2020 IEEE Conference on Control Technology and Applications (CCTA)
August 24-26, 2020. Montréal, Canada

A Tutorial on PES Pareto Methods for Analysis of Noise Propagation in Feedback Loops

Daniel Y. Abramovitch*

Abstract— **This paper represents a tutorial on the so called PES Pareto methodology of analyzing the sources of noise in a feedback loop. Originally conceived for analyzing noise contributors in magnetic hard disk drives, the method provides a means of systematically identifying uncertainty contributors to a servo loop. Once identified and ranked according to their overall effect on the error and output signals, the top ranking sources can be worked on first, either by finding ways to reduce their magnitude or by altering system components to reduce sensitivity to the the noise contributors.**

The PES Pareto Method is based on three ideas: (1) an understanding of how Bode's Integral Theorem applies to servo system noise measurements, (2) a measurement methodology that allows for the isolation of individual noise sources, and (3) a system model that allows these sources to be recombined to simulate the servo loop's error signal. The method requires the measurement of frequency response functions and output power spectra for each servo system element. Each input noise spectrum can then be inferred and applied to the closed loop model to determine its effect on error signal uncertainty.

The PES Pareto Method is illustrated by decomposing PES signals that were obtained from a hard disk drive manufactured by Hewlett-Packard Company in the mid 1990s. However, beyond this old lab data, the method should be applicable to a wide variety of control loops.

I. Introduction

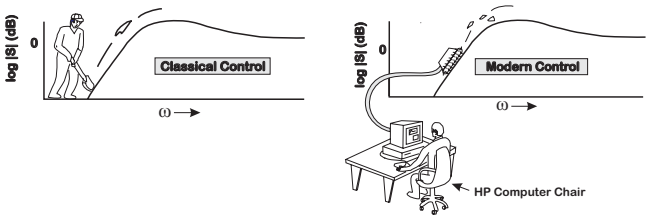

Fig. 1. Gunter Stein's dirt digging analogy, recreated from memory circa 1994.

The PES Pareto method arose out of trying to quantify the fundamental limits of position accuracy for hard disk drives (HDD) [1], [2], [3], [4]. The work, first internal to Hewlett-Packard, was published after HP exited the disk drive business. In the years that followed, it became apparent from the disk drive control sessions at the American Control Conferences in the early 2000s, that many of the disk drive manufacturers of the day had adopted this method for their work. In the years since then, the number of disk drive companies has reduced to three and the disk drive control sessions at ACC are only a fond memory. It is not clear if PES Pareto is still used in the remaining industry, but it seems like a good time to teach a new generation of control engineers how to apply it to their control loops.

*Daniel Y. Abramovitch is a system architect in the Mass Spectrometry Division, Agilent Technologies, 5301 Stevens Creek Blvd., M/S: 3U-DG, Santa Clara, CA 95051 USA, danny@agilent.com

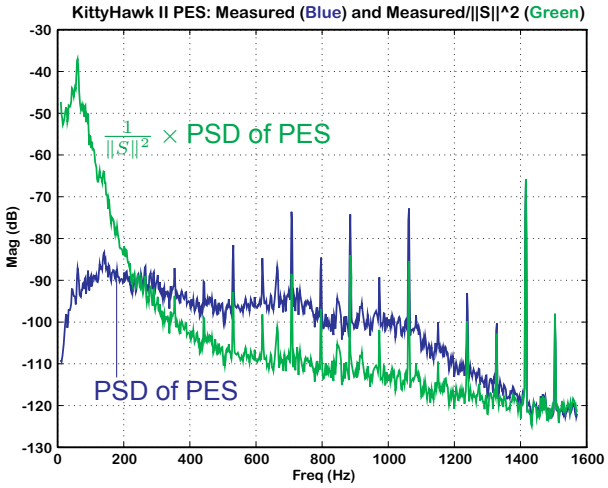

Fig. 2. HP KittyHawk 1.3" disk drive: PSD of PES, and PSD of PES filtered by $\frac{1}{\|S\|^2}$.

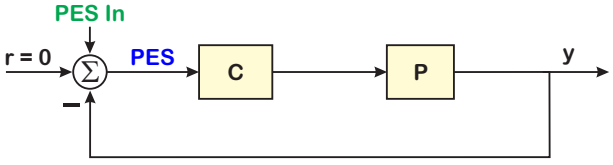

Fig. 3. Block diagram of original KittyHawk measurement that led to PES Pareto.

It is worth spending a moment to consider why the PES Pareto methodology was so quickly accepted by the disk drive industry while barely penetrating control practice anywhere else. Servo engineers in the disk drive industry had several common practices that made an intuitive understanding of PES Pareto easy for them, once it had been explained:

- They were used to making extensive, but not unified, measurements on physical systems. In fact, they had a variety of relatively expensive electronic test instruments in the lab and were well versed in making measurements in both time and frequency domain. What was far less common was a unified view of how to practically combine those measurements.

- They were used to working on difficult to control problems with severe limits on processing power, sample frequency, cost, and model completeness. This meant that they were receptive to the notion that there was no one "magic bullet" measurement method that would apply to all the different components around the feedback loop, and were ready to find a reasonable way to piece these together. Put another way, they had no issues with the "mixed metaphor" approach that physical system limits mandate and that PES Pareto embraces.
- They were acutely aware of noise and uncertainty limiting what they could do in their control designs, working against what they called a "noise budget." However, they were were flying blind as to where the noise was actually originating and how much any one source consumed of their noise budget.

It took this author years to realize that this kind of knowhow was not prevalent in the academic controls research community [5] and that the lack of these very utilitarian practices had likely prevented PES Pareto from filtering up to them and then out to their students. For this reason, this tutorial will teach far more about the pedantic intricacies of measurement between domains to give the reader the intuitive familiarity with what has to happen to assemble the measurements and models needed to get a much better picture of noise throughout the loop. It is hoped that by doing this, PES Pareto can become useful to a new generation of control systems engineers, far beyond what remains of the disk drive industry.

This paper will focus on understanding of the effects of noise through the feedback loop, or how to back noise measurements out to their sources. In the legendary "Respect the Unstable" Bode Lecture of 1989 [6], [7] Gunter Stein educated us to the idea that loops do not eliminate noise, they merely move it around, as he brilliantly illustrated with a dirt digging problem, reconstructed from memory in Figure 1.

The original measurement that was spawned by the persistent memory of Stein's talk (nine years before the paper [7] was published) was a disk drive problem measuring the Position Error Signal (PES) in HP's KittyHawk 1.3" disk drive in the early 1990s. That measurement is diagrammed in Figure 3 and the data displayed in Figure 2. The blue Position Error Signal (PES) was considered flat in the frequency domain by the servo engineers at HP's Disk Memory Division (DMD) in the early 1990s. Certainly, the blue measurement of PES in Figure 2 looks relatively flat across most frequencies. It was Stein's pile of closed-loop dirt that led to the realization that the blue curve was a closed-loop quantity, that had been filtered by the servo loop. What would that noise look like if it were a signal being injected into the loop, if it were the PES In signal?

The "Aha!" moment was realizing that the filtered, closed-loop PES signal could be quantified in frequency using its Power Spectral Density (PSD) and that PSD looked like what would happen if PES In had been filtered by the magnitude squared of the sensitivity function. If one had a model or measurement of that sensitivity function that covered the same frequency bins as the measurement of PES, one could do some math. Denoting the PSD of PES as Φ_{EE} and the PSD of PES In as Φ_{EE}, if

$$\Phi_{EE}(f) = \left\| \frac{1}{1+PC} \right\|^2 \Phi_{II}(f), \tag{1}$$

then

$$\Phi_{II}(f) = \left\| \frac{1+PC}{1} \right\|^2 \Phi_{EE}(f). \tag{2}$$

The resulting green curve in Figure 2 and was out of norm with what the engineers had been used to that it took a while to accept as correct. If the noise looked flat after being filtered by the sensitivity function (which has a lot of rejection at low frequency), then the unfiltered curve should have high levels at low frequency. Realizing this launched the PES Pareto methodology.

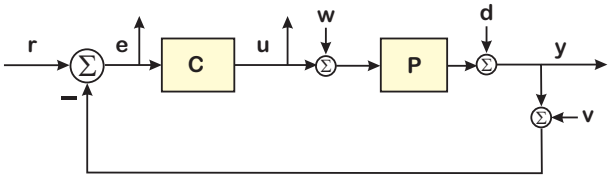

Fig. 4. Closed-loop system with a few more noise sources.

Even in such a simple diagram as Figure 3, it becomes a fundamental set of questions. If we are track following ($r = 0$), what does the noise in PES look like as the input PES In. We get that in Figure 2, and in the case of a disk drive, a lot of the broadband noise can be modeled to enter at that point, be in noise in the generation of the PES signal (called Position Sensing Noise, PSN) or air flow over the disk drive heads buffeting the physical position around. However, even in a disk drive, there are far more noise injection points around the loop, as diagrammed in Figure 4. If $\Phi_{EE}(f)$ is flat, or no matter what shape it takes, what do the input PSDs of $\Phi_{WW}(f)$ and $\Phi_{VV}(f)$ look like? Furthermore, how do we quantify how much of $\Phi_{EE}(f)$ is due to one versus the other? It is clear that without any other knowledge, it is hard to separate out the effects of $\Phi_{WW}(f)$ versus $\Phi_{VV}(f)$.

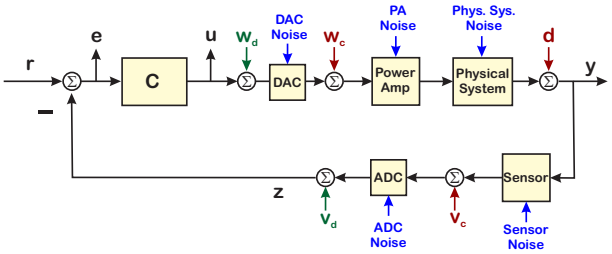

Fig. 5. Closed-loop system with each block having its own noise source.

In a more complex model where each block can be modeled to have its own noise source (Figure 5), how do we:

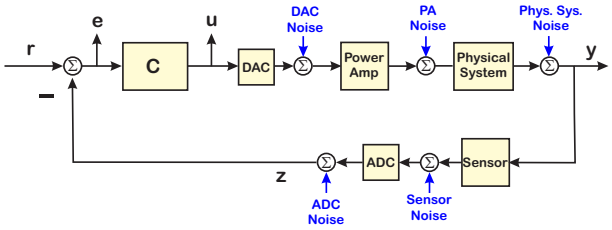

Fig. 6. Simplification for analysis of Figure 5. Closed-loop system with each block having its own noise source as an additive output noise.

- Isolate and measure some noise source at some downstream output or measurement point?
- Back up through whatever effective filter there is to get to the particular noise source as an input?
- Push that source (and others) forward through the closed-loop system to see the effects of that noise source on the rest of the loop?

The second two questions are answered by setting up the math:

- We need Power Spectral Densities (PSDs) in a measurement frequency range with consistent frequency bins.
- We need measurements and/or models of all the blocks in such a way that we can match the frequency bins.
- We need to understand the relationship between physical PSDs, the integral across the frequency band, and (via Parseval's Theorem [8]) the noise variance in time, σ^2.

The first question involves a lot of hands-on cleverness and some fudging, but it is worth it.

- We see right away that in order to isolate some noise sources, we need to open the loop.
- In other cases, we cannot make the measurements without the system being in closed loop.
- Some noises are arrived at when we channel Sherlock Holmes and eliminate all the others as a4 potential source [9].

The model of Figure 5 can become intractable for our analysis, unless we can simplify it to that of Figure 6, in which we have taken the block noises and modeled them as individual output noises of the blocks. Furthermore, we have made a reasonable assumption that we can consider the digital noises (apart from quantization) to be negligible, at least in the sense that they can be mitigated via numerical stability methods. In that respect, they are part of any reasonable controller design. Figure 6 will serve as a prototype for the type of block diagram that best fits into the PES Pareto environment.

Once we have these noise "sources" and their effect on the loop, we can model the effects of changing one of those sources. This measurement based modeling tells us where to put our system design and control effort. The achievable bandwidth of even the best control algorithm is eventually limited by time delay and noise on a sensor that the loop amplifies at high frequency [5]. In such cases, attention to the sensors, to the critical and dominant noise sources, and perhaps to how to demodulate signals [10], often enables a much higher performance increase than the 10% improvement achieved through 20× the math in the controller.

Finally, it is worth understanding why we focus on broadband noise. In the case of a hard disk drive, the error signal, called the Position Error Signal (PES), can be decomposed in the frequency domain into four components:

External Shock and Vibrations are heavily influenced by the drive's operating environment. It has been shown that accelerometer feedforward control can considerably reduce the effect on PES [11], [12].

Synchronous or Repeatable Excitations are due to the rotation of the spindle and therefore synchronous with it or one of the spindle orders. While synchronous excitations may be large, standard practice in the disk drive industry includes using feedforward cancelers to reduce the effects of synchronous excitations [13], [14]. More importantly in the hard disk industry, the move of drives from the office into the living room meant that the audio noise caused by the ball bearing spindles were disturbing enough to people viewing their recorded movies that the ball bearing spindles quickly were replaced with fluid bearing spindles [15]. In recent years, feedforward has seen a dramatic increase in popularity with its use in nanopositioning control loops such as atomic force microscopes [16], [17], [18] as well as macro-positioning control such as wind turbines [19].

Non-synchronous or Non-repeatable Excitations include sharp spectral peaks due to spindle bearing cage orders and structural resonances (which are less sharp but still narrow band). There was work suggesting that disturbances due to resonances or cage orders could be considerably reduced by the use of damped disk substrates and fluid bearing spindles [20], [21], [22]. As mentioned above, this happened for other reasons. In applications beyond rotating machinery, these are also minimized via physical redesign. Alternately, loop gain can be raised over a narrow portion of the pass band with a bump filter, or dropped outside the passband). In these cases, if the band is narrow enough, the area of noise amplification can be kept relatively small.

Broadband or Baseline Noise is what remains when all of the narrow band components have been removed. Of the four categories, baseline noise has received the least attention in the literature. Therefore, broadband noise became the focus of PES Pareto. Broadband noise was not viewed as something that could be managed via repetitive or other feedforward control methods. By its very nature, it could not be minimized with narrowband filters. When combined with Stein's revelation about Bode's Integral Theorem, understanding the effects broadband noise gained much higher priority.

Consider that if signals including noise are filtered by closed-loop (CL) dynamics to get to PES (the error signal), then inverse filtering by closed-loop filter dynamics should give us a reference or noise input. How do we filter noise? It is better to ask what kind of noise can be analyzed through a filter? The answer is additive, white, Gaussian

noise (AWGN). AWGN has the property that auto and cross spectra can be analyzed when passed through a linear filter [23]. This means that if we can generate frequency responses for our closed loop dynamics, generate magnitude squared filters, and invert those, we can back out noise sources. We need to keep in mind that each source injection point has its own back filter from the measurement point, and each measurement location has its own forward filter from any injection point.

This became the basis for the PES Pareto methodology of analyzing the effects of noise on a system [1], [2], [3], [4]. While that work was focused on the control of hard and optical disk drives, this tutorial will attempt to give participants a view to applying this method to all manner of control problems.

The structure of this tutorial is as follows. Section II will review how Bode's Integral Theorem and its discrete-time counterpart motivate the study of noise passing through a feedback loop and outline the methods for doing this. Section III will establish a common mathematical framework that will motivate and underpin all of our measurements and the rest of our analysis. Section IV will provide an old but real example system, illustrates the methods. Section V will discuss the "Simple Tricks and Nonsense" [24] needed to get actual noise measurements out of a physical system, using the two examples from Section IV. Section VI will illustrate backing closed-loop noise measurements back to their open-loop sources, and then propagating them forward to the loop error signal so that they can be ranked by their effects on that error. Section VII will present some examples of how these new measures can be used to evaluate the effects of a particular noise source increasing or being minimized. Finally, Section VIII will discuss methods of minimizing the effect of noise sources before they enter the feedback loop [10].

II. Bode's Theorem and Noise Shaping

Bode's Integral Theorem [25] deals with what Bode calls regeneration, and dates back to the 1940s. In the years since Stein's Bode Lecture [6], it has gained prominence in the controls community, as this theorem has significant implications for and applications to control systems.

The sensitivity function, S, is also known as the disturbance rejection function because it shows how disturbances, d, go through the system and show up at the output, y, or at the error signal e.

$$S = \frac{e}{r} = \frac{1}{1+PC} = \frac{y}{d} = -\frac{e}{d}. \tag{3}$$

While the mathematics used to prove both versions of Bode's Theorem can be fairly complicated, the result is fairly simple and extremely powerful. We will leave the proofs to the references [25], [26] and talk simply about the interpretation. Looking at Figure 7 it says simply that:

$$\begin{array}{c}\text{the area of}\\ \text{disturbance}\\ \text{amplification}\end{array} = \begin{array}{c}\text{the area of}\\ \text{disturbance}\\ \text{rejection}\end{array} + \begin{array}{c}\text{a non-}\\ \text{negative}\\ \text{constant.}\end{array} \tag{4}$$

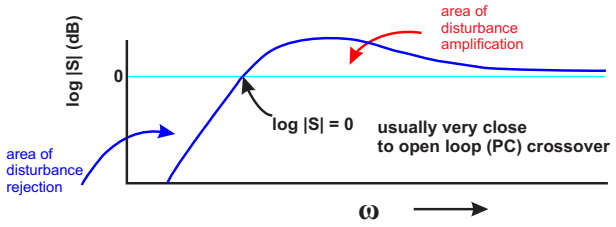

Fig. 7. Sensitivity function.

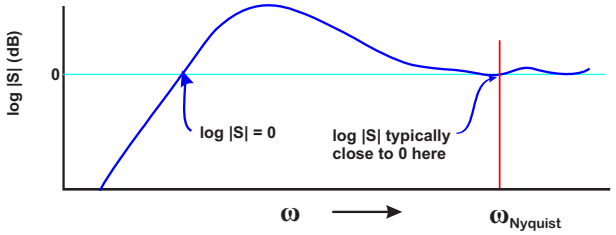

Fig. 8. Sensitivity function in discrete time.

Mathematically, this is stated as

$$\int_0^\infty \log|S| d\omega = c, \tag{5}$$

where c is some positive constant dependent only on the open loop unstable poles and non-minimum phase zeros.

Consequences: "Sooner or later, you must answer for every good deed." (Eli Wallach in the *The Magnificent Seven*)

Translation: If you make the system less sensitive to noise at some frequencies, you then make the system more sensitive at other frequencies.

Typical control designs attempt to spread the increased sensitivity (noise amplification) over the high frequencies where noise and/or disturbances may be less of an issue. The image of this was provided in the Bode Lecture at the 1989 IEEE Conference on Decision and Control (Tampa, FL)[6]. The talk, by then Honeywell Researcher and MIT Professor, Gunter Stein, was entitled "Respect the Unstable." Stein described the net effect of control systems design as trying to get a certain amount of disturbance rejection at some frequency span while trying to thinly spread the amplification over a large frequency span. Stein's drawing had a guy shoveling disturbance amplification "dirt" as in Figure 1. The dirt can be moved, but not eliminated. Furthermore, the discrete-time version of Bode's Integral Theorem [26] has some implications for discrete time systems [2], however they are essentially those of the continuous time theorem with the Nyquist rate forming a retaining wall for the disturbance amplification dirt 8.

There are two reasons why Bode's Integral Theorem is important in a discussion of a feedback loop's error signal. First of all, it gives a very good gauge on what can and cannot be done with disturbance rejection and noise in a control system. An intelligently designed control system puts noise

amplification in places where there is little noise. A poorly designed system results in significant noise amplification.

The second reason will become apparent in Section V on making measurements. It turns out that when the error signal is measured from a closed loop system, the loop should actually be opened up to look at the error's contributing sources. The *exact same effects* that are the point of the above theorem affect a measurement of error in closed-loop. Before we can do any of this, we need to establish a common mathematical framework that will motivate and underpin all of our measurements and the rest of our analysis. Basically, we need the machinery to do what was discussed above. This will be in Section III.

III. Noise Analysis and PSDs

The PES Pareto methodology requires measuring, averaging, and isolating the spectra of signals at different points in a feedback loop and then filtering them in different ways to get to input noises and output noises. It goes without saying that all of the operations need to be done on the same frequency axis. Whether a spectrum is measured using a spectrum analyzer, or generated from a time trace, the frequency bins (width, count, and location) have to be the same. The models of the different loop components that "filter" the spectra must also have the same frequency axis so that the filtering can be done bin-by-bin. This requires some rethinking of our measurements. Frequency response functions are often measured on a logarithmic frequency scale since control engineers are used to plotting frequency response functions using logarithmic frequency. Spectrum analyzers typically generate spectra on a linear frequency axis, so our frequency response functions (even if we are generating them from an analytic model) really need to be generated on this same linear frequency axis. Calculating spectra from time measurements – usually via FFT calculations – also are usually done on a linear frequency scale. Generally, the most reasonable way to do this is to generate an analytic model of the different loop components from our measurements. That analytic model can then be evaluated at the frequency points of the spectrum measurements. This brings in all the difficulties of extracting models from measurements [5], [27], but is a needed step.

A. Useful PSDs from Measurements

Our goal in this section is to describe how to measure Power Spectral Densities, either directly from a spectrum analyzer, or from one or more time domain measurements that have been transformed via a Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT). As mentioned above, we need these spectra measurements to have the same number and distribution of bins in the frequency domain. The easiest way to get a spectrum measurement of a signal is with a spectrum analyzer, an instrument that does this computation automatically . Spectrum analyzers are nice because they essentially package all the necessary computations. The modern ones all compute FFTs [28], [29], while some of the older ones computed a spectrum by sweeping an narrow band Gaussian filter across the frequency range and simply using the output of that filter as the spectral value at that frequency point [30].

The issue with any spectrum analyzer is that they have a fixed sample frequency and typically have a fixed number of frequency bins at which they evaluate the spectrum. It is also often difficult to extract from their documentation exactly what scaling is used in their FFT calculations, as the equations for computing FFTs are scaled differently in different devices. Let's get back to basics so we can have a unified understanding of all these measurements.

The Fourier Transform of a signal $x(t)$ is defined as

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt, \tag{6}$$

while the inverse Fourier Transform becomes

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df. \tag{7}$$

In some cases, the authors use ω in place of f. This doesn't affect the first integral (which is over the time variable, t)

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt, \tag{8}$$

but in integrating over ω in place of f we need to factor out the 2π, so the inverse Fourier Transform becomes

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(f)e^{-j\omega t}d\omega. \tag{9}$$

If only a finite data record of time length T exists then the finite length Fourier Transform is

$$X(f,T) = \int_{0}^{T} x(t)e^{-j2\pi ft}dt. \tag{10}$$

We should note that for any practical measurement, only a finite data record of time length T will ever exist, so to apply Fourier Transforms in real life, we need to make use of Equation 10.

If that signal is sampled with a sampling period of Δt then the sequence that results is

$$x_n = x(n\Delta t) \qquad n = 0, 1, 2, \ldots N-1 \tag{11}$$

and Equation 10 can be recast as the discrete Fourier Transform:

$$X(f,T) = \Delta t \sum_{n=0}^{N-1} x(n)e^{-j2\pi fn\Delta t}. \tag{12}$$

By letting $f_k = \frac{k}{T} = \frac{k}{N\Delta t}$ in Equation 12 we get

$$X(k) = \frac{X(f_k)}{\Delta t} = \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi nk}{N}}. \tag{13}$$

Let $W_N = e^{\frac{-j2\pi}{N}}$ and $\tilde{W}_N(u) = e^{\frac{-j2\pi u}{N}}$. Then Equation 13 can be written as

$$X_k = X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} = \sum_{n=0}^{N-1} x(n)\tilde{W}_N(kn). \tag{14}$$

This is what a standard FFT, including the one in Matlab computes. Note that

$$X(f_k) = \Delta t X(k) \tag{15}$$

which returns the FFT to something closer to the physical units. From this definition of the FFT, the inverse FFT is given by

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kn}{N}} = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-kn}. \tag{16}$$

Note that the placement of the $\frac{1}{N}$ is arbitrary. However, it is significant in trying to return the FFT calculation to physical units. Alternate FFT definitions are available as:

$$\tilde{X}_k = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} x_n W_N^{kn} \Longleftrightarrow x_n = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-kn} \tag{17}$$

or

$$\hat{X}_k = \sum_{n=0}^{N-1} x_n W_N^{kn} \Longleftrightarrow x_n = \frac{1}{N}\sum_{k=0}^{N-1} \hat{X}(k) W_N^{-kn}. \tag{18}$$

This is generally a pain because we want physical units when measuring signals in the lab and the physical units do not have arbitrary scaling.

In order to make any of this analysis self-consistent, we need to have all the FFTs computed with the same scaling. This means that we need to know what any spectrum analyzer or digital oscilloscope is using as its primary equation, and this often involves digging through the middle pages of the manuals. Beyond that, we need to consider the sample period ($\Delta T = T_S = 1/f_S$) used in each measurement, and the number of points in each measurement as these two factors define the frequency bins available in an FFT calculation.

So, whether by spectrum analyzer, by digital oscilloscope, or by using our digital control system to capture a block of data, we understand that we have a linear spectrum of the data. Considering the spectrum analyzer as more of a corner case moving forward, we might use a digital scope or a time capture feature of our digital control system to capture a long stretch of sampled data and use a Digital Fourier Transform (DFT) or Fast Fourier Transform (FFT) to compute the linear spectrum of that signal. No matter how they are computed linear noise spectra do not pass through filters in an analytical way, so we need to generate the complex conjugate of this spectra, multiply it times the original at each frequency point, and then normalize it to get a power spectral density. This gives us a Power Spectral Density (PSD). Section III-B will discuss some details about this calculation.

B. Power Spectral and Cross Spectral Densities

This section deals with the practical generation of auto and cross spectra from spectra generated by Fourier Transforms (or FFTs) or Fourier Series calculations. When the cross or auto spectrum is normalized by the bandwidth of the measurement, we get a spectral density. Because we are making discrete time measurements with a sample period, $\Delta t = T_S = 1/f_S$, our measurement bandwidth is from $-f_S/2$ to $f_S/2$. Thus, we end up normalizing by the sample frequency, f_S. A common term for the auto-spectral density is power spectral density (PSD). Cross spectral densities may be referred to as CSDs. Note that if we assume the same sample frequency for all of our FRF measurements, then we can use the FRFs to filter the spectras in each frequency bin.

For PES Pareto, where we are backing noises out to the point where they are independent inputs, we can consider the CSDs of these noises to be 0. Thus, we really care about the Power Spectral Density (PSD).

Let's consider that we have made a measurement sampled in time at intervals of $\Delta t = T_S$, where there are at least N points in the measurement and N is a power of 2. From this time measurement, we could compute the FFT on the range from $-\frac{f_S}{2}$ to $\frac{f_S}{2}$ using Equation 13. We scale this to physical units via Equation 15. To produce the PSD from $X(f)$, we compute the complex conjugate $X^*(f)$. This is a fairly straightforward computation. At that point we compute the element by element product of the two complex vectors:

$$PSD(x) = \frac{X^*(f)X(f)}{B_e} \tag{19}$$

where B_e is the Resolution Bandwidth of the filter used to compute the spectrum (or the Noise Equivalent Bandwidth which is technically not the same but very close to the Resolution Bandwidth) and where $X(f)$ is the Fourier Transform from Equation 6. This is the smallest change in frequencies that a given measurement can resolve.

In general B_e is inversely proportional to the length of the time window over which a measurement is made , *i.e.,*

$$B_e = \frac{1}{T} \tag{20}$$

where T is the length of the time record. For an FFT,

$$B_e = \frac{1}{T} = \frac{1}{N\Delta t} \tag{21}$$

where N is the number of points in the FFT and Δt is the sample period between points.

Note that for an FFT, the resolution bandwidth is fixed as all the integrations are done over a single period of time ($N\Delta t$, as in Equation 21. Band Selectable Fourier Analysis or Zoom-FFT [31], [32] can be used to maximize the resolution, but this is usually only known to experts. In a brute force spectrum calculation, we could computes a separate integral for each frequency. To eliminate errors due to a partial period integral, the integration should be done over an integer number of periods of the frequency in question. This would that except in special cases, the actual resolution bandwidth of the calculations at different frequencies will differ slightly, but the spectral leakage would be largely eliminated.

MATLAB computes the FFT in Equation 14 and (with some details to be filled in later) then produces

$$P_{xx} = PSD(x) = \frac{X^* .* X}{N} \tag{22}$$

where X^* is the complex conjugate of X and N is the number of points in the FFT and the $.*$ operation is the element by element multiply of two same-sized vectors in MATLAB . (Windowing and scaling are standard methods of improving the performance of FFTs by driving the time signal to 0 at the beginning and end of the data run, but we will not discuss those here.)

Note that these units are not physical. From Bendat & Piersol[23], page 407 there is a procedure for computing a PSD from FFT based measurements. At any frequency, f_k, the PSD of a signal x is given by:

$$\tilde{P}_{xx}(f_k) = \frac{X^*(f_k)X(f_k)}{N\Delta t} \tag{23}$$

where $X(f_k) = \Delta t X_k$. This means that

$$\tilde{P}_{xx}(f_k) = \frac{(\Delta t)^2 X_k^* X_k}{N\Delta t} \tag{24}$$

or

$$\tilde{P}_{xx}(f_k) = \frac{\Delta t X_k^* X_k}{N} \tag{25}$$

so

$$\tilde{P}_{xx} = \frac{\Delta t X^* .* X}{N} \tag{26}$$

and thus

$$\tilde{P}_{xx} = \Delta t P_{xx} \tag{27}$$

i.e. to go from MATLAB units to physical units, multiply the MATLAB PSD by Δt. Now, the MATLAB PSD function has been deprecated, and now they favor a Welch algorithm that does the physical scaling found in Equation 27. Alternately, one can use MATLAB 's periodogram function. So, for a time domain measurement in the vector, d, of length $N_{meas} \leq N = NFFT = 2^M$, we can:

- Compute from MATLAB 's FFT routine:

$$d.FFT = fft(d, NFFT); \tag{28}$$
$$d.FFT_{phys} = T_S * d.FFT; \tag{29}$$
$$d.PSD_{FFT} = \frac{(d.FFT)^* .* d.FFT}{NFFT}; \tag{30}$$
$$d.PSD_{FFT,T_S} = T_S * d.PSD_{FFT}; \tag{31}$$

- Compute using MATLAB 's Welch routine:

$$[P_{xx,w}, f_w] = pwelch(d, NFFT, [], NFFT, fs); \tag{32}$$

- Compute using MATLAB 's periodogram routine:

$$d.PSD_{Per} = periodogram(d, [], NFFT); \tag{33}$$

These three methods give the same PSD and give a PSD that is scaled to be equivalent to the Fourier transform of an analog signal. We believe that for most purposes, the first method has the advantages that it is relatively straightforward and completely transparent. Furthermore, the routines are easily mimicked in computer languages besides MATLAB or Python, making the math far more portable.

The FFT calculations above all compute an FFT from $-\frac{f_S}{2}$ to $\frac{f_S}{2}$ but since we really can only make use of the positive frequency axis, we want to use the FFT frequency bins from 0 to $\frac{f_S}{2}$. The Fourier Transform of real function is Hermitian [8], which means that the real part is even and the imaginary part is odd, but when we multiply the FFT by its complex conjugate, we get a real and even function on the frequency axis. We can then take double the values of the PSD for the positive frequency bins, leave the bin at DC the same, and discard the portion with the negative frequency bins.

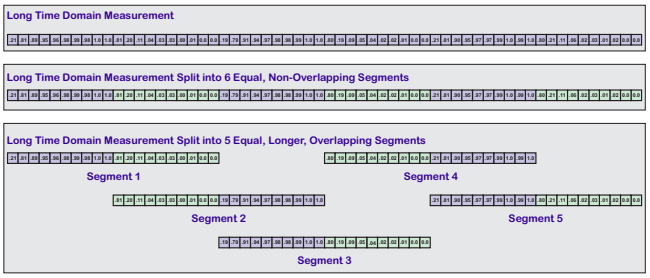

Fig. 9. Diagram of overlap processing.

Note that it is typical to pick the next power of 2 above N_{meas} to compute the FFT. However, we may have a situation where we have a lot of data i.e. N_{meas} is far larger than any reasonable FFT we might wish to make. This means that we have the option to do some averaging. Averaging is a good thing in noise analysis because – to put it simply – one cannot get an expected value from a single vector of data, and one cannot approximate an expected value without a bit of averaging. If one makes a long data measurement, and one assumes that the noise process is ergodic, that is that the time averages are the same as the ensemble averages [23], then we can break a long measurement into sections to be FFTed and those FFTs can be averaged. One of the tricks one could do with the old MATLAB PSD algorithm was perform overlap processing: a long run of data could be broken in to multiple segments for FFTs, but those segments could overlap giving the illusion of more points while still averaging, as diagrammed in Figure 9.

C. Quantization Noise: The Widrow Model and Others

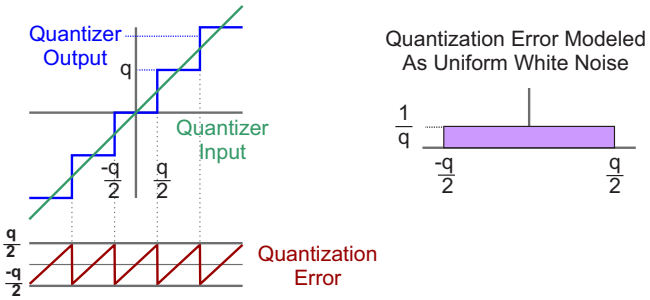

Fig. 10. Diagram of quantization and Widrow model.

There are other time domain measurements that yield only a single number, such as the variance due to quantization in the Widrow model [33]. This variance must be spread across the frequency band in some logical way, so the authors chose to normalize it by the frequency bandwidth. This is consistent

with the texts [34], [35] on quantization devices and with will be described briefly here. Note that [35] is erroneous in that the normalization by $T_S = 1/f_S$ has been omitted.

The Widrow model [33] of quantization is based on an analog of the Nyquist sampling theorem [36]. A conceptual quantizer is shown on the left side of Figure 10. Quantization is not a random process, but a deterministic, nonlinear operation. This makes it hard to do anything with passing the math through a filter. Widrow's insight was that while quantization was deterministic and nonlinear, if the signal excited enough of the scale of the quantizer, and the quantization bins were fine enough, that the probability of the quantized signal falling anywhere in the quantization bin could be modeled as a uniform density white noise on the interval $[-\frac{q}{2}, \frac{q}{2}]$, where q is the size of a minimum quantization interval. This is displayed on the right side of Figure 10. With this model, computing the mean and the variance of the quantization "noise" reveals that the mean and variance are:

$$\mu_q = 0 \text{ and } \sigma_q^2 = \frac{q^2}{12}. \tag{34}$$

This number for variance is used in texts all over the world [37]. That is all well and good, but how do we translate this variance into a PSD that we can pass through filters?

The leap of faith here is to treat that Additive Uniform White Noise as an Additive Gaussian White Noise, with zero mean and variance of $\frac{q^2}{12}$. As both are white, they have an autocorrelation that is only nonzero at an offset of 0. At an offset of 0, the autocorrelations are a Delta functions with a hight equal to the variance. We know from Parseval's Theorem [8], [38] that the variance in time is equal to the integral of the PSD over frequency, in this case, from $-\frac{f_S}{2}$ to $\frac{f_S}{2}$, so we can as a first approximation, give the quantization noise a uniform PSD on the interval $[-\frac{f_S}{2}, \frac{f_S}{2}]$ with magnitude $\frac{q^2}{12 f_S}$. In summary, we set the noise PSD due to quantization as:

$$\Phi_{QQ}(f) = \frac{q^2}{12 f_S} = \frac{q^2 T_S}{12}, \text{ using } f \in [-\frac{f_S}{2}, \frac{f_S}{2}] \tag{35}$$

If we are only using the interval from $[0, \frac{f_S}{2}]$, then the PSD magnitude is $\frac{q^2}{6 f_S}$.

$$\Phi_{QQ}(f) = \frac{q^2}{6 f_S} = \frac{q^2 T_S}{6}, \text{ using } f \in [0, \frac{f_S}{2}] \tag{36}$$

This is a correction to the classic circuit text [35] where the noise PSD is set to $\frac{q^2}{12}$ from $f \in [0, \frac{f_S}{2}]$. The key to understand is that the integral of the PSD over the frequency range, must equal the total variance of $\frac{q^2}{12}$.

The swap between a uniform and Gaussian additive white noise is not perfect, but it does allow us to deal with quantization noise in the PES Pareto method, i.e. to pass it through digital filters. We will discuss how to apply these ideas to measurements of ADC and DAC "noise" in Section V-D. This model serves us as a theoretical minimum for the quantization noise, if all of the noise in the ADC or DAC noise and can be used to generate a PSD as described above.

Other measures of ADC and DAC noise depend upon the access to the system. If one can open up the system and drive the ADC with either a single tone, or with an open circuit, one can establish (for the ADC) an input noise level. Applying an open circuit to the ADC means that all that is coming in is noise. However, this doesn't exercise the full range of the ADC. An alternate input is to drive the ADC with a single sinusoid that excites the full range of the ADC. The issue then is that the tone dominates the response, so it needs to be removed digitally. This can be done in the time domain with an adaptive noise canceler [39], which adapts coefficients of a generated sine and cosine to cancel the magnitude and phase of the input signal. If done after an FFT, we need to look at the frequency bins that contain the tone and set it equal to some average of side bins outside the skirt of the tone. If the tone is removed in the time domain, we can also remove the DC level by averaging the entire data run to establish the sample mean. For an N-sample measurement,

$$\bar{x}_{k,N} = \frac{1}{N} \sum_{i=0}^{N-1} x_{k-i}, \tag{37}$$

and we proceed as above.

We can also use this to back out a measure of how many bits we get out of the ADC. Consider if we have an N-bit ADC that spans a particular full voltage scale, FS. Then

$$\frac{FS}{q} = 2^N \text{ or } q = \frac{FS}{2^N} \text{ or } FS = q 2^N. \tag{38}$$

Now using (34) we can substitute:

$$\sigma_{meas}^2 = \frac{q_{eff}^2}{12}, \text{ so} \tag{39}$$

$$q_{eff} = \sigma_{meas}\sqrt{12} = \frac{FS}{2^{N_{eff}}}. \tag{40}$$

This means that

$$2^{N_{eff}} = \frac{FS}{\sigma_{meas}\sqrt{12}}, \text{ so} \tag{41}$$

$$N_{eff} = \log_2\left(\frac{FS}{\sigma_{meas}\sqrt{12}}\right) \text{ bits.} \tag{42}$$

D. Using PSDs in PES Pareto

With all these tricks, one ends up being opportunistic, generating as many different spectra from as many different locations as possible, with the loop open whenever possible, so as to isolate different noise sources. Because PSDs can be added, they can be subtracted from each other and so by a process of elimination, we arrive at isolated noise sources. However, there are measurements that can only be made when the system is in closed-loop. The measurement of PSN is one such case. In this case, we must use subtraction of the PSD from the sources that we could isolate to leave us with the PSD of the remaining source.

As we try to convert any measurement into a PSD, the following is useful to keep in mind:

- We will be working on the waveform. That is, we will have a frequency axis from some lower frequency to

some higher frequency. That higher frequency will be no more than the Nyquist frequency.

- In order to do waveform math on multiple waveforms (scalar math one frequency bin at a time), all waveforms must share the same frequency axis.
- If we have an instrument, or instrumentation functionality that can measure the PSD directly, we can use that. If the instrument can measure the FFT of the signal, then we must multiply it, frequency bin by frequency bin, times its complex conjugate. If we simply have a single time domain variance number, then this must be distributed across the frequency spectrum as described in discussion of quantization in Section III-C.
- PSDs generated by independent processes can be added. Thus, if we have different, independent noise sources generating the PSDs at a particular measurement point, then the results of those sources can be considered added at the measurement point. We have a strata of noise contributors.
- By model or measurement, generate frequency response functions (FRFs) for the loop components on the same frequency axis as our PSD measurements. Generically call these H for this discussion.
- Multiply H times its complex conjugate to get $\|H\|^2$.
- Multiply $\|H\|^2$ times the power spectrum or PSD to get effect of the loop on noise. Note that $\|H\|^2$ must be the appropriate units for filtering the PSDs.
- The resulting output is another power spectrum or PSD.
- We can use superposition to built up contributions from many sources.
- We need to do some "loop unwrapping" to extract proper *input* noise levels for model.

The chief restriction of manipulating PSDs is that we will have to limit ourselves to linear models of the system. However, by doing so we are able to actually add and subtract PSDs. In order to do so, we formally should require some knowledge that the noise sources are independent. It turns out that there is no way to verify this for all sources, but it is very likely true. While any measured signal in the loop is correlated to several noise sources, each source arises from an independent physical phenomenon. Furthermore, without allowing for superposition of noise measurements, it would be next to impossible to analyze the noise of a *measured* system. Thus, it is a starting point we must choose.

IV. Using the HDD Example Guide Us

When we introduced the PES Pareto methodology in the 1990s, we were working in the context of the disk drive industry and had plenty of measurements from actual disk drives. In the years since then, there have been few chances to measure more disk drives. That being said, the measurements from that original work are still quite instructive. We will reference the block diagram in Figure 11 to help us "walk around the loop." We will have an augmented discussion of the measurements made on that system, showing how different PSD measurements were extracted for different blocks in Section V. We and how we put it all together to get noise strata in Section VI and then Section VII will show how to use our extracted sources and models to extrapolate how changes in a noise source would affect a particular measurement point.

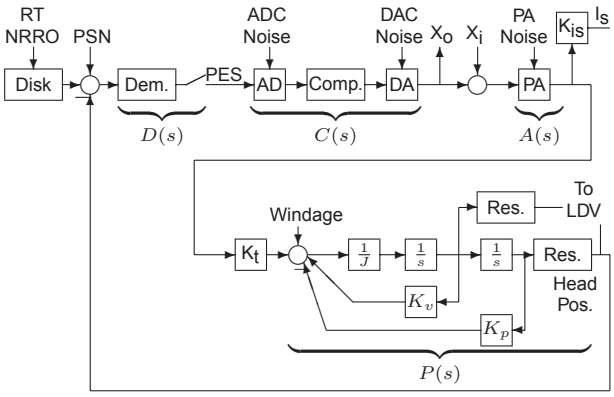

Fig. 11. Generalized view of track following model in an HDD. Each block can be considered both a source of noise and affected by noise. PSN refers to Position Sensing Noise, that is the inaccuracies caused in trying to extract position information from alternating polarities of magnetic domains.

V. Measurements for PES Pareto

The ideas of Section II fundamentally depend upon having measurements of noises at different parts of the loop to manipulate back and forth. Some noises can be measured simply by breaking the loop at some point and making a measurement downstream of the block in question. In other cases, the measurements are only available when the loop is closed in feedback. For the most part, the method is assuming that we can apply linear analysis through our feedback loop, especially when the signals are small (as we expect them to be for noise analysis). This breaks down when we want to relate the effects of quantization from analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) to these linear methods.

It should be noted that this section contains the most ad-hoc techniques in the tutorial. The measurement test points we want are not always available. Some measurements can only be made when the loop is broken, while others are only feasible in closed-loop. A certain amount of faith is needed that the noises we measure when we isolate a component are representative of those noises when the loop is operational.

A. Measurements in Open and Closed Loop

In order to do a practical analysis of the contributors to error, the fundamental question that must be answered is: *What can be measured?* In any real system, we will not have access to all the measurement points that we desire. Furthermore, although many different analysis tools might theoretically be available, they are useless to us if they cannot make use of the actual laboratory measurements available to us.

In order to guide our measurements and our modeling, it is useful to have a map of the system. In the original hard disk drive example, the Figure 11 served as the map for our tour

of noises in the system. Starting at the left of this diagram, the reference position that the actuator arm must follow is the position of the magnetic track written on a disk, turning on a spindle. Only the position error – the difference between the reference track position and the readback head position – is sensed by the readback head, and this error signal is sent to the demodulator. The demodulator outputs a set of numbers at the system sample rate, and these are combined electronically to form PES. This PES signal is then converted to a digital format via an analog to digital converter (ADC), filtered by the compensator and then sent back out to the power amplifier via a digital to analog converter (DAC). The power amp converts the desired voltage into a current to drive the voice coil actuator (with torque constant K_t). The actuator itself has rigid body behavior as well as resonances. Through this, the head position is set. The position error is then sensed by the head. Absolute head position is not generally known from what is read off of the disk surface, but can be obtained in the laboratory by shining a laser spot from a Laser Doppler Vibrometer (LDV) off of the side of the head. While this nominally measures velocities, the result can be accurately integrated in time (for the frequencies we are concerned with) to obtain position.

We had the following useful models that we could measure from our physical system. Our block groupings in Figure 11, are the plant, $P(s)$:

$$P(s) = \frac{1}{K_t}\left(\frac{LDV}{I_{sense}}\right)_{\text{3WM}} \approx \frac{1}{K_t A(s)}\left(\frac{LDV}{X_{in}}\right)_{\text{OfCLM}}, \tag{43}$$

the power amplifier, $A(s)$:

$$A(s) : \text{from block model}, \tag{44}$$

the compensator, $C(s)$:

$$C(s) = \left(\frac{X_{out}}{PES}\right)_{\text{meas}} \approx \text{Analytic Model}, \tag{45}$$

and the servo position generator or demodulator, $D(s)$:

$$D(s) = \frac{\left(\frac{NPES}{I_{sense}}\right)_{\text{meas}}}{\left(\frac{LDV}{I_{sense}}\right)_{\text{meas}}}. \tag{46}$$

In traditional measurement systems using external instruments, such as this disk drive example, there were a limited number of measurement points that could be accessed around the loop: X_{out}, I_{sense}, PES, and head velocity (and position) via the LDV. In general test signals could only be injected into the loop at X_{in}. However, in the two decades since the original PES Pareto work, the nature of instrumentation and measurements has changed quite a bit. As digital controllers have gotten more ubiquitous, it has made less and less sense to connect analog instruments to the loop, that is, instruments that required their own ADCs and DACs. Many of the classic instruments from the original PES Pareto work [1], [2], [3], [4] are no longer manufactured. Even for these instruments,

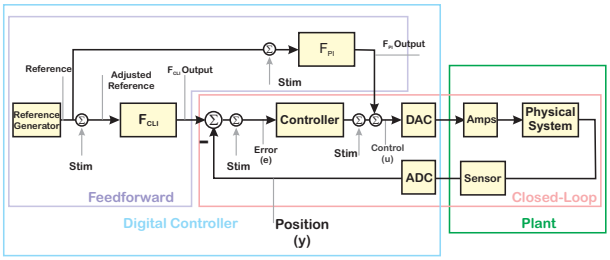

Fig. 12. Generic feedforward-feedback loop with measurement points. Note that if measurement IP is built into the digital controller, then the number of test injection and measurement points gets much larger. Furthermore, the measurements are made at frequencies compatible with the digital control system.

many of the digital control loops run at sample rates an order of magnitude higher than these instruments could allow.

Instead, it has become more logical to build in measurement tools right into the digital controller [27]. Diagrams such as the one in Figure 12 should become the norm rather than the exception. In particular, such built-in measurements allow the measurement algorithms to use the same data conversion path as the control loop. Furthermore, the number of digital measurement points (both for signal injection and measurement) are dramatically increased and the data for measurement does not need a format conversion from the data for running the loop.

The classic laboratory instruments for these kinds of measurements include spectrum analyzers (discussed earlier), network analyzers (which calculate the ratio of input and output spectra), and digital storage oscilloscopes (DSO). Any of these can be recreated within a modern digital controller and should be for the reasons mentioned above. What are known as dynamic signal analyzers (DSA) used for measuring frequency response functions (FRFs) of control systems, are generally low frequency versions of network analyzers. Rather that focusing on the specific instruments of the original work, it seems more useful to discuss the types of measurements that we want for the methodology.

Returning to our disk drive example, there are several likely noise sources in the loop. First of all, there are the noises associated with the moving disk and the readback process. These all enter the loop at the same point, but have different root causes. The noise due to the motion of the disk attached to a ball bearing spindle creates both Repeatable Run Out (RRO) (typically at orders of the spindle rotational frequency) and Non-Repeatable Run Out (NRRO). One of the interesting properties of servowritten disks is that one pass of the NRRO is usually locked into the servo position information when it is written. Thus, this written in NRRO is repeated at every revolution of the disk. The other noise source that enters at this point is the noise from the readback process of position information, called Position Sensing Noise (PSN). This noise can be due to the magnetic

domains on the disk, the behavior of the magnetic readback head, the interaction of these two, or the action of the demodulator. (We lump demodulator noise into PSN for our current analysis.) Downstream in the loop, there are potential noise sources at the ADC and DAC (due to quantization), noise at the power amp, and finally windage. Windage is caused by the air flow generated as the disk spins. This air flows over, under, around, and into the actuator arms and the readback head, disturbing the head position. Given all these potential noise sources, there is a fundamental need to identify which of these – if any – are the most significant contributors to PES. With this information, the effort to reduce the noise in PES can be concentrated on the critical few.

It is worth noting that we purposely ignore external shock and vibration in this analysis for two reasons. First of all, external shock and vibration is heavily influenced by the drive's operating environment while the above noises are a function primarily of the drive. The second is that prior work in this area [11], [12] gives us some confidence that we already have a reasonable engineering solution to many types of external shock and vibration. Thus, PES Pareto focuses on internal or self-generated noises.

B. Measurements/Modeling of Power Amplifier Noise

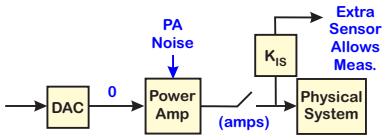

Fig. 13. Generating different conditions to measure power amplifier noise.

Referring back to Figure 6, power amplifiers are usually needed to convert the low voltage signals out of DACs into currents large enough to drive an actuator. This is pretty universal, unless that amplification is done in the actuator itself. They tend to have a low-pass nature, but their bandwidth is usually considered beyond that of the actuator, plant, or closed-loop system. The nature of their amplification allows for the generation of noise, and so we can often break out these components as individual blocks to be analyzed for their noise contribution. When a model is needed for filtering noise, it can often be generated in MATLAB to mimic the data sheet frequency response function (FRF), but at the frequencies that we want for our PSD measurements. Alternately, with enough control over the digital controller (as diagrammed in Figure 12), one can stimulate the DAC to generate an FRF measurement, provided we have a way to measure the current being produced by the power amplifier. In Figure 13 there is a sense resister often used for monitoring the current being produced by the power amp to drive the actuator. In the disk drive, such a signal has utility as a secondary measurement of what is going into the actuator. Note that we still need to convert the voltage (converted from the sensed current using a reference resistor).

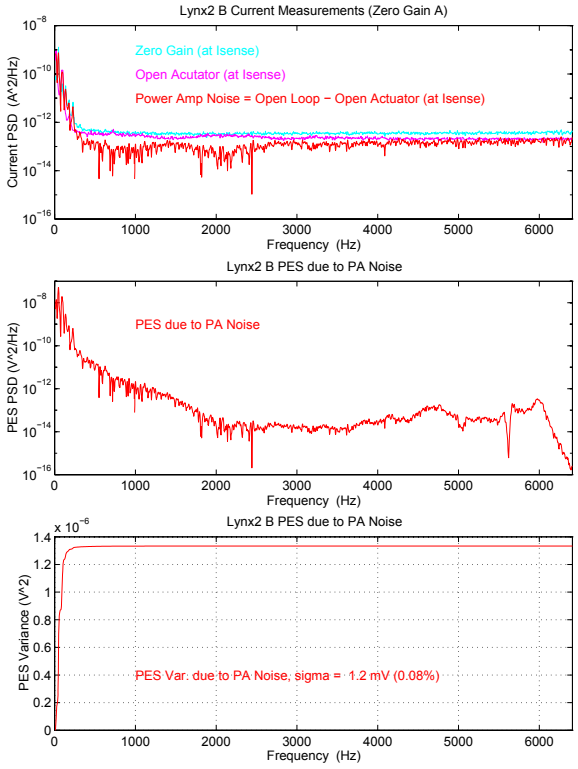

Fig. 14. Measurement of Power Amplifier Noise on Lynx 2 Disk Drive. PSDs are generated from open loop and open actuator configurations, allowing the difference to be considered the power amplifier noise.

To isolate a PSD of the noise due to the power amplifier we need some sort of differential measurement. Since the amplifier can be run typically in open loop, we can set the DAC output to 0 to generate a PSD of the PA and ambient actuator input noise (Figure 13). In this case, we can drive the DAC with an all zeros code. We would normally be concerned with DAC quantization, but an all zeros code should produce a calibrated 0 output at the DAC voltage. (The signal is not moving, so there is no quantization to speak of.) In the second measurement, we physically open the circuit from the power amp to the rest of the actuator. The current sense measurement now only has the ambient actuator input noise. If we take the expected values of the PSDs from both measurements (and this means averaging multiple measurements) we can assume independence and subtract the two PSDs. This means taking more than one measurement of these signals to get the statistical independence to kick in.

We then measure directly at I_{sense} and since that is the source point as well, the back filter to the source is simply 1 for all frequencies of interest. We difference the PSDs in the top plot of Figure 14. In the middle plot of Figure 14, we forward filter to PES through the magnitude squared filter:

$$\left\| \frac{K_t P(s) D(s)}{1 + K_t P(s) D(s) C(s) A(s)} \right\|^2 \tag{47}$$

to get the contribution to PES, where $P(s)$, $A(s)$, $C(s)$, and

$D(s)$ are defined in Equations 43 – 46. In the lower plot of Figure 14, this PSD is integrated to get the variance (in the frequency domain) due to the power amplifier.

C. Measurements/Modeling of Plant Disturbance

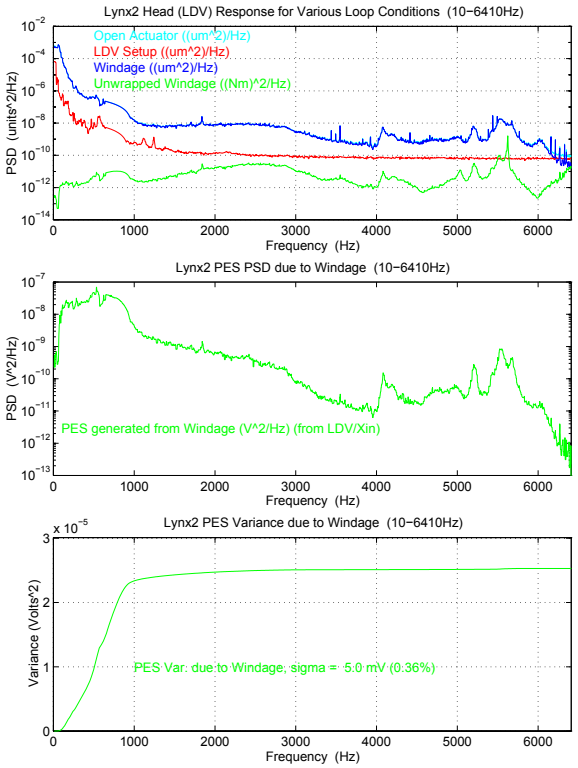

Fig. 15. Measurement of the disturbance due to air flow on the drive head (windage) on the drive example via an LDV. Backfiltering the LDV measurement through the inverse plant model gave the windage as an open-loop source, which could then be filtered forward through the model to see its effect on PES.

Separating out plant disturbances from noise in the position sensing has a particular challenge when the same sensor is used in the measurement for the error signal as is used to try to quantify plant disturbances. This almost always calls for an external sensor, one that is of high enough quality to give a signal that can be trusted on its own. In the case of the HDD, the issues with the loop sensor were that it had a limited bandwidth (due to the sectored servo patterns used in HDDs) and the generation of the error signal itself was part of the experiment. There was no option to up the sample rate and filter, and the error signal (PES) was only really available when the drive was in closed loop. Furthermore, it was an error only sensor, not giving us the actual position of the drive head. This is a major limitation, since two significant noise suspects were the buffeting of the drive head by the air flow generated by the spinning disks and the actual generation of the error signal from the servo position dibits. An instrument such as an laser interferometer [40] or a laser Doppler vibrometer (LDV) could provide an actual position measurement (against a calibrated distance set to 0) if an appropriate spot on the drive head could be found for reflecting the laser spot.

In the disk drive example, a Polytec LDV provided highly accurate position measurements for any frequency above 10 Hz. We could minimize the feedback loop gain in the controller and measure the head position (not the error) by shining the laser spot on the side of the drive head. We could then back filter to the noise source (modeled at the plant input) by back filtering through:

$$\left\| \frac{1}{P(s)} \right\|^2 . \tag{48}$$

With this windage "input noise", we now filtered forward through

$$\left\| \frac{P(s)D(s)}{1 + K_t P(s)D(s)C(s)A(s)} \right\|^2 \tag{49}$$

to get the effect of windage on PES.

This is an example of a more general problem: there will always need to be a way to separate the disturbance generated movement from the error generated in measuring position. After all, we cannot perform any reasonable tradeoffs in a Kalman filter design [41] if we have no idea what the noise powers of w and v are. There are other noises in the loop, but at some point, we need to be able to isolate these as well.

D. Measurements/Modeling of ADC and DAC Noise

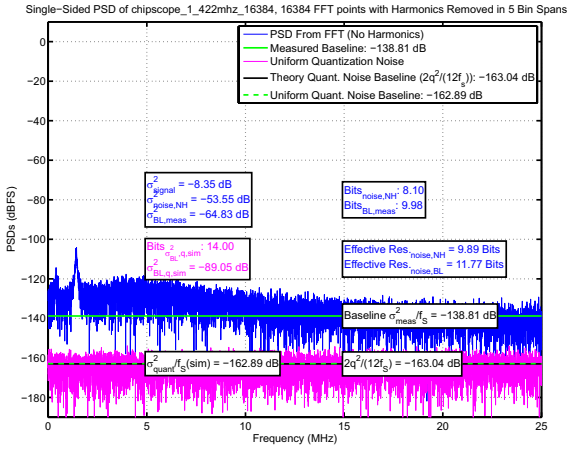

Fig. 16. Decomposition ADC noise with a removed tone.

In Section III-C we discussed some basic modeling for quantization noise. In this section, we will apply those ideas to generating noise as a source for our Pareto measurements. Recalling Section III-C, we had arrived at a quantization PSD (35) of $\Phi_{QQ}(f) = \frac{q^2}{12 f_S} = \frac{q^2 T_S}{12}$. At a first cut, we can use this to define the quantization noise being added in at the output of our ADC or DAC.

However, there are methods of making measurements on ADCs. Usually, the engineer injects either a simple tone or 0 at the analog input to the ADC. The signal is sampled and quantized and the spectrum of the measurement is FFTed. In the case of the tone, this signal will dominate any integral that is done, so it is removed, either by excising the frequency bin and replacing an average of neighboring bins, or by applying an adaptive noise canceling approach [39] to match

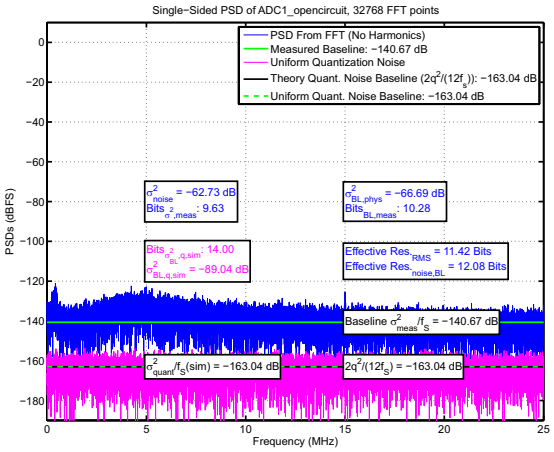

Fig. 17. Decomposition ADC noise with an open circuit.

and remove the tone before the FFT is computed. In either case, the sample mean of the data is removed.

Now, why inject a tone at all if we are going to remove it? The tone exercises the full range of the ADC, which was one of the assumptions of the Widrow model discussed in Section III-C. As such, the noise properties of the ADC with and without a tone might be different, so it is good to have both measurements. Secondly, there are several possible noise measures used with respect to signals, and one of them is dBc or Decibels with respect to a carrier. We need a tone, a carrier, to compute this. Removing the tone itself has its own issues. The best method is probably the adaptive noise canceling approach, since that removes the matched signal only and not other signals at that frequency. Removing a tone once an FFT has been done has issues of spectral leakage, that is that the FFT bin itself bleeds into others. To combat that long data runs being FFTed are often windowed – in this context meaning that they are multiplied by a window that drives the end points of the data record to 0. (This is another reason why overlap processing is a useful trick, allowing us to lengthen the record to a point that the window doesn't affect the data points we want.)

Assuming we have made a measurement (with a tone that was removed or without a tone) and computed the FFT, we then can compute the PSD and make it one-sided using the methods described in Section III-B. How do we related this back to our flat quantization PSD model? We fit that data to a flat PSD across the frequency band (usually looking at the high frequency tail to establish the level), and then set that level to $\frac{q_{eff}^2}{12f_S}$. We can now back out our effective quantization level, q_{eff}.

An example is shown in Figures 16 and and 17. Figure 16 shows a measurement of a 14-bit ADC sampled at 50 MHz, into which a 1.422 MHz tone was injected. A 16,384 point FFT was performed and the frequency bins containing the tone and its harmonics were flattened by replacing their signals with the averages of nearby bins. To keep the spectral leakage low, the data was windowed with a 16,384 point Hanning window to drive the end points to 0. If we start with by normalizing the full scale range, FS to ± 1, we can then calculate the effective value of q and using the math in Section III-C, compute the PSD for the theoretical quantization noise. This is the magenta curve, and the computed PSD is plotted along with the pure theoretical curve. Another curve, taken from averaging the high frequency end of the computed PSD is also drawn, as a way of verifying that the curve represented the original theoretical value. We then could take the measured signal (with the tone removed) and calculate the average of the high end of the PSD, allowing us to draw the green curve back, to reveal an effective level of quantization PSD. Again, using the calculations from Equation 42 in Section III-C, we back out an N_{eff}. Figure 17 repeats the same calculation, but with an open circuit input to the ADC. There is no tone to remove, but it is helpful to see that the resulting calculations give us similar numbers for the effective bits.

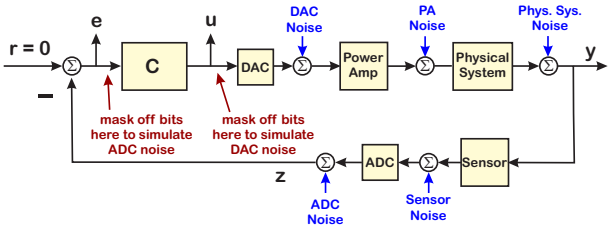

Fig. 18. Simulating ADC and DAC quantization by masking off bits at different signal points. On their own, the masked off bits are still one of many noise sources, but when they are differenced from each other, the unchanging noise PSDs drop out, revealing the effects due to the quantization.

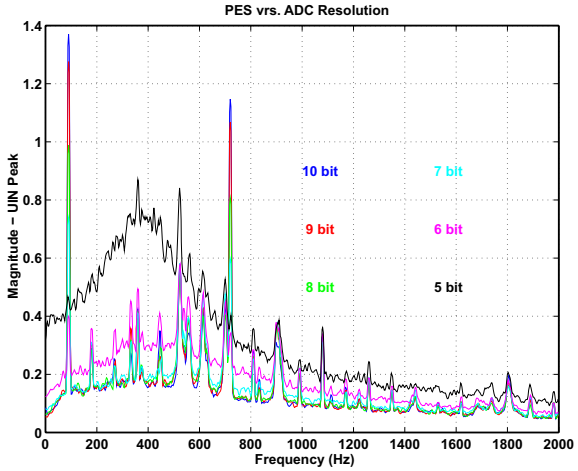

Fig. 19. Measurement of ADC quantization by masking off bits at the error signal.

This can be done with an ADC if we have a reliable analog signal source of higher resolution than the ADC and if we have the ADC connected to a test system. The first characteristic is to make sure we are not including the noise in the signal source, and the second characteristic is needed just to get the discretized signal into a place where it can be processed. It becomes more difficult for a DAC, because while we may be able to generate a high fidelity signal to send out to the DAC, we now need to digitize the analog voltage produced and get that back into some analysis

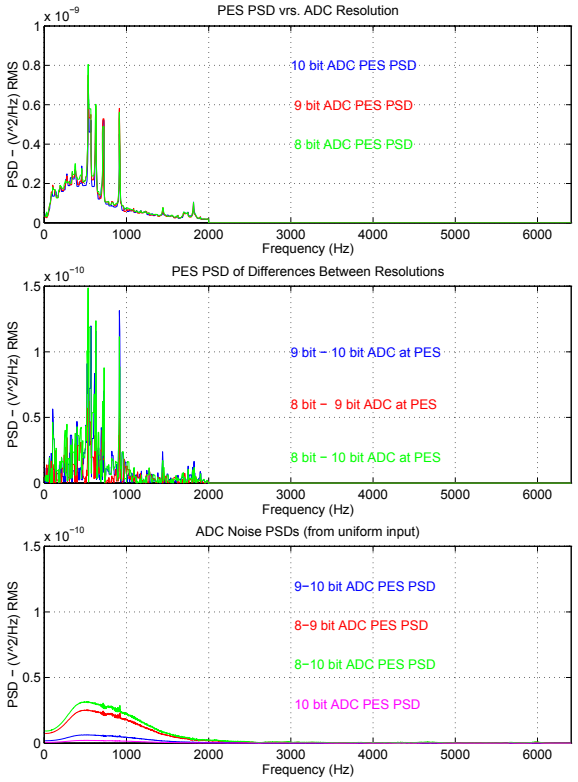

Fig. 20. Measurement of ADC noise in closed-loop by differencing measurements of the type in Figure 19. Rather than try a numeric fit of the differences in PES, a scaled uniform noise is filtered through to the PES to match the difference signals.

engine, without that discretization becoming the dominant or even a significant part of that measurement. For either device, we often do not have the ability to evaluate the components separately. In this case we can employ a scheme diagrammed in Figure 18.

When we cannot separate out the ADC from the loop, we use this scheme of masking off bits in the error measurement (PES) (assuming the reference signal, $r = 0$). By artificially masking off these bits, we can generate a PSD of the different levels of PES. For the hard disk example, the resulting plots are shown in Figure 19. Note that this PSD was made with frequency bins out only to 2 kHz, which did not match the other measurements, so the frequency bins were zero padded with more bins at higher frequency to produce the top plot of Figure 20. The differencing of those measurements is shown in the middle plot of Figure 20. We find that differencing these small PSD levels made it difficult to have any sort of non-visual fit to the data. Instead, the effective level of quantization noise was back calculated from PES measurements by assuming a uniform input noise distribution. This was then forward filtered to PES via:

$$\left\| \frac{K_t P(s) D(s) C(s) A(s)}{1 + K_t P(s) D(s) C(s) A(s)} \right\|^2 = \left\| \frac{1}{T_{cl}} \right\|^2. \quad (50)$$

Eventually, the scaling of quantization that matched the

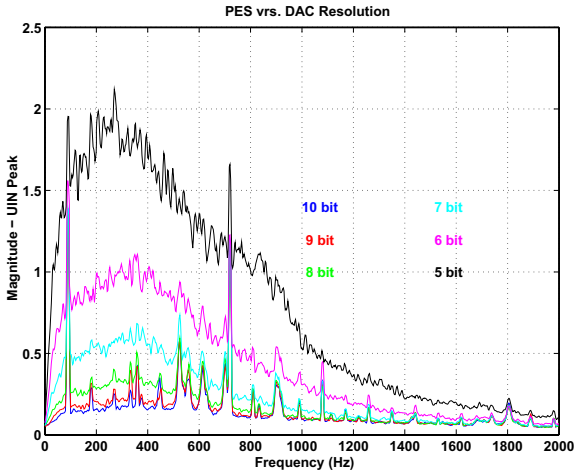

Fig. 21. Measurement of DAC quantization by masking off bits at the controller output signal.

curves in the lower plot of Figure 20 was:

$$q = \frac{1.25\ V}{512\ counts} * \frac{1}{20}. \quad (51)$$

We employ a similar measurement scheme for quantization noise from the DAC. By masking off bits being sent to the DAC, we can generate a PSD of the different levels of PES. For the hard disk example, the resulting plots are shown in Figure 21. Again, this PSD was made with frequency bins out only to 2 kHz, which did not match the other measurements, so the frequency bins were zero padded with more bins at higher frequency to produce the top plot of Figure 22. The differencing of those measurements is shown in the middle plot of Figure 22. Once again, we find that differencing these small PSD levels made it difficult to have any sort of non-visual fit to the data. Instead, the effective level of quantization noise was back calculated from PES measurements by assuming a uniform input noise distribution. This was then forward filtered to PES via:

$$\left\| \frac{K_t P(s) D(s) A(s)}{1 + K_t P(s) D(s) C(s) A(s)} \right\|^2 = \left\| \frac{C(s)}{T_{cl}} \right\|^2. \quad (52)$$

Eventually, the scaling of quantization that matched the curves in the lower plot of Figure 22 was:

$$q = \frac{0.3125\ V}{512\ counts} * \frac{1}{9}. \quad (53)$$

E. Channeling Sherlock Holmes

At some point, we have isolated all the contributors to the error signal (PES in our disk drive example) that we can and we can put them together. We can show them independently to show the most important ones, or stack them cumulatively to show how much of the PES PSD we have accounted for, as shown in Figure 23. There is still a large amount of noise that is unaccounted for. If we plot the PSD of PES - the Cumulative PSD of all the noises we have accounted for, we get the plot of Figure 24. A feel for sensitivity functions, especially those plotted on a linear frequency scale reveals that this looks a lot like:

$$k * \|D(s) S_{cl}(s)\|^2 \quad (54)$$

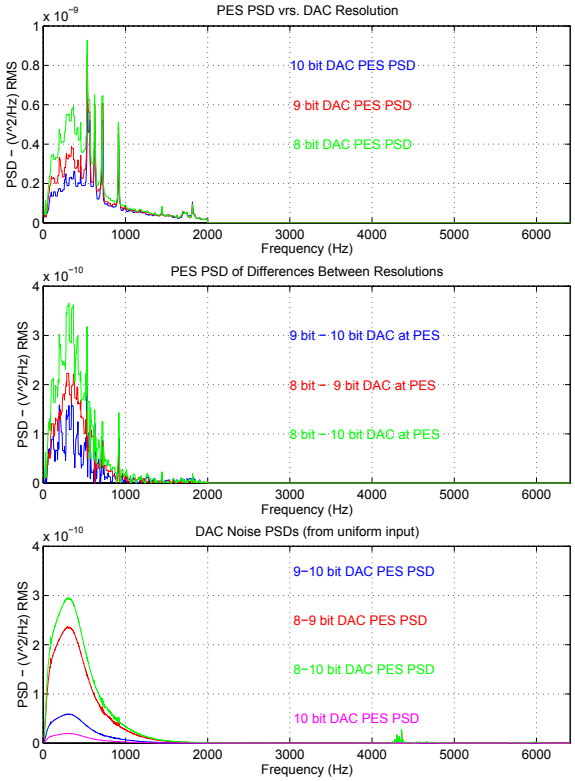

Fig. 22. Measurement of DAC noise in closed-loop by differencing measurements of the type in Figure 21. Rather than try a numeric fit of the differences in PES, a scaled uniform noise is filtered through to the PES to match the difference signals.

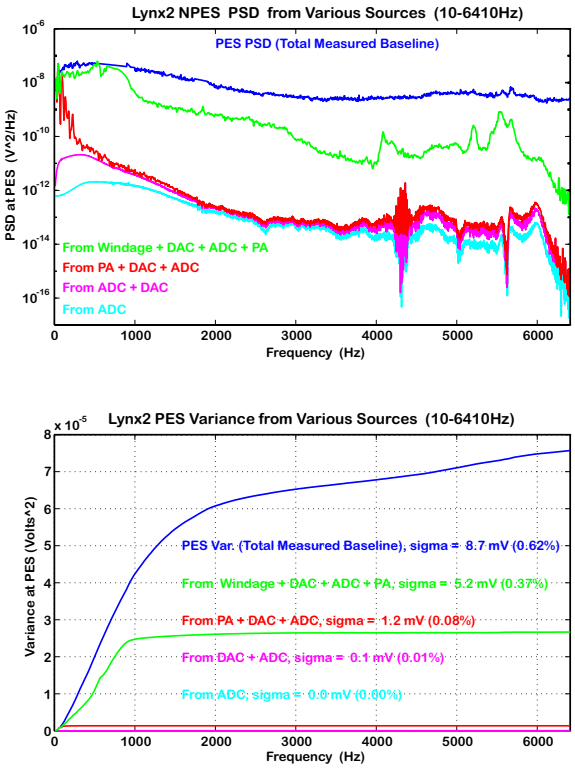

Fig. 23. Decomposition of baseline noise sources in a hard disk. Cumulatively plotted to account for all of PES. The top plot shows the PSDs. The bottom plot has them integrated across frequency.

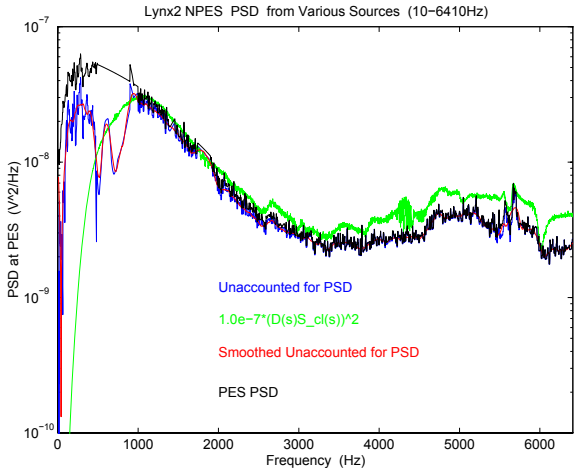

Fig. 24. Unaccounted for PES PSD noise.

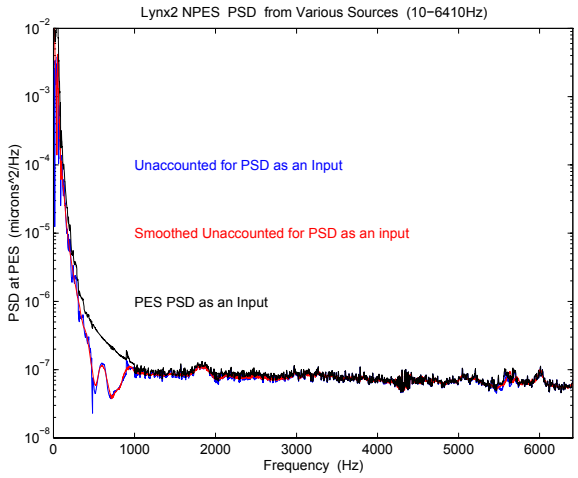

Fig. 25. Unaccounted for PES PSD noise as an input.

At this point in the original effort [1], [2], [3], [4], we turned to the wisdom of Sherlock Holmes, who although fictional [9], was an ancestor of the equally fictional Mr. Spock. "An ancestor of mine maintained that if you eliminate the impossible, whatever remains, however improbable, must be the solution. [42]" To make this unaccounted for noise an input to the error signal (which is where any noise in position sensing would appear in this diagram), we filter the PES PSD and unaccounted for PSD by

$$\left\| \frac{1}{D(s)S_{cl}(s)} \right\|^2 \tag{55}$$

to get to a possible input Position Sensing Noise (PSN) PSD. This is shown for our example in Figure 25.

In our measurements, we have been able to find "simple tricks and nonsense [24]" to isolate noise PSDs from quite a few noise sources. We have brought in extra sensors and measurement points to give representative measures of other noise sources. In the disk drive example, windage (Section V-C) was a large component. Still, we get to the total PSD of the broadband noise at the error signal, remove the PSDs that we have isolated, and we have something left over. This

is the unaccounted for noise, but if we have confidence in our structural model, and if we have found a way to the other noises, then what is left, no matter how improbable, must be the sensing noise PSD.

VI. Noise Sources and Noise Strata

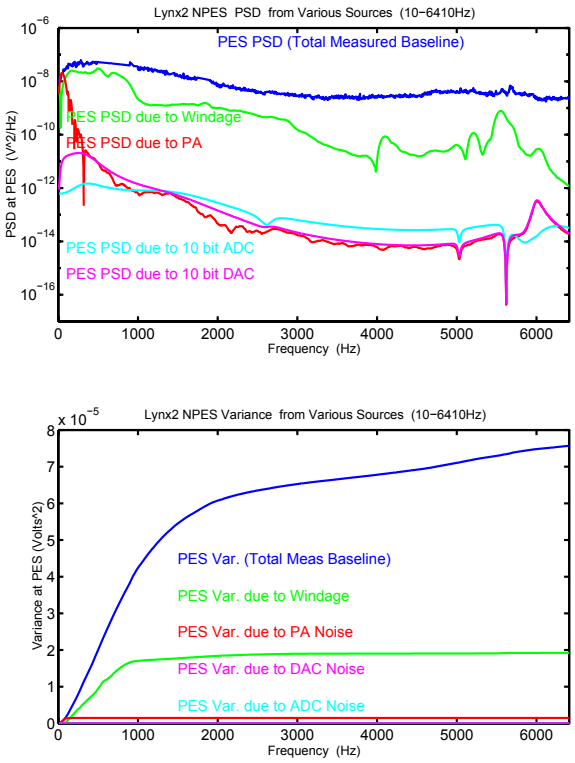

Fig. 26. Decomposition of baseline noise sources in a hard disk. Independently plotted to show relative importance. The top plot shows the PSDs. The bottom plot has them integrated across frequency.

> A theory is a good theory if it satisfies two requirements: It must accurately describe a large class of observations on the basis of a model that contains only a few arbitrary elements, and it must make definite predictions about the results of future observations. — *Stephen Hawking in "A Brief History of Time[43]"*
>
> Science, therefore, for all the reasons above, is not what it appears to be. It is not objective and impartial, since every observation it makes of nature is impregnated with theory. Nature is so complex and so random that it can only be approached with a systematic tool that presupposes certain facts about it. Without such a pattern it would be impossible to find an answer to questions even as simple as 'What am I looking for?' *James Burke in "The Day the Universe Changed[44]"*

If we have a good model, and we have made all these measurements of different noises, we are now in a position to put them all together at the error measurement. We use our best models to filter the noises from their backed out sources to their effect on the error (with appropriate unit conversions). There are two reasonable ways to think of this. One can compare them independently to see which noise sources are the dominant ones that we should worry about. This is shown in Figure 26.

Once the potential contributors to the error signal are identified, they can be ranked in terms of their overall effect on error and thus the most critical ones can be worked on first. In our HDD example, Figure 26 shows these independent noise sources and what is telling is not only how little the ADC and DAC quantization contribute to the overall error in this system, but how much of it is due to two sources, the disturbance of windage buffeting the drive head and the actual sensing of the position error. The former led to Terril Hurst working on redesigning the air flow inside of hard disks before HP exited the business in 1996. This latter fact, that close to 2/3 of the baseband noise in the PES was due to sensing errors led to the work on improving the demodulation circuits for HDDs [45], [46], [47]. While improving the noise in the actual magnetic domains from which position was derived might have taken considerable effort, providing smarter electronics to cleanly process these signals was an inexpensive and dramatic improvement. The benefits of understanding demodulation are discussed in the tutorial of [10].

To be clear, as quantization is nonlinear, the fundamental assumptions of the Widrow model assumes that quantization is fine enough to model the error as noise. These results argued against worrying whether to add extra bits to the next generation of of ADCs and DACs. On other systems, they may be the dominant noise sources.

VII. Using Pareto Models for Extrapolation

The quality of the results allow the effort to be put on the sources that most significantly limit the servo loop performance. Beyond analyzing the key sources of noise, present in a current system, the PES Pareto method, because it gave us noises as open-loop inputs and gave us a method tho add them independently at the error signal, allowed us to model the amplification or attenuation of any of these noise sources. Whether we could make a measurement (say of increasing the air flow by spinning the disk at a higher rotational speed) or simply increasing or decreasing a noise in a particular band of its input or across all frequencies, we could see what that change would have on our control loop.

For windage (or more generally the disturbance into the plant) we were able to change the level and measure the new result with the external sensor (the LDV). This is shown in the results of Figure 27. On top is the measurement at the LDV. The middle plot gives the effects of the different windage levels on PES. The lower plot integrates these to show the variance. Those variances are summarized in Figure 28. In the case of the Position Sensing Noise (PSN) we could scale that signal up and down, as shown in Figure 29. The plots show the potential benefit of dramatically reducing the PSN, which motivated the work on advanced demodulation.

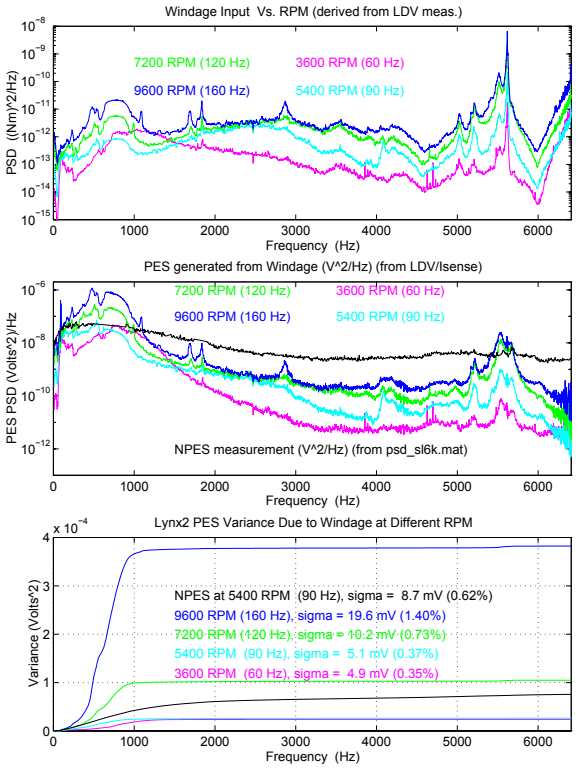

Fig. 27. Effects of changing spindle RPM. On top is the measurement at the LDV. The middle plot gives the effects of the different windage levels on PES. The lower plot integrates these to show the variance.

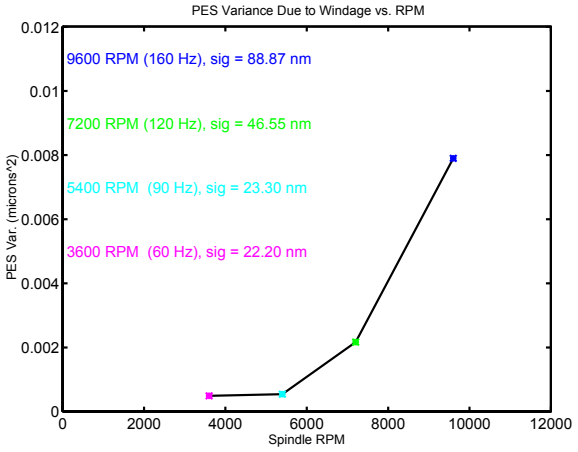

Fig. 28. Integrated PES PSDs due to Windage.

VIII. Minimizing Noise Before It Enters the Loop

One of the take aways from Stein's explanation of Bode's Integral Theorem [7] and the discussion above is that since attempting to suppress noise at one frequency range will amplify it in another range, it makes sense to try to minimize the noise before it enters the loop.

One way to manage this is through greater use of feedforward control when possible. If most of the reference signal following can be accomplished with a feedforward component [16], [17], [18] (which does not amplify any sensor noise), then the feedback loop can be optimized to minimize the effects of disturbance. Likewise, if external

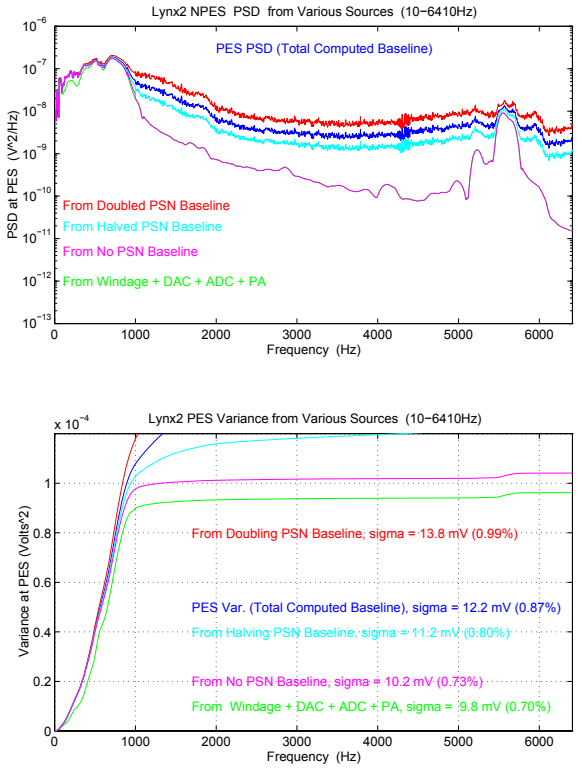

Fig. 29. Effects of changing Position Sensing Noise (PSN). The top plot gives the effects of the different PSN levels on PES. The lower plot integrates these to show the variance.

disturbances can be sensed [11], [12], this also changes the constraints of the feedback system design. Another possibility is that many of the sensor signals feeding a control loop require demodulation and being careful about the demodulation methods used can dramatically lower the noise that is allowed into the feedback loop [10],

IX. Summary

For the purposes of feedback control, once we have done everything else right, we are limited [5] by latency around the loop and by the noise that Bode's Integral Theorem tells us we cannot completely eliminate [7]. It follows then that a key to understanding the limits of performance of our control system then is to understand the noise environment in which we operate. The PES Pareto methodology allows us to isolate different noises as inputs to the system and then to combine them at specific measurement points – such as the error signal – so that we can rank the main contributors. While the method depends upon a few assumptions of linear analysis and the ability to measure and isolate PSDs of different noise sources, it is inherently practical and flexible, making use of whatever measurements are available. As such, the method is applicable to a wide variety of control systems that lend themselves to frequency domain analysis. The ability to scale these independent noise sources, and see the effects of this at different points around the loop, allow us to localize the key noise sources that should be the focus of engineering design effort. We hope that this is helpful.

REFERENCES

[1] D. Abramovitch, T. Hurst, and D. Henze, "An overview of the PES Pareto Method for decomposing baseline noise sources in hard disk position error signals," *IEEE Transactions on Magnetics*, vol. 34, no. 1, pp. 17–23, January 1998.

[2] ——, "The PES Pareto Method: Uncovering the strata of position error signals in disk drives," in *Proceedings of the 1997 American Control Conference*, AACC. Albuquerque, NM: IEEE, June 1997, pp. 2888–2895.

[3] T. Hurst, D. Abramovitch, and D. Henze, "Measurements for the PES Pareto Method of identifying contributors to disk drive servo system errors," in *Proceedings of the 1997 American Control Conference*, AACC. Albuquerque, NM: IEEE, June 1997, pp. 2896–2900.

[4] D. Abramovitch, T. Hurst, and D. Henze, "Decomposition of baseline noise sources in hard disk position error signals using the PES Pareto Method," in *Proceedings of the 1997 American Control Conference*, AACC. Albuquerque, NM: IEEE, June 1997, pp. 2901–2905.

[5] D. Y. Abramovitch, "Trying to keep it real: 25 years of trying to get the stuff I learned in grad school to work on mechatronic systems," in *Proceedings of the 2015 Multi-Conference on Systems and Control*, IEEE. Sydney, Australia: IEEE, September 2015, pp. 223–250.

[6] G. Stein, "Respect the unstable," December 1989, bode Lecture presented at the 1989 IEEE Conference on Decision and Control, Tampa FL.

[7] ——, "Respect the unstable," *IEEE Control Systems Magazine*, vol. 23, no. 4, pp. 12–25, August 2003.

[8] R. N. Bracewell, *The Fourier Transform and Its Applications*, 2nd ed. New York: McGraw-Hill, 1978.

[9] S. A. C. Doyle, *The Sign of Four*. London: Lippincott's Monthly Magazine, 1890.

[10] D. Y. Abramovitch, "The Demod Squad: A tutorial on the utility and methodologies for using modulated signals in feedback loops," in *Proceedings of the 2020 IEEE Conference on Control Technology and Applications*, IEEE. Montreal, Canada: IEEE, August 2020.

[11] D. Abramovitch, "Rejecting rotational disturbances on small disk drives using rotational accelerometers," in *Proceedings of the 1996 IFAC World Congress*, IFAC. San Francisco, CA: IEEE, July 1996, pp. 483–488 (Volume O).

[12] D. Y. Abramovitch and G. Hsu, "Mitigating the effects of disturbances of a disk drive," in *Proceedings of the 2015 Multi-Conference on Systems and Control*, IEEE. Sydney, Australia: IEEE, September 21-23 2015, pp. 1473–1478.

[13] A. Sacks, M. Bodson, and W. Messner, "Advanced methods for repeatable runout compensation (disc drives)," in *Digests of The Magnetic Recording Conference 1994*. Pittsburg, PA: IEEE, August 1994.

[14] M. Bodson, A. Sacks, and P. Khosla, "Harmonic generation in adaptive feedforward cancellation schemes," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1939–1944, September 1994.

[15] D. Y. Abramovitch, "A tale of three actuators: How mechanics, business models and position sensing affect different mechatronic servo problems," in *Proceedings of the 2009 American Control Conference*, AACC. St. Louis, MO: IEEE, June 10-12 2009, pp. 3357–3371.

[16] L. Y. Pao, J. A. Butterworth, and D. Y. Abramovitch, "Combined feedforward/feedback control of atomic force microscopes," in *Proceedings of the 2007 American Control Conference*, AACC. New York, NY: IEEE, July 11–13 2007, pp. 3509–3515.

[17] H. Perez, Q. Zou, and S. Devasia, "Design and control of optimal feedforward trajectories for scanners: Stm example," in *Proceedings of the 2002 American Control Conference*, AACC. Anchorage, AK: IEEE, May 2002, pp. 2305–2312.

[18] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *ASME Journal of Dynamic Systems, Measurement, and Control*, March 1987.

[19] L. Y. Pao and K. E. Johnson, "Control of wind turbines," *IEEE Control Systems Magazine*, vol. 31, no. 2, pp. 44–62, April 2011.

[20] J. S. McAllister, "The effect of disk platter resonances on track misregistration in 3.5 inch disk drives," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1762–1766, May 1996.

[21] ——, "Characterization of disk vibrations on aluminum and alternate substrates," *IEEE Transactions on Magnetics*, vol. 33, no. 1, p. 968, May 1996.

[22] ——, "Disk flutter: Causes and potential cures," *Data Storage*, vol. 4, no. 6, pp. 29–34, May/June 1997.

[23] J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*, 2nd ed. New York, NY: John Wiley & Sons, 1986.

[24] H. Solo, "Musings on mystical energy fields," in *Star Wars, Episode IV*, G. Lucas, Ed., Lucasfilm. A galaxy far away from here: 20th Century Fox, 1977.

[25] H. W. Bode, *Network Analysis and Feedback Amplifier Design*. New York: Van Nostrand, 1945.

[26] C. Mohtadi, "Bode's integral theorem for discrete-time systems," *Proceedings of the IEE*, vol. 137, no. 2, pp. 57–66, March 1990.

[27] D. Y. Abramovitch, "Built-in stepped-sine measurements for digital control systems," in *Proceedings of the 2015 Multi-Conference on Systems and Control*, IEEE. Sydney, Australia: IEEE, September 2015, pp. 145–150.

[28] *3585A Spectrum Analyzer Operating Manual*, Part number 03585-90003 ed., Hewlett Packard, February 1979.

[29] *CXA X-Series Signal Analyzer, Multi-touch N9000B: 9 kHz to 3.0, 7.5, 13.6, or 26.5 GHz Data Sheet*, Part number 5992-1274en ed., KeySight, February January.

[30] *8566B Spectrum Analyzer: 100 Hz–2.5GHz/2-22GHz, Manual*, Part number 08566-90040 ed., Hewlett Packard, March 1984.

[31] H. W. McKinney, "Band-selectable fourier analysis," *Hewlett-Packard Journal*, vol. 64, pp. 20–24, April 1975.

[32] N. Thrane, "ZOOM-FFT," *Brüel & Kjær Technical Review*, no. 2, pp. 3–43, 1980.

[33] B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory," *IRE Transactions on Circuit Theory*, vol. 3, pp. 266–276, 1956.

[34] R. G. Lyons, "Reducing ADC quantization noise," *Microwaves and RF Magazine*, pp. 10–21, June 2005.

[35] W. Kester, Ed., *Linear Design Seminar Notes*, 1st ed. Norwoood, MA: Analog Devices, Inc., 1995.

[36] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 379–423,623–656, 1948.

[37] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Menlo Park, California: Addison Wesley Longman, 1998.

[38] Wikipedia. (2020) Parseval's theorem. [On line; accessed May 22, 2020]. [Online]. Available: https://en.wikipedia.org/wiki/Parseval's_theorem

[39] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, 1985.

[40] R. Loughridge and D. Y. Abramovitch, "A tutorial on laser interferometry for precision measurements," in *Proceedings of the 2013 American Control Conference*, AACC. Washington, DC: IEEE, June 17–19 2013.

[41] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. 1010 Vermont Ave., N. W., Washington, D.C. 20005: Hemisphere Publishing Co., 1975.

[42] M. Spock, "Musings on mystical energy fields," in *Star Trek, VI: The Undiscovered Country*, L. Nimoy, L. Konner, and M. Rosenthal, Eds., Star Fleet. Alpha Quadrant: Paramount Pictures, 1991.

[43] S. Hawking, *A Brief History of Time*. New York, NY: Bantam Dell Publishing Group, 1988, iSBN: 978-0-553-10953-5.

[44] J. Burke, *The Day the Universe Changed*. Boston and Toronto: Little, Brown and Company, 1985.

[45] D. Abramovitch, "Customizable coherent servo demodulation for disk drives," *IEEE/ASME Transactions on Mechatronics*, vol. 3, no. 3, pp. 184–193, September 1998.

[46] ——, "Customizable coherent servo demodulation for disk drives," in *Proceedings of the 1998 American Control Conference*, AACC. Philadelphia, PA: IEEE, June 1998, pp. 3043–3049.

[47] D. Y. Abramovitch, "Disk drive servo demodulation system which suppresses noise on the position error signal," Hewlett-Packard, Palo Alto, CA USA, United States Patent 5,801,895, September 1 1998.