2021 American Control Conference (ACC)
New Orleans, USA, May 25-28, 2021

Determining Kalman Filter Input Noises Using PES Pareto

Daniel Y. Abramovitch*

Abstract— **One of the key questions that befalls anyone about to design a Kalman Filter is the question of determining the noise inputs. As methodical and systematic as Kalman filter design is, it is wholly dependent upon the choice of noise source values, and these are usually ad-hoc. On the other hand, the PES Pareto Methodology [1] provides a systemic method for modeling and measuring broadband noises in a feedback loop. As with the old Reece's Cup commercials, it seems that one might want to know if these two techniques can be combined, so that we obtain well quantified process and sensor noise estimates based upon our PES Pareto based measurements. That is the subject of this paper.**

I. Introduction

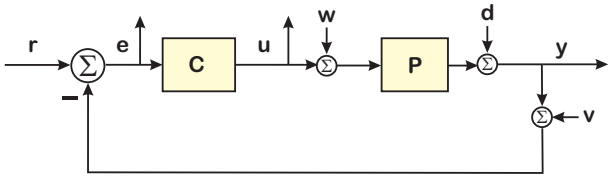

Fig. 1. Closed-loop system with a standard process (w) and measurement (v) noise sources.

Kalman Filters provide a well known way to modify a current mode estimator [2], [3], [4] so that the observer feedback gains are determined as a least-squares balance between the process noise and measurement noise. The process noise disturbs the plant, while the measurement noise is separate from the plant dynamics. Once a discrete-time, linear time-invariant model of the plant is determined, the difficulty is really about estimating the process (w) and measurement (v) noises in Figure 1. These are generally modeled as Additive White Gaussian Noise (AWGN) with a zero mean. Given the importance that is attributed to (w) versus (v), it would seem that there should be a more systematic way to quantify those values.

The PES Pareto methodology [1] provides a way of analyzing the sources of noise in a feedback loop. Originally conceived for analyzing noise contributors in magnetic hard disk drives [5], [6], [7], [8], 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 noise contributors.

This paper demonstrates how PES Pareto can use the quantified measurements of a variety of modeled noise inputs and map their effects as if they were contributing at either the w or v inputs to the loop. From that point, the cumulative Power Spectral Densities (PSDs) can be integrated over the relevant frequency range to give the effective noise covariances, W and V. The contribution of this paper is that the Kalman Filter is run based on accurate noise estimates derived from measurable quantities.

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

II. A Review of Basic Kalman Filtering

The Kalman Filter starts with an assumption that the model of the physical world that is linear, time-invariant, and in discrete time [2], [3], [4]. For a Single-Input, Single-Output (SISO) system,

$$x_{k+1} = Fx_k + Gu_k + G_W w_k \quad (1)$$
$$z_k = Hx_k + Du_k + v_k. \quad (2)$$

The matrices describe how the system evolves from one time step to the next. The terms, w_k and v_k, denote noises that affect the system, with w_k being a "process" noise that actually affects where the actual states will go, while v_k is a "measurement" noise that does not affect the actual states, but affects our ability to measure the output.

As used in feedback systems, a Kalman Filter takes the form of a Current Estimator. In this usage, we have access to the plant input generated by the controller, u_k, which we would not have in many signal processing formulations. For feedback systems where u_k is generated and therefore known, this results in the D term canceling out from the error dynamics. The Current Estimator Time Update is:

$$\bar{x}_{k+1} = F\hat{x}_k + Gu_k \quad (3)$$

while the Measurement Update is:

$$\begin{aligned} \bar{y}_k &= H\bar{x}_k + Gu_k \\ \hat{x}_k &= \bar{x}_k + L_C\left[z_k - \bar{y}_k\right] \end{aligned} \quad (4)$$

In the Kalman Filter we modify L_C to possibly vary every step, $L_{C,k}$ and it is chosen as a least squares balance between how the process noise, w_k, and the measurement noise, v_k, affect the measurement error. Here, W is the covariance matrix of the process noise, w_k, and V is the covariance of the measurement noise, v_k, i.e. $w_k \sim N(0, \sqrt{W})$ and $v_k \sim N(0, \sqrt{V})$. The key update to the current estimator is the propagation of uncertainty which is used to pick the feedback gain, $L_{C,k}$ at every step.

The Kalman Time Update is now:

$$\bar{x}_{k+1} = F\hat{x}_k + Gu_k \quad (5)$$
$$M_{k+1} = FP_kF^T + G_wWG_w^T \quad (6)$$

where (5) represents the state time update as before and (6) represents the uncertainty time update.

From a programming/simulation point of view, the Time Update projects forward one or more steps in time to the point of the next measurement. (For the Kalman Filter, one step in time.) Therefore at the time of the next measurement (the new k), the Kalman Measurement Update is:

$$\bar{y}_k = H\bar{x}_k + Gu_k, \quad (7)$$

$$P_k = M_k - M_k H^T \left(HM_k H^T + V\right)^{-1} HM_k, \quad (8)$$

$$L_{C,k} = P_k H^T V^{-1}, \quad (9)$$

$$\hat{x}_k = \bar{x}_k + L_{C,k}\left[z_k - \bar{y}_k\right], \quad (10)$$

$$\hat{y}_k = H\hat{x}_k + Gu_k. \quad (11)$$

Here (7) is the part of the Time Update that uses the projected state to estimate the incoming measurement. It could very well have been grouped with the previous equations, but from the use of the time indices, we want to be certain to understand that we form an error based on the predicted and actual measurement. The state Time Estimate is in (8), and the uncertainty update is in (8). We can also give our best estimate of the system output, $\hat{y}_k$, although this is not necessary for feedback control.

Likewise, the steady-state Kalman Filter is can be written using the same state update equations, but now we have steady-state values of uncertainty, $P_k = P_{SS} = P$ and $L_{C,k} = L_{C,SS} = L_C$ as:

$$P = FPF^T + W - FPH^T\left[HPH^T + V\right]^{-1} HPF^T \quad (12)$$

$$\text{and } L_C = FPH^T\left[HPH^T + V\right]^{-1}, \quad (13)$$

where (12) is an Algebraic Riccati Equation (ARE) [2]. For a given system model, the noise covariances, W and V, uniquely determine the feedback gains, whether they be evolving, $L_{C,k}$, or stead state, L_C.

The Kalman Filter is the least squares solution to a particular mathematical problem, but when one asks if it is optimal, the true answer is in the form a question: *How well did we model reality?* More specifically, *How accurate are* $\{F, G, G_w, H, D, W, \& V\}$*?* It is this author's experience that people usually pour a lot more effort into the optimization algorithm than they do in obtaining a good model [9]. This author is still confused about what utility optimizing on a poor model can have, and so this paper is intended to address part of that problem.

In particular, while we will not discuss improvements to the model parameters, we believe that the PES Pareto Methodology [1] for measuring and quantifying noise propagation through a feedback loop can be used to create measurement based estimates for W and V, which should improve at least half of this problem.

III. A Brief Review of PES Pareto

This section will review the fundamentals of the PES Pareto Method. The original papers can be found at [5], [6], [7], [8], while a recent tutorial was presented in [1].

The PES Pareto Method is based on three ideas: (1) an understanding of how Bode's Integral Theorem [10], [11], [12] 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 account for the strata of the servo loop's error signal. The method requires the measurement of frequency response functions (FRFs) 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.

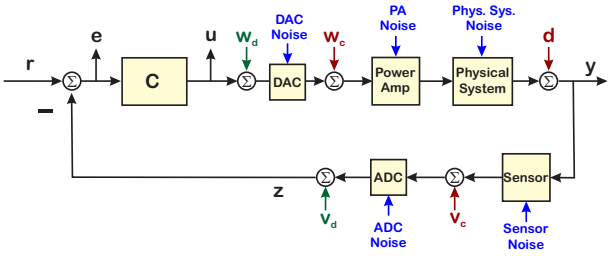

Fig. 2. Closed-loop system with each block having its own noise source.

In a more complex model where each block can be modeled to have its own noise source (Figure 2), how do we:

- 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 frequency bins that are consistent with the Frequency Response Functions (FRFs) generated from our system model.
- 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 Parsevall's Theorem [13]) 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 a potential source [14].

To quantify the noises in the loop more systematically, we would like to include more of the noise sources in our

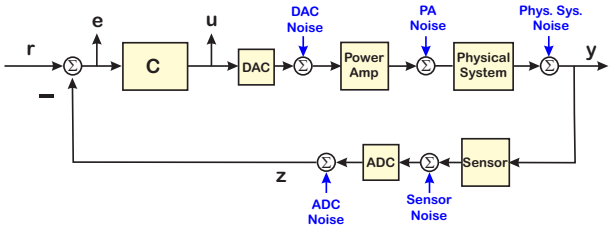

Fig. 3. Simplification for analysis of Figure 2. Closed-loop system with each block having its own noise source as an additive output noise.

block diagram, as shown in Figure 2. In this model we have included such sources of "noise" as quantizers, power amplifiers, broadband disturbances, and sensor noise. While more complete, this formulation is not overly tractable. A compromise model is diagrammed in Figure 3, in which the noises are now modeled as additive noise on the output of each block.

In PES Pareto, we choose a set of potential input noise sources. Knowing that AWGN is still considered additive and Gaussian after passing through a linear filter, we realize that we can often model noise measured inside a closed-loop system as a closed-loop reference input coming into the system, by applying the appropriate magnitude-squared filter at the same frequency points as the Power Spectral Density (PSD) of the noise measurement. Using a different magnitude-squared filter, we can then compute the effect of that noise source at different points along the loop. We can also make a rather weak assumption that the reference input noises are independent of each other, meaning that their cross-correlation and cross spectra will be 0. In practical terms, it means that multiple measurements need to be made, their auto and cross spectra computed, and these need to be averaged to approximate an expected value. This allows us to have a set of independent input noises characterized at a set of inputs around the loop. As these noise sources are independent, their filtered PSDs can be compared or added at any particular measurement point. For this paper, our interest will be in translating some of these noises to be associated with the process noise, w, and some to be associated with the measurement noise, v.

Given that we have decided which noises we wish to quantify, say from a diagram such as Figure 3, we need to find a measurement or set of measurements that can give us an isolated measurement of that source. As mentioned in the tutorial paper, [1], this is by far the most ad-hoc part of the method, particularly when the measurements are not built into the digital controller itself. For brevity will not spend too much space on this aspect [1], except to give a handful of useful insights here:

1) The measurement point is often downstream in the loop from the closed-loop injection point. Generally, the closer to the closed-loop injection point, the cleaner the measurement.
2) The point of a measurement is to generate a Power Spectral Density (PSD) of the isolated noise signal (with physical unit scaling) from DC to the Nyquist frequency ($0 \longrightarrow \frac{f_S}{2}$). To evaluate the noise content at other points along the loop, it must be filtered by the magnitude-squared frequency response of the appropriate frequency domain model. The magnitude-squared filter must have the same frequency bins as the computed PSDs of the noise measurements.
3) Some measurements can be made with the loop open, and so translating them to a different reference input along the loop does not involve closed-loop dynamics. Many measurements require the loop to be closed and so a translation from a measured closed-loop signal to an independent reference input involves inverse filtering with a magnitude-squared filter that has the inverse of the closed-loop dynamics.

A. Useful PSDs from Measurements

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

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt, \tag{14}$$

If $x(t)$ is sampled N times over a span, T, with a sample period of Δt, then $x_n = x(n\Delta t)$. By letting $f_k = \frac{k}{T} = \frac{k}{N\Delta t}$ we get

$$X_k = X(k) = \frac{X(f_k)}{\Delta t} = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}. \tag{15}$$

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

$$X(f_k) = \Delta t X(k) \tag{16}$$

which returns the FFT to something closer to the physical units. We want physical units when measuring signals in the lab but they do not have arbitrary scaling. We often need to learn the scaling of a particular instrument or software tool, so we can scale everything in the same way. Given that physical units stay consistent, it is a good idea to map everything to them.

As we are backing noises out to the point where they are independent inputs, we consider the Cross Spectral Densities (CSDs) of these noises to be 0. PES Pareto is really about Power Spectral Density (PSD). To produce the PSD from $X(f)$, we compute the complex conjugate $X^*(f)$. At that point we compute the element by element product of the two complex vectors:

$$PSD(x) = \frac{X^*(f)X(f)}{B_e} \tag{17}$$

where B_e is the Resolution Bandwidth of the filter used to compute the spectrum and where $X(f)$ is the Fourier Transform from Equations 14–16. This is the smallest change in frequencies that a given measurement can resolve and generally is inversely proportional to the length of the measurement time window, T. For an FFT,

$$B_e = \frac{1}{T} = \frac{1}{N\Delta t} \tag{18}$$

where N is the number of points in the FFT and Δt is the sample period between points, so that we have the

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

Several other ways of obtaining PSDs from measurements and using other values of B_e are discussed in [1].

IV. Combining the Methods

Given that a set of noises have been measured and backed out to their reference inputs, and given that we can consider the reference inputs to all be statistically independent, we can now combine them to form the reference noise inputs of w and v.

The key judgment comes in which noises to combine at w and which to be combined at v. If we look at Figures 3 and 1, we need to first understand what to do with the physical system disturbances (labeled d in Figure 1 and Phys. Sys. Noise in Figure 3). We can first divide this into any part of d that can be measured using a sensor or seen as a deterministic input. This would clearly not fall into the KF's assumption of Additive, White, Gaussian Noise (AWGN) and should not be part of the w or v noises. However, if there is a portion that is or looks random, then we must make another distinction about whether we consider it to be altering the behavior of the plant or whether it simply disturbs the measurement.

Anything that disturbs the plant should be transformed as part of the process noise input, w. This would include not only any relevant portions of d, but also the DAC noise and Power Amp noise in Figure 3. The air flow disturbance described in the earlier PES Pareto papers would also be part of this process noise. Anything that only affects the measurement would be assigned to the measurement noise input, v. Sensor noise, demodulation noise, input electronics noise (which often shows up as noise above the quantization noise in the ADC), and ADC noise would go into this term.

One more distinction needs to be made: PES Pareto involves measuring and magnitude-squared filtering PSDs, while the Kalman Filter is looking for noise variance (or covariance) quantities. The former is a far richer, more detailed trove of data, but we can reduce the cumulative noise input PSDs at w and v by integrating each over the relevant frequency range. This is most likely over the frequency bins from DC (0 Hz) to Nyquist ($\frac{f_S}{2}$). This gives the variance in frequency for each of these noises and then we know from Parseval's Theorem that these equal the noise variance in time for each of these [13].

We can make a few assumptions that simplify our explanation here. We will assume that we can model the quantization in an ADC or a DAC via the Widrow model [15] and spread the variance across the frequency band from $-\frac{f_S}{2}$ to $-\frac{f_S}{2}$ [16], [17]. As we will only measure from 0 to $\frac{f_S}{2}$ we will double the variance to stay consistent, so the nominal PSD due to quantization level, q is:

$$\Phi_{QQ}(f) = \frac{q^2}{6f_S} = \frac{q^2 T_S}{6}, \text{ using } f \in [0, \frac{f_S}{2}], \tag{20}$$

and is flat across frequencies from DC to $f_S/2$[18]. However, we have seen that we often measure fewer bits of resolution than what is quoted via the converter number of bits, and we can scale the quantities in (20) to represent an equivalent number of quantizer bits [1]. We will assume that apart from these scaled quantization noises, the gains of ADC and DAC are 1, although we could get equivalent results with longer equations by assuming $K_{ADC} = 1/K_{DAC}$. We will assume that in the Nyquist band, the ADC and DAC have a flat response.

For clarity and brevity, this paper assumes that the power amplifier has a response, $I(f)$, that is flat out far beyond the Nyquist frequency as does the sensor response, $M(f)$. Here, we let $I(f) = K_{PA}$ and $M(f) = K_M$ in the frequencies of interest. Setting the plant and controller models to $P(f)$ and $C(f)$, the closed-loop denominator used in forward filtering any independent references is given by:

$$\|\Delta(f)\|^2 = \|1 + K_M K_{PA} K_{ADC} K_{DAC} P(f) C(f)\|^2. \tag{21}$$

Consider if we could measure the spectrum of the disturbance noise, d, at y in Figure 3, ($\Phi_{dd,y}(f)$), but only in closed-loop. To get that back to an independent reference input at w, we would filter the measured PSD by:

$$\left\| \frac{1 + K_M K_{PA} K_{ADC} K_{DAC} P(f) C(f)}{K_{PA} K_{DAC} P(f)} \right\|^2. \tag{22}$$

Similarly, if we could measure the power amplifier noise by zeroing out the DAC output, thereby opening the loop. As this is an open loop measurement we do not need the closed-loop dynamics and merely need to scale our measured quantity, ($\Phi_{ii,i}(f)$), by $\frac{1}{(K_{DAC} K_{PA})^2}$ to get it back to w. Our DAC, with quantization level q and unity gain is an input quantity right at w, so no inverse filtering is needed. Thus, we can related these three noise inputs, d, DAC noise, and Power Amplifier noise, back to the traditional process noise input, w.

In measuring the components of v we have assumed that we would include the ADC noise, using a scaled Widrow model (20), and assume that this is already generated as a reference input at the same point as v. The sensor noise has to go through the ADC, so it is affected by the ADC gain, K_{ADC}. It is often the case that we cannot assess the sensor noise without the loop being closed, [1], and so we assume that we measure it at e with reference input, r that is known (and can be subtracted off of the measurement). Often, we determine the sensor noise by eliminating all other sources at e, but in this example we will assume that we have measured it at z in closed-loop ($\Phi_{mm,z}(f)$), and so must get to the noise PSD as a reference input, $\Phi_{mm,m}(f)$ via

$$\left\| \frac{1 + K_M K_{PA} K_{ADC} K_{DAC} P(f) C(f)}{K_{ADC}} \right\|^2, \tag{23}$$

but to shift this over to v, we would have to forward filter the results filtered by (23) by $\|K_{ADC}\|^2$.

We can now add the constituent PSDs at both the w and v inputs – as they are independent noise sources – and finally have a noise spectra PSD input at each of these

injection points. The Kalman Filter does not need noise spectra, though, only the noise variance. Recalling Parseval's Theorem [13]). the noise variance in time,

$$\sigma_w^2 = \int_0^{f_S/2} \Phi_{ww}(f)df \approx \sum_{k=0}^{NFFT-1} \Phi_{ww}(f_k)\Delta f, \text{ and} \tag{24}$$

$$\sigma_v^2 = \int_0^{f_S/2} \Phi_{vv}(f)df \approx \sum_{k=0}^{NFFT-1} \Phi_{vv}(f_k)\Delta f. \tag{25}$$

Obviously, things could be much more complicated than the above map, but the key point is that PES Pareto provides a systematic way to go from noise measurement PSDs to the critical noise variances required by the Kalman Filter.

V. A Design Example

In this section, we will demonstrate the method using a simple but illustrative example. For a simple plant we choose a double integrator, $P(s) = \frac{K}{s^2}$. Our controller needs phase lead. so we will chose a discretized Proportional Plus Derivative (PD) controller. One of the first things that we need to do is make an adjustment to the plant. Normally, we evaluate the FRF of a plant with integrators on a logarithmic frequency scale, so the infinite values at DC are never an issue. When computing the PSDs for PES Pareto, we need to use a linear frequency scale from 0 – $\frac{f_S}{2}$, so that we cannot avoid the infinite DC gain. Instead, we replace the integrators with low pass filters containing extremely low frequency poles. Thus, our plant model is

$$P(s) = \frac{K}{s^2} \approx \frac{K}{(s + a_{int})^2} \tag{26}$$

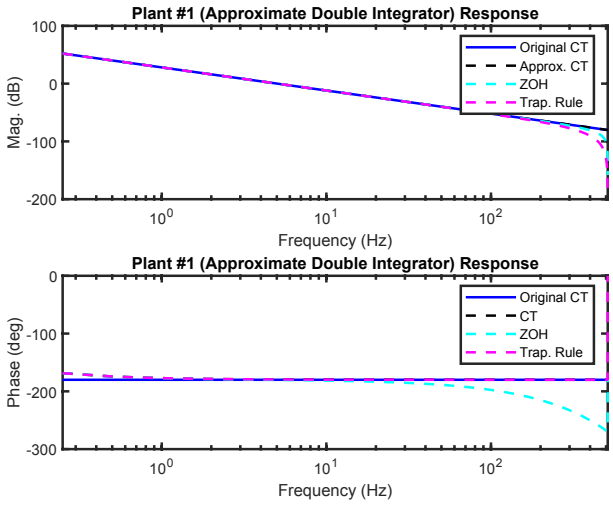

Fig. 4. Double integrator and finite gain approximation.

In Figure 4 we replace each integrator with a low pass pole at 0.1 Hz. We have chosen a sample rate of 1024 Hz and 4096 frequency bins for the FFTs so that the frequency bins tile into integer frequency spans. We have the finite DC gain we need for the filtering of PSDs, but the differences between the different continuous time curves is negligible and the discretized responses, both Zero-Order Hold and Trapezoidal

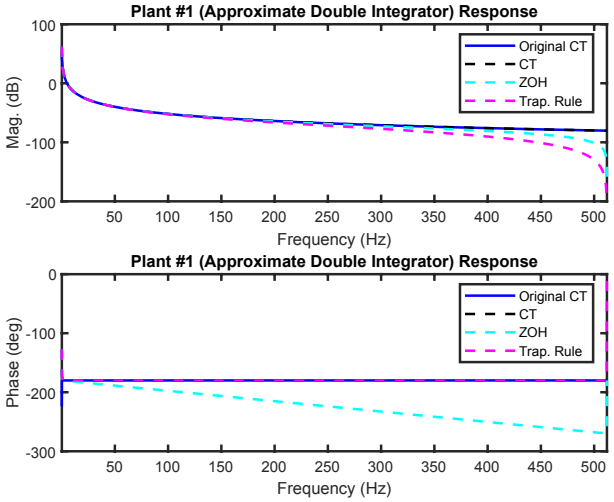

Fig. 5. Double integrator and finite gain approximation, plotted on a linear frequency axis.

Rule equivalents, are also very close. The same responses look very different in Figure 5 when they are plotted on the linear frequency scale we use in our spectrum analysis. Our PD controller – discretized with the backwards rectangular rule equivalent used almost universally for PID controllers [19] – is:

$$C_{BR}(z) = (K_P + K_D)\left[\frac{z - \frac{K_D}{K_P+K_D}}{z}\right]. \tag{27}$$

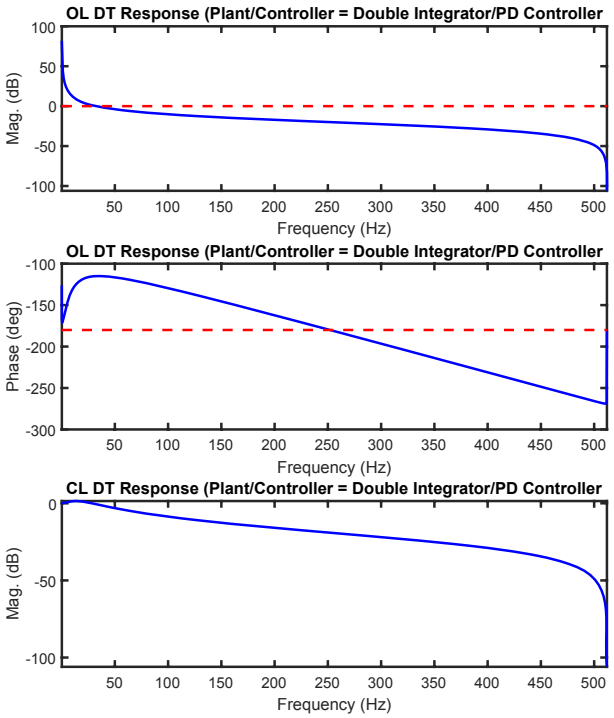

Fig. 6. Open and closed-loop response of $P(z)C(z)$, plotted on the linear frequency axis used for noise measurements and analysis.

Most readers will be very familiar with open and closed-

loop frequency response functions of double integrators stabilized with a lead, but are probably less familiar in seeing these plotted on a linear frequency scale, as shown in Figure 6. This provides an interesting perspective, because while we often focus on lower frequency dynamics, broadband noise factors in across much of the frequency range beyond our closed-loop bandwidth. The linear plots help to bring that into focus.

We will apply the simplifying assumptions of Section IV for clarity. We model the power amplifier and sensor bandwidths far above the Nyquist frequency so that we can use a model of simple gains for K_{ADC}, K_{DAC},K_{PA}, and K_M. The quantization noises and power amplifier noises are considered flat out to Nyquist. To make it interesting, we will shape the disturbance noise and the sensor noise.

A logical assumption is that the disturbance would be a physical phenomena and therefore low pass and therefore could be modeled as AWGN filtered by a low pass filter to generate the disturbance input spectrum. For this example, the input shaping filter is a discretized second order low-pass filter with a natural frequency of $f_{N,d} = 40Hz$, a damping factor of $\zeta_{N,d} = 2$, and a DC gain of 0.1. Applying a magnitude squared version of this filter to an AWGN noise variance of $\sigma^2_{dist} = 0.01$ spread from 0 – $\frac{f_S}{2}$ results in a shaped PSD at the disturbance input (the **green curves** in the upper plots of Figures 7 and 9). Integrating these curves from 0 to $\frac{f_S}{2}$ yields $\sigma^2_{dist,in} = 19.152 \cdot 10^{-6}$.

Similarly, we can assume the sensor noise to be high pass, and therefore could be modeled as AWGN passed through a high pass filter to generate the sensor input spectrum. In this example, we chose a simple lead with the zero at $f = 5Hz$, the pole at $f = 50Hz$, and the DC gain of 0.001. Applying a magnitude squared version of this filter to an AWGN noise variance of $\sigma^2_{sense} = 0.01$ spread from 0 – $\frac{f_S}{2}$ results in a shaped PSD at the sensor input (the **blue curves** in the lower plots of Figures 7 and 9). Integrating these curves from 0 to $\frac{f_S}{2}$ yields $\sigma^2_{sense,in} = 5.3764 \cdot 10^{-6}$. The plots of the **ADC noise**, **PA noise**, and **DAC noise** are – because of our simplifying assumptions for these first example – flat. The fact that some of the noises do not have flat PSDs means that their relative importance as contributors to w and v cannot be known until they are magnitude squared filtered to that point and then integrated to generate variances.

In the first example, a 12-bit ADC and DAC are chosen. The input noises that feed w, that is noise in the DAC, the power amplifier, and the disturbance, are shown in Figure 7. By the time they are magnitude squared filtered to the inputs we need at w and v in Figure 8, their relative importance has changed. The individual source PSDs can be summed at each of the "Kalman inputs" resulting in the black dashed curves. Note that for w the disturbance noise is the key component, despite falling below the power amp noise and DAC noise for much of the frequency span. For v, the sensor noise dominates the overall noise PSD for most of the frequency span. While the specific numbers in this example are not important, we can note that $V = \sigma^2_v$ is an order of magnitude higher than $W = \sigma^2_w$.

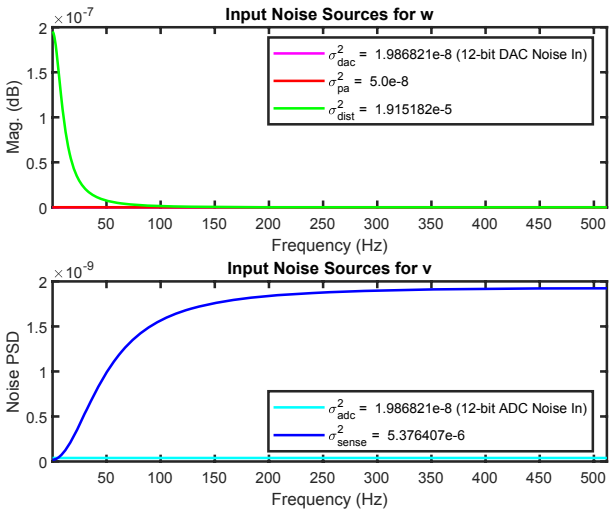

Fig. 7. Input noise reference PSDs that feed w (top) and v (bottom) for 12-bit ADCs and DACs.

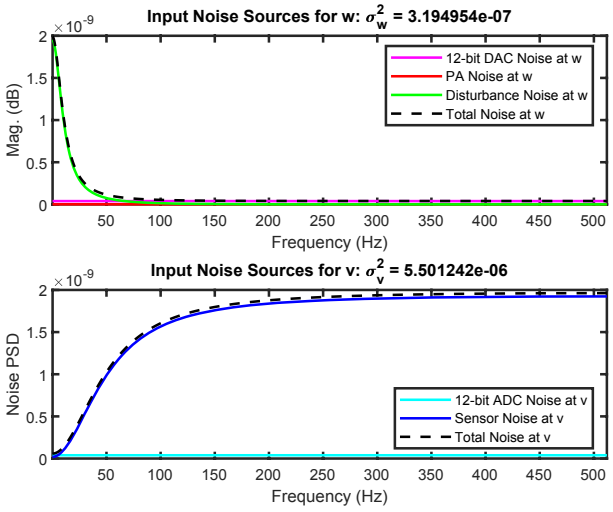

Fig. 8. Input noise reference PSDs mapped to w (top) and v (bottom) for 12-bit ADCs and DACs.

In the second example, we merely reduce the number of bits in the ADC and DAC from 12 to 10, shown in Figure 9. This small change, once filtered around to w and v result in considerably different looking w and v PSDs, seen in Figure 10. For w, the DAC noise becomes a far more dominant signal, while for v the ADC noise is no longer trivial, but still not the dominant noise source. Because of how the different noise inputs are shaped in being mapped to w and v, the net effect is a disproportionate scaling of $W = \sigma^2_w$ and $V = \sigma^2_v$, so that W is now only a third of V. Since the Kalman Filter feedback gain matrix is dependent upon W and V, we see how these small changes in input noises at their sources can be mapped to different Kalman feedback gains.

These examples are simple, but they do illustrate the method. Furthermore, they demonstrate that because of where the different noise inputs are and how they follow different paths to their respective "Kalman inputs", changing

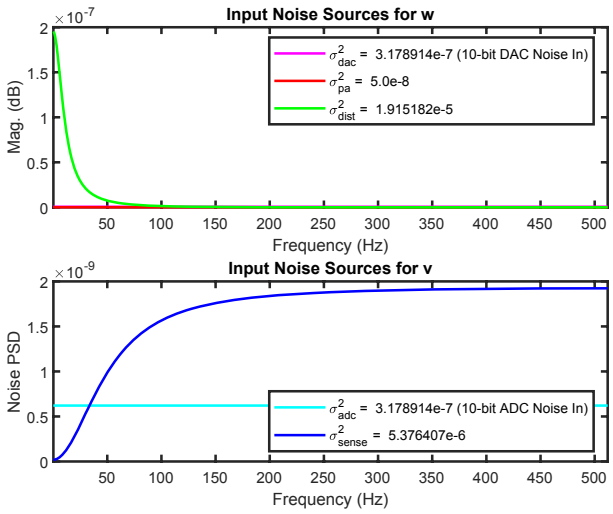

Fig. 9. Input noise reference PSDs that feed w (top) and v (bottom) for 10-bit ADCs and DACs.

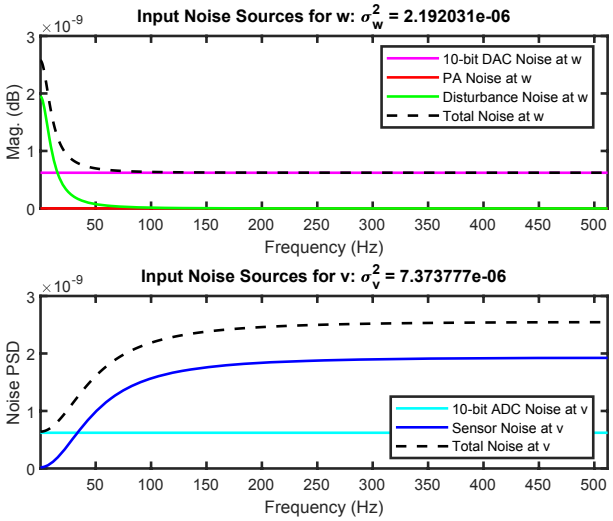

Fig. 10. Input noise reference PSDs mapped to w (top) and v (bottom) for 10-bit ADCs and DACs.

something as simple as the flat PSDs of the ADC and DAC noises can affect w and v in dramatically different ways, thereby altering both the uncertainty propagation and the gains of the Kalman Filter.

VI. Conclusions

This paper has shown how to use the noise analysis in PES Pareto to map a variety of noise sources around a loop into the process and sensor noises used for determining (along with the system model) the gains of a Kalman Filter. The process described in Section IV was demonstrated in two simple examples in Section V to show that once the system model is known and the input noises have been quantified (in terms of PSDs) we can filter this to show up as variances at w and v. Because of the different filters that each noise must pass through to arrive at it's "Kalman input", small changes in flat spectrum input noises can dramatically change the relative importance of W and V. The user now has a method for extracting these from measurements.

This can be applied to systems with much more complex block models and noise inputs while maintaining the same essentially methodology. A similar application could be made to the continuous-time Kalman-Bucy Filter [2], with the caveat that as we invariably measure noise spectra using discrete-time tools, we must be careful to properly translate those measurements to continuous-time spectral models.

References

[1] D. Y. Abramovitch, "A tutorial on PES Pareto methods for analysis of noise propagation in feedback loops," in *Proceedings of the 2020 IEEE Conference on Control Technology and Applications*, (Montreal, Canada), IEEE, IEEE, August 2020.

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

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

[4] T. Kailath, *Linear Systems*. Englewood Cliffs, N.J. 07632: Prentice-Hall, 1980.

[5] 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, pp. 17–23, January 1998.

[6] D. Abramovitch, T. Hurst, and D. Henze, "The PES Pareto Method: Uncovering the strata of position error signals in disk drives," in *Proceedings of the 1997 American Control Conference*, (Albuquerque, NM), pp. 2888–2895, AACC, IEEE, June 1997.

[7] 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*, (Albuquerque, NM), pp. 2896–2900, AACC, IEEE, June 1997.

[8] 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*, (Albuquerque, NM), pp. 2901–2905, AACC, IEEE, June 1997.

[9] 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*, (Sydney, Australia), pp. 223–250, IEEE, IEEE, September 2015.

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

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

[12] G. Stein, "Respect the unstable," *IEEE Control Systems Magazine*, vol. 23, pp. 12–25, August 2003.

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

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

[15] 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.

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

[17] W. Kester, ed., *Linear Design Seminar Notes*. Norwoood, MA: Analog Devices, Inc., first ed., 1995.

[18] J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*. Wiley Series on Probability and Statistics, New York, NY: John Wiley & Sons, third ed., 2000.

[19] D. Y. Abramovitch, "A unified framework for analog and digital PID controllers," in *Proceedings of the 2015 Multi-Conference on Systems and Control*, (Sydney, Australia), pp. 1492–1497, IEEE, IEEE, September 2015.