8 nm, nearly the same as the 5 nm we achieved in the past with laboriously hand-tuned PID plus notch filters plus narrow-band external-mode feedback. However, the RMS before feedback was only 9.2 nm here, where it was 90 nm for the previous tests. Many details of the environment and the platform were different, in particular most of the 90 nm before was at low frequency which is easy to control away; there was less low-frequency disturbance here.

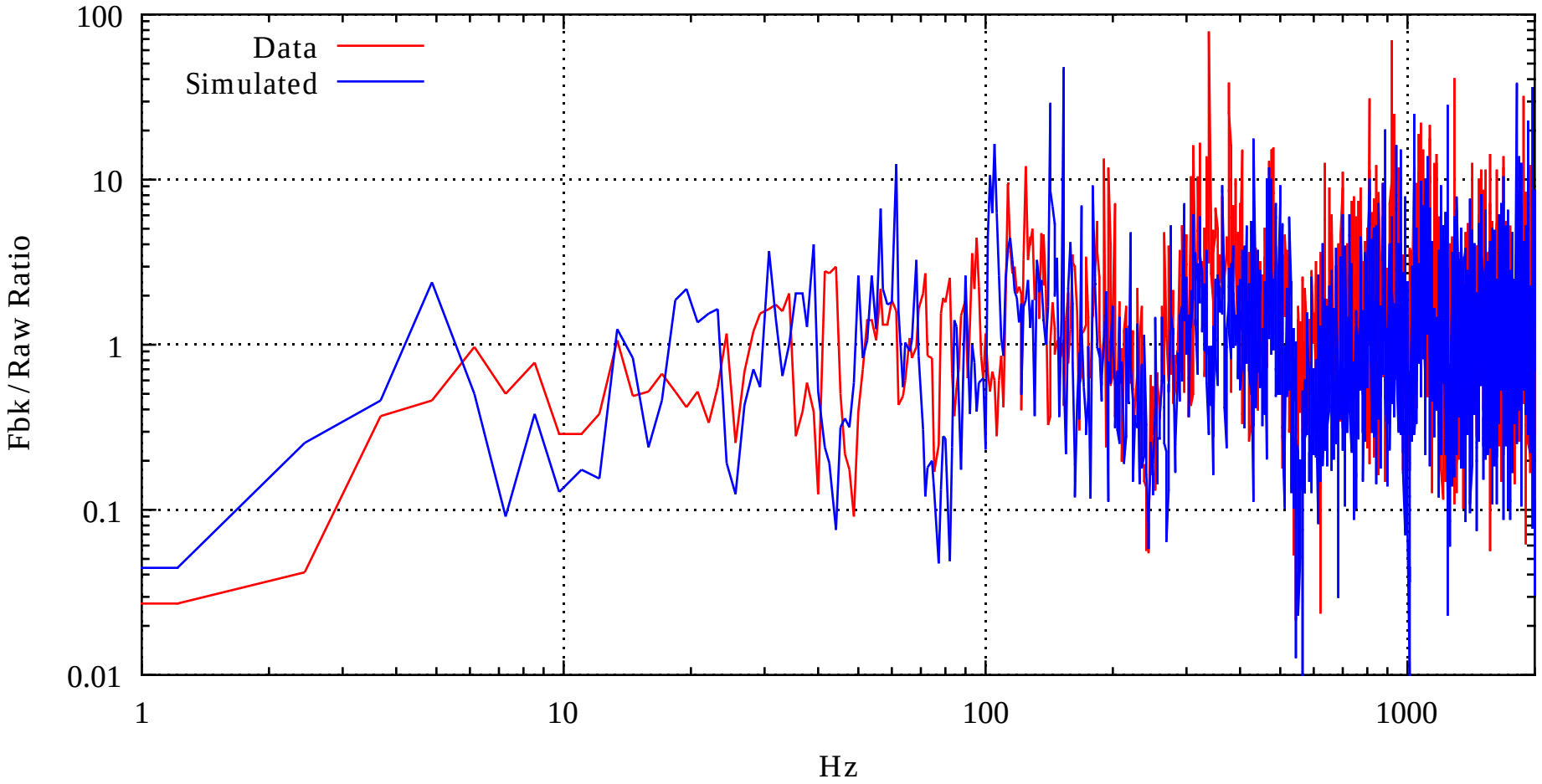
Data and Simulation Spectra for 10 kg Platform



In both cases, predicted control is much worse than residuals, and actual control is nearly identical to predicted control, nowhere near as good as residuals.

Data vs Simulation for 10 kg Platform

Data: RMS was 5.8 nm with feedback, 9.2 nm without, factor of 1.58 reduction
Simulation: RMS was 4.8 nm with feedback, 18.0 nm without, factor of 3.75 reduction



The improvement factor from feedback is good at low frequency, and similar for data and simulation, but not very impressive at and above the resonance frequency.

What Does It All Mean?

A system with lots of resonances will be hard to control at high frequency by canonical feedback methods, but I had hoped that the extra information that state-vector feedback has available would circumvent that.

But the hoped-for improvement in control of the 10 kg platform was not found.

The surprise was that even the simulation of the 10 kg platform (based on measurements of the real platform) didn't perform very well. That can't be blamed on the model not agreeing with reality. And the dilution of the gains required to make the real platform stable didn't degrade the simulated performance very much.

Apparently there is something about my platform that is intrinsically hard to control, presumably related to the higher modes and/or the slow response to the piezo.

I suspect that my reference-mirror mount is being excited indirectly by the piezo. If so, that's not an intrinsic problem with the concept, just my apparatus. Bolting down the base or using an isolated reference mirror may help a lot.

I'm going to try to invent a simulated system that has similar problems.

Then I'll try simulating a very stiff piezo between the "payload" and an added large reaction mass, with the soft spring between the reaction mass and other masses and springs (perhaps with an extra piezo to control the reaction mass).

Conclusions

A small and subtle interferometry calibration problem found, and fixed in an approximate way, with a better fix on the way.

Explained limits to canonical PID control for complex systems,
and introduced state-vector feedback

Methods to extract from measurements the information needed for state-vector feedback have been developed

Methods verified with piezo-mirror system, control to less than 0.1 nm, in good agreement with simulation.

Methods also work with 10 kg test platform (after adding minor control penalty and enforcing stability of individual modes), in reasonable agreement with simulation. Performance similar to best hand-tuned canonical control achieved.

It's not sufficient to have a sub-nanometer sensor (interferometer) and sub-nanometer actuator (piezo) running at 10 kHz with a very detailed model of the system being controlled. There can be details of the system that make it intrinsically hard to control.

I have a few ideas on what the problem with my platform might be, and on ways to make systems with similar problems more robust (reaction mass).

Credits

The work has been done by me and a succession of students, mostly undergrads

- Ken Yau, engineering-physics (computer) co-op student
- Jason Thompson, eng-phys (mechanical) co-op student
- Parry Fung, engineering-physics (computer) summer and undergrad thesis student
- Travis Downs, eng-phys (computer) NSERC summer student, honours phys thesis
- Michelle Chen eng-phys (computer) co-op student
- Mandy Wong, honours physics NSERC summer student
- Andrew Turner, eng-phys (computer) summer student
- Russ Greenall, physics M. Sc. student, thesis done
- Marc L'Heureux, physics summer student, just graduated