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

The Demod Squad: A Tutorial on the Utility and Methodologies for Using Modulated Signals in Feedback Loops

Daniel Y. Abramovitch*

Abstract— **This paper is a brief tutorial on methods for using modulated signals in feedback loops, and especially of the different methods and trade offs used for demodulating those signals to get information for the control loop. Modulation and demodulation are often understood as being part of communication theory, but the methods used to demodulate control signals can have a dramatic effect on the amount of delay, noise, and nonlinearity that gets passed from the desired control information into the feedback loop itself. In this tutorial, we hope to give a more fundamental understanding of how to properly use demodulation to minimize "sensor noise" before it ever enters into the loop.**

I. Introduction

Understanding the utility and methodologies of using modulated signals in feedback loops is not common, and is usually limited to a particular type of modulation in a particular application. Seen primarily as belonging to communication theory, this subject gets little attention. Often only the simplest embodiments are presented, ignoring the potential noise, delay, and nonlinearities that these methods can introduce in the desired signal. While the simplest methods have the advantage of ease of implementation with simple analog circuits, they are decades behind the times. New digital (and analog) methods make practical a whole slew of coherent methods. Even then, these methods are often only understood from a communications perspective, where the lack of a feedback loop make the role of noise, delay, and nonlinearities in the sensor signal far less significant. To get the most out of our control loops, it is worth fundamentally understanding demodulation.

Still, whether by design or necessity, many of the signals that are used to mark position, velocity, and/or acceleration are modulated onto some carrier. The position signals in hard disk drives are in a pattern of alternating polarity magnetic domains. Motor control is often achieved via shaft encoders with alternating patterns around the circumference. Laser interferometers work by detecting the phase between a reference signal reflected off of a mirror at a fixed distance and a measurement signal reflected off of a mirror on a moving object.

Nature also makes use of modulation in the firing of neurons [1], [2] which "fire" not with a DC level, but with the presence or absence of a string of pulses. A higher level of firing corresponds to a higher frequency of the pulses, not a greater amplitude. As these pulses are all non-negative, demodulation of the neuron signal can be done simply with averaging. Why would nature choose to use a modulated signal? Modulating a signal makes it less susceptible to offsets and baseline noise. Baseband (DC) signals often can't travel far, encode biases and offsets, and are susceptible to drift.

Often, a modulated signal is the only way to encode a position or velocity measurement with sufficient Signal-to-Noise Ratio (SNR) for feedback control. Sometimes, we are looking for the effects of the system on the modulated signals. Sine-dwell (also known as swept-sine in industry) measurements of dynamic systems rely on the system's response to a particular sinusoid at a particular frequency. For atomic force microscope (AFM) measurements of soft biological samples [3], [4], [5], it's more advantageous to tap the probe tip across the surface using an AC drive signal, so that the surface experiences compression forces but minimal shear. In these problems, extracting the surface's affect on the cantilever oscillation is the key to characterizing the surface. The faster and more accurately an AFM can do this, the more effective the measurement [6], [7], [8], [9].

No matter what the original motivation, we often need to demodulate these signals to actually use them in a feedback loop. How we do this demodulation depends not only on the modulation method, but on the technology available to do the math. The advent of improved digital electronics has allowed for a far more exotic set of modulation/demodulation schemes. For use in feedback control loops, we must further consider the latency of the demodulation computation and the achievable SNR from a method. This paper will attempt to give a cohesive overview of the different demodulation methods as they are applied in feedback loops.

For the purposes of feedback control, once we have done everything else right, we are limited [10] by latency around the loop and by the noise that Bode's Integral Theorem tells us we cannot completely eliminate [11]. A practical linear analysis of noise through a loop can be accomplished using a PES Pareto methodology [12], but the final takeaway from that method is that one should pay attention to eliminating noise before it enters the loop. While the simplest, non-coherent, "slow and noisy" demodulation methods may suffice for a large set of important problems, there are problems for which doing a little bit more math – in the right way – can dramatically improve the signal to noise and dramatically lower the latency of the extracted signal. In other words, we can minimize some of the factors that limit the performance of our feedback loop. Those

*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

978-1-7281-7139-5/20/$31.00 ©2020 IEEE

factors are hard to limit from control design only.

II. Pulse Modulations

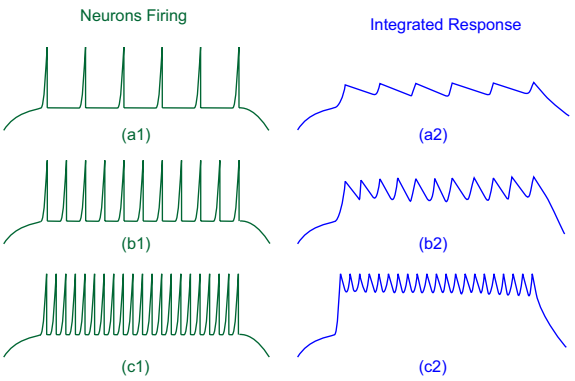

Fig. 1. Drawings of pulse modulation of the type used by neurons firing. On the left, we see that an increase in the input to a neuron causes an increase in the frequency (not the magnitude) of the pulse train. On the right, we see the simplest demodulation, a kind of averaging, that can be done on the pulse train to extract a lower frequency value.

One family of modulation schemes involves modulating pulses of fixed height, either by their position (Pulse Position Modulation, PPM) or by their duty cycle (Pulse Width Modulation, PWM). One of the simplest forms of modulation can be found in nature, where neurons "fire" with a series of positive voltage pulses. The absence of the pulses can be viewed as not firing, but once the pulses do start firing, an increase in signal is denoted not by a higher amplitude, but a higher frequency of the positive pulses [1], [2]. This is sketched in Figure 1. Clearly, the lack of pulses denotes a baseline for the received signal, but the level above a baseline must be obtained through some finite length averaging of the pulse train, which can be seen on the right side drawings, symbolizing integration of their counterparts to their left. The higher the pulse frequency, the higher the baseline of the integrated result.

It is worth noting that averaging only works when the signal is single sided and that the speed of obtaining a usable output is limited by the length of the average taken.

Outside of nature, pulse-position-modulation (PPM), pulse-frequency-modulation (PFM), and pulse-width-modulation (PWM) use essentially the same ideas. While PFM is the closest analog to neurons, PWM is perhaps the most ubiquitous, showing up in many forms of low level, slow control. While it shows up in communication systems [13] as NRZ data, for the purpose of control, it is usually used as a method to encode a control input or output signal into a binary, [0,1] signal.

Pulse Position Modulation (PPM) is usually a matter of whether a pulse is there or not. The meaning of the signals is – in large part – related to whether the output signal is meant to be a baseband (low frequency) value or whether it is a driving a clock or oscillator signal. In the former case – and if the pulses are all single signed (e.g. all positive) then a low pass filter (LPF) will average the signals and give a usable low frequency output. If the thing being driven is a clock or oscillator, then the leading edges can go through a phase-detector (along with the clock leading edges) to generate a phase error (Section VI). Pulse Frequency Modulation (PFE) is similar to PPM, but the value of the modulation is encoded in the density of the pulses. When driving a clock, there is usually a minimum pulse rate to allow the clock (a phase tracking loop based on a PLL) to maintain synchronization.

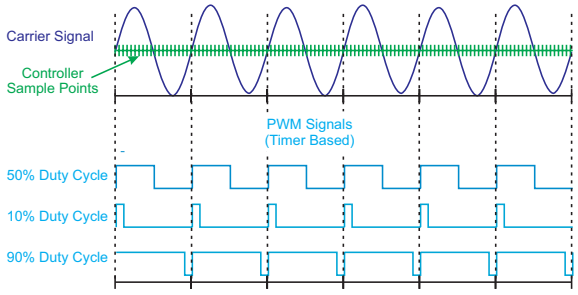

Fig. 2. Classic PWM. The numerical input modulates the duty cycle of the pulses with respect to a fixed carrier frequency.

Pulse Width Modulation (PWM) (Figure 2) is commonly used in control systems as a cheap substitute for a digital-to-analog converter (DAC). The control signal that has values ranging from 0 to 1 is modulated into the width of a pulse stream with 0 being represented by a 0% duty cycle and 1 being represented by a 100% duty cycle. PWM allows the modulated signal to be transmitted as binary data on a digital signal line, greatly increasing the immunity to low level random noise. The digital signal – being received at the plant – is simply low pass filtered (LPF) to produce an averaged output that once again spans the [0,1] range. This signal can be converted into a voltage and amplified to drive an actuator.

Pulse Width Modulation (PWM) need not have a nonzero offset, although this enables the most trivial form of demodulation: averaging. For our purposes, averaging and low pass filtering are used interchangeably in this context. PWM is a critical method to understand in part because of its ubiquity in controlling slow processes, such as voltage references, pressure, and temperature. In these contexts, the control law is most likely a PID or PI controller producing an "analog" (multi-bit) value. These values could be sent to the plant via a Digital-to-Analog Converter (DAC), and transmitted via an analog signal line, but this is more expensive than the PWM function and the analog signal is more susceptible to noise than a binary signal that is pulse-width-modulated. The PWM signal, transmitted on a binary signal line, can have small amounts of noise cleaned up by simply adding a relay centered at the voltage equivalent of 1/2 to estimate if the received signal was at logical 0 or logical 1. Once inside the receiving device, an average of the signal recovers the original multibit value (between 0 and 1). This can then be appropriately scaled to drive whichever device is being controlled.

PWM relies on the assumption that the actual signal value changes far more slowly than the modulation. This

means that a slowly changing "analog" value can be encoded into the PWM and the averaging of the 0-1 binary input will produce a value on the [0,1] range that represents the original multi-bit signal. While demodulating the signal can be accomplished simply with LPF/averaging, modulating the slowly changing signal into PWM involves producing a binary signal that is changing far faster than the "multibit" signal. This would require a microprocessor to spit out signals at a far faster rate than the control loop signaling. For example, a processor sampling analog data at say 0.5 Hz might need to modulate this signal two orders of magnitude higher, say at 100 Hz, so that the averaged/LPF signal was relatively smooth when looked at with a 0.5 Hz sample rate.

For this reason, many microcontroller chips include a bit of PWM logic that accomplishes the modulation without the processor needing to "bit bang" – a term used to describe when a processor spends a significant fraction of its processing time merely flipping bits at a high rate. The function can also be accomplished using programmable logic (PL) such as an FPGA (Field Programmable Gate Array).

III. Basic Modulation of Sine Waves

Another family of modulations used in control systems can be explained as variants on modulating a sinusoidal carrier. Consider the carrier signal, $c(t)$,

$$c(t) = A_0 \sin(\omega_0 t + \theta_0), \tag{1}$$

where A_0 is the fixed amplitude of the carrier, $\omega_0 = 2\pi f_0$ is the carrier frequency, and θ_0 is the carrier phase. Often θ_0 is defined as 0 for simplicity. For **amplitude modulation (AM)**, we modify the carrier, $c(t)$, by changing the amplitude [14] i.e.

$$s(t) = (A_m(t) \cdot A_0) \sin(\omega_0 t + \theta_0). \tag{2}$$

Phase modulation (PM) [15] is shown in

$$s(t) = A_0 \sin(\omega_0 t + (\theta_0 + \theta_m(t))), \tag{3}$$

while **frequency modulation (FM)** [16] is shown in

$$s(t) = A_0 \sin((\omega_0 + \omega_m(t))t + \theta_0), \tag{4}$$

The development of sophisticated electronics has allowed for more complex modulation schemes, such ad the amplitude-phase modulation schemes (e.g. QAM) that enabled faster modems in the days before high speed Internet connections [17].

IV. Non-Coherent AM Demodulation

It is worth making a distinction here between AM and PM/FM signals. AM signals, like the common use of Pulse Modulated Signals, can be demodulated – if one is not too precise about values or timing – using non-coherent methods. That is, a carrier or clock need not be used. There is a cost to using non-coherent methods (no clock synchronization), but the simplicity of the circuits often make up for it. Phase and frequency modulation require the use of a precise mixing signal (a signal normally at the carrier frequency) which requires more complex electronics.

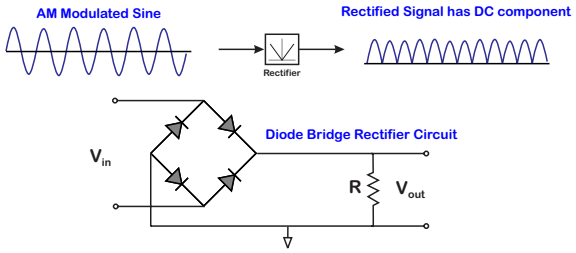

Fig. 3. Non-coherent demodulation of an AM signal with a rectifier made from a passive diode bridge. The amount of ripple in the output can be limited by adding a low pass filter on the output. It can be something as simple as adding a capacitor across V_{out}.

Sticking with non-coherent methods, we see one of the most common forms in the drawings of Figure 3. On top we see a diode bridge circuit which is one way to build a full wave rectifier using passive circuits. The output of the bridge is tied to a resistor so that if the input is an amplitude modulated (AM) sine wave (upper left), the output (upper right) is rectified. That rectified signal can now be averaged using a low pass filter (LPF) to show extract the amplitude modulation. There is an inherent assumption that the modulation is at frequencies far lower than the carrier frequency, f_0. The averaged signal will exhibit different levels of ripple (signal at $2f_0$ getting past the LPF) depending upon the LPF itself, but generally we see that the averaged signal can return the modulated amplitude. One implementation of this is known as an analog RMS-to-DC circuit[18]. In this reference the example of a 36 ms settling time of the AD736 [18] is 3,168 periods of the 88 kHz signal used in [6]. A 36 ms settling time sets the Nyquist frequency at 0.5*(1/36e-3) = 13.89 Hz. From this a reasonable closed-loop bandwidth limit would be 1/10 of the Nyquist frequency or about 1.4 Hz, which severely limits achievable bandwidth from a fairly high speed signal.

For many low speed, low to moderate precision control problems, this type of signal demodulation is sufficient. The disadvantages of such a simple scheme are that it allows through broadband noise, nonlinearities, and harmonics of the carrier frequency. Adding more LPF to minimize these effects will also lower the effective bandwidth of the demodulated signal.

V. Basic IQ Demodulation: Lock-In Amplifiers

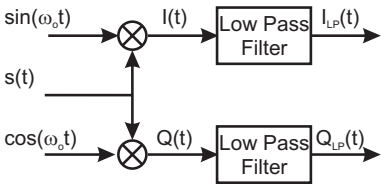

Fig. 4. Operation of a lock-in amplifier.

The next important demodulation component to understand is the in-phase/quadrature (IQ) demodulator. These are commonly used in communication systems and in precision

instrumentation. Among the simplest of these instruments to understand is the lock-in amplifier (Figure 4). Lock-in-amplifiers (LIAs) and coherent demodulation have typically been used in a variety of communication and measurement systems [19], [20], [21], [22], [23]. The difference is that those systems did not close the feedback loop and thus were not affected by latency. Furthermore, many LIAs require post-integration low pass filtering to minimize the effects of harmonics [24], [25].

Lock-in amplifiers are common measurement instruments which produce an in-phase and quadrature signal to mix with the incoming modulated sinusoid. In Figure 4, the input signal is mixed with a sine and a cosine at frequency $\omega_0 = 2\pi f_0$. By convention one signal, $I(t)$, is called the in-phase signal and the other, $Q(t)$, is called quadrature. The mixing signals will cause any component of $s(t)$ at ω_0 to produce a signal at baseband (no carrier) and one at $2\omega_0$. The low pass filters are supposed to remove this 2X frequency harmonic, as well as anything else at high frequency. Generally any signal not at ω_0 should average out in the low pass filters.

The integrated in-phase ($I(t)$) and quadrature ($Q(t)$) branches now have signals from which the magnitude and phase can be extracted via a standard rectangular to polar coordinate transformation. That is, if

$$s(t) = A\sin(\omega_0 t + \theta), \text{ then} \tag{5}$$

$$A = \sqrt{I(t)^2 + Q(t)^2} \text{ and } \theta = \arctan\frac{I(t)}{Q(t)}. \tag{6}$$

Again, there are inherent assumptions about A and θ having frequency contents far below ω_0. Lock-in amplifiers are very useful laboratory instruments for examining signals when delay doesn't matter much, but when we chose to use IQ demodulation in feedback loops, we are far more concerned about delay. A second issue is doing the computations of Equation 6. Square root and arctangent functions, are relatively expensive for real-time computations. Even the famous CORDIC algorithms developed in the 1950s for a US Air Force computer and used in the original HP-35 pocket calculator [26], [27] typically took 1 computation cycle per bit of accuracy, meaning that 32 bits of accuracy took up to a whopping 32 computer clocks. While this seems relatively small by current standards, it is not when the computation is being pushed by extremely high sample rates, as found in nano-mechatronics. For such systems, both the number of cycles and the variability of the number of cycles needed for any particular computation, are a particular hindrance to high bandwidths. We will see later, that there are ways of mitigating this. First, we need to introduce one more foundational piece, the phase-locked loop (PLL).

VI. Basic Coherent Demodulation: Phase-Lock Methods

One last critical component in this modulation/demodulation universe is that of phase-lock methods. The most basic component is the phase-locked loop (PLL)

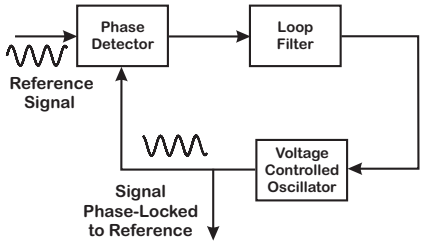

Fig. 5. A general PLL block diagram. Each PLL has a phase detector, an oscillator, a loop filter, and operates in feedback.

(Figure 5), which will be described below, but these methods are far more general and thus deserve their own mental subset. This author has argued that with PLLs in every computer, smart watch, television, radio, and generally any piece of digital electronics, the PLL is the most ubiquitous feedback loop designed by humans [13]. PLLs are unique amongst most feedback systems in that they include two intentionally inserted nonlinearities: the voltage or numerically controlled oscillator (VCO/NCO) and the phase detector. The former generates an oscillator frequency in response to an input voltage level or number and the latter extracts that phase and/or frequency from combining a pair of oscillating signals [28], [29], [30], [31], [32], [33].

In its most mathematically pure form, a PLL involves the same mixing (multiplying of two sinusoids) described in Section V on IQ demodulators. The difference here is that the latter are open-loop devices while PLLs are feedback mechanisms. While the non-coherent demodulation methods described in Section V are only useful for extracting the amplitude of a modulated signal. A PLL – by aligning the internal oscillator with the fundamental oscillation of the incoming signal – allows the phase (and sometimes frequency) of the incoming signal to be determined. This then opens up a world of other demodulation methods [34], [35], [36], [37].

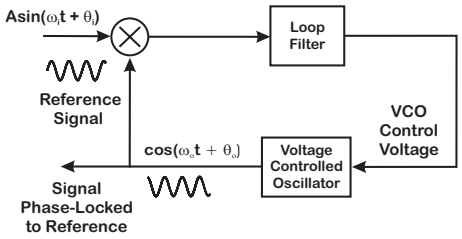

Fig. 6. A classical mixing (analog) phase-locked loop.

First introduced in the 1930s [38], a classical analog phase-locked loop (Figure 6) takes a reference sinusoid and mixes (multiplies) another sinusoid that is conceptually in quadrature (90^o out of phase) with it. Via the use of trigonometric identities and low pass filtering (LPF), the loop creates an error signal in the baseband which is a sector 1-3 nonlinearity. This allows the loop to be closed and the difference signal – corresponding to the phase – to be driven to 0.

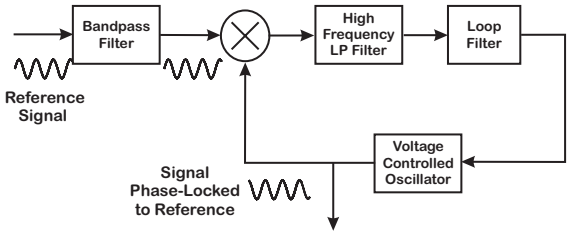

Fig. 7. A practical version of the classic mixing phase-locked loop: note the addition of a bandpass filter preceding the loop to limit input noise and a high frequency low pass filter within the loop to attenuate the 2X frequency component with minimal impact on the loop dynamics.

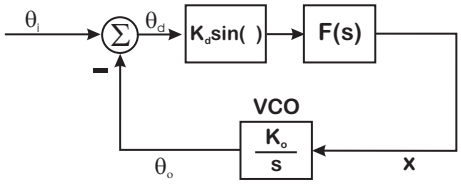

Fig. 8. Conceptual block diagram of PLL with sine detector. This is a transition stage in the analysis of the classical mixing loop. This model represents the effect of the multiplying detector once the high frequency component has been attenuated.

Because the low frequency (baseband) behavior of the multiplied sinusoids is a sine – a quadrant 1 and 3 nonlinearity (Figure 9), the phase detector output signal can be used as an error signal to drive the frequency and phase differences between the reference sinusoid and the internal sinusoid to some constant value or 0 (depending on the system type). The internal sinusoid then represents a filtered or smoothed version of the reference sinusoid.

For multibit analog input signals, one group of digital PLLs (DPLLs) approximates the analog loop in the same sense way that digital controller approximate analog ones: the modulated carrier signals are digitized using an ADC, acted upon by a set of digital filters, fed into a numerically controlled oscillator (NCO), and that signal can be converted back to analog if needed using a DAC. However, this limits the carrier signals to only those that can be effectively digitized by an ADC. PLL circuit designers, being a clever bunch, have found ways to modify these methods so as not to require a full ADC conversion. For these binary digital signals, Walsh functions replace sinusoids. Special phase detectors work on edges of clock signals, or even on simply [0,1] bits coming in from a communication link. In these cases, the analysis moves from the comfortably analytical methods that use trigonometric identities on sinusoids to almost purely heuristic methods based on an intuitive understanding of the phase detector behavior both in its baseband signals (the demodulated ones) and in the residual high frequency signals. Although uncomfortable for those of us used to having analytical descriptions, these circuits are ubiquitous in low power digital electronics and therefore should not be ignored.

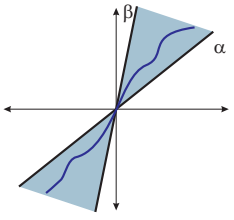

Fig. 9. A quadrant 1-3 sector nonlinearity assumed for almost all PLL phase-detectors.

Typical block diagrams of PLLs in the literature resemble Figure 6, however practical loops often more closely resemble Figure 7, in which a high frequency low pass filter is used to attenuate the double frequency term and a bandpass filter is used to limit the bandwidth of input signals to the loop. A general sinusoidal signal at the reference input of a PLL as shown in Figure 7 can be written as:

$$v_i = R_1(t) = A\sin(\omega_i t + \theta_i). \tag{7}$$

Without loss of generality, we can assume that the output signal from the Voltage Controlled Oscillator (VCO) into the mixer is given by

$$v_o = VCO_{out}(t) = \cos(\omega_o t + \theta_o). \tag{8}$$

The output of the mixer in Figure 7 is then given by

$$v_d = Mixer_{out}(t) = AK_m \sin(\omega_i t + \theta_i)\cos(\omega_o t + \theta_o), \tag{9}$$

where K_m is the gain of the mixer.

Typically, analysis of such a PLL is done by taking several simplifying steps. Using the familiar trigonometric identity in terms of the PLL:

$$2\sin(\omega_i t + \theta_i)\cos(\omega_o t + \theta_o) = \sin((\omega_i + \omega_o)t + \theta_i + \theta_o) + \sin((\omega_i - \omega_o)t + \theta_i - \theta_o) \tag{10}$$

and then making two fundamental assumptions leads to the commonly used model of the analog PLL. Let $\theta_d = \theta_i - \theta_o$. Then these assumptions are:

1) The first term in (10) is attenuated by the high frequency low pass filter in Figure 7 and by the low pass nature of the PLL itself.
2) $\omega_i \approx \omega_o$, so that the difference can be incorporated into θ_d. This means that the VCO can be modeled as an integrator.

Making these assumptions leads to the PLL model shown in Figure 8. The problem is that this is still a nonlinear system, and as such is in general difficult to analyze. The typical methods of analysis include:

1) Linearization: For θ_d small and slowly varying

$$\sin\theta_d \approx \theta_d,\ \cos\theta_d \approx 1,\ \text{and}\ \dot{\theta_d}^2 \approx 0.$$

While this is useful for studying loops that are near lock, it does not help for analyzing the loop when θ_d is large.
2) Phase plane portraits [28], [30]. This method is a classical graphical method of analyzing the behavior of low order nonlinear systems about a singular point. The disadvantage to this is that

phase plane portraits can only completely describe first and second order systems. The saving grace here is that by far the vast majority of PLLs are first or second order.

3) Simulation. Note that explicit simulation of the entire PLL is relatively rare. Because the problem is stiff, it is more typical to simulate the response of the components (phase detector, filter, VCO) in signal space and then simulate the entire loop only in phase space.

The linearized model is used for most analysis and measurements of PLLs. We can still analyze the the sinusoidal phase detector model shown in Figure 8, it has been possible to apply the technique of Lyapunov Redesign [39] to phase-locked loops [36], [13]. This can even be applied when the phase detector is digital, but the rest of the loop is analog, known as a classical digital phase-locked loop [37].

Changing the phase detector and VCO can result in a system for which this model is very accurate. It is possible to learn quite a bit about the phase behavior of the PLL from linear analysis.

Again the difference with an I-Q demodulator is the latter is an open-loop device – never minimizing the phase difference between input and output signals – while a PLL closes the loop on the phase difference so as to drive it towards 0. In contrast to an I-Q demodulator, in a PLL, the magnitude of the demodulated signal is trivially available from the remaining baseband signal. The phase error can be viewed as a residual instantaneous phase difference between the input and the locked oscillator signal. In an IQ demodulator, the magnitude and phase are typically extracted via a rectangular to polar coordinate transformation as shown in Equation 6.

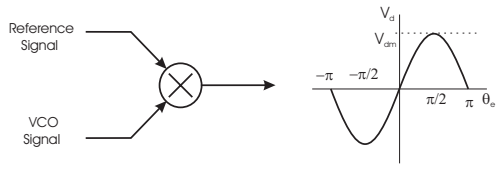

Fig. 10. Classical mixing phase detector

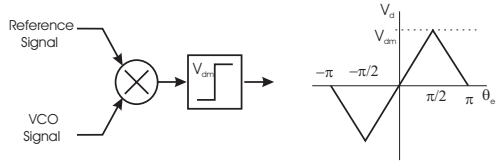

Fig. 11. Over driven mixing phase detector

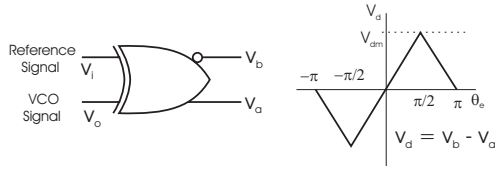

Fig. 12. Phase detection using an XOR gate. Note that this accomplishes the same thing as an over driven mixer, but with digital circuitry.

A classical mixing (multiplying) phase detector is shown in Figure 10. Once the $2X$ frequency component has been

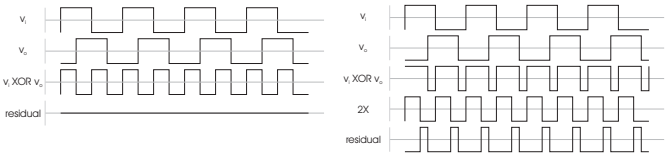

Fig. 13. Phase detection using a XOR gate. On the left, a phase shift between reference and VCO output of $\pi/2$ produces an output of the phase detector whose baseband component is 0. On the right a relative phase shift of $\pi/4$ results in an output of the phase detector whose baseband component is $v_d/2$. The output is broken up here into a 2X frequency signal and a residual. The 2X signal averages to 0, while the residual averages to the baseband phase error.

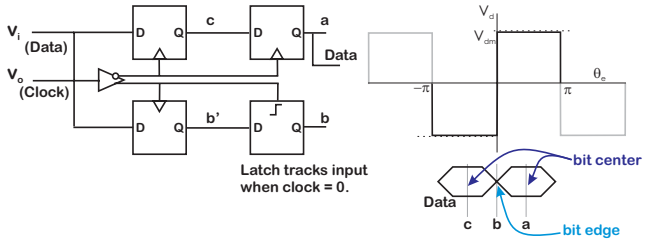

Fig. 14. Block diagram for a Bang-Bang phase detector used in clock-data recovery PLLs.

integrated out with the loop filter (and any high frequency low pass filter), the resulting phase characteristic is the sinusoidal one that we discussed earlier.

If one overdrives the circuit so that it saturates, then we get the phase response that is shown in Figure 11. Understanding the output of such a phase detector relies on a combination of averaging analysis and heuristics. However, one of the more interesting features of such a phase detector is that it can be implemented using an Exclusive-OR (XOR) gate as shown in Figure 12. One advantage of such a phase detector is that the loop gain is now independent of input signal amplitude. Furthermore, an XOR phase detector's response can have a larger linear range than a sinusoidal detector (mixer). The disadvantage is that the linearity of the baseband response is affected by the relative duty cycles of the input and VCO signals [31], [32]. The standard analysis done by PLL engineers involves drawing out square waves as shown in Figure 13 and then doing some heuristic "analysis" to convince themselves that the baseband (low frequency) component of the signal behaves with the triangular phase response shown in the right of Figure 12 (for a 50% duty cycle of the input signal).

Even more sophisticated digital circuits, such as a phase-frequency detector can integrate pulses to lock in not only the phase, but the frequency of an incoming signal. The analysis for such loops is often heuristic and graphical, but since most PLLs model the oscillator – either a voltage controlled oscillator (VCO) for analog loops or a numerically controlled oscillator (NCO) for digital loops – as an integrator and the loop filter often contains another integrator, the analysis often follows that of control of an integrator plant under PI control.

If all modulated signals were simply sinusoids, demodulated with analog circuits, the field of phase-

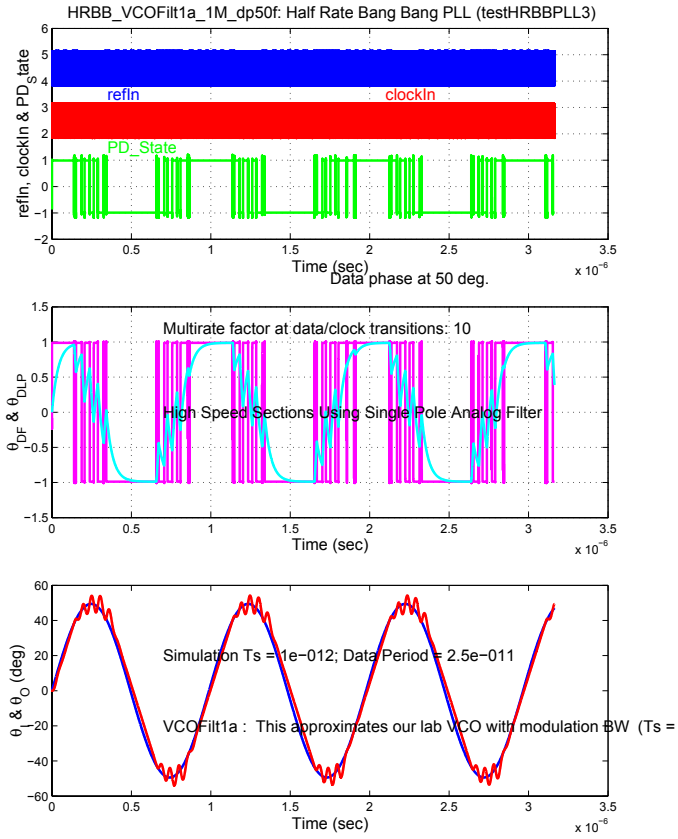

Fig. 15. Time domain response of Half Rate Bang Bang PLL simulation. Vertical fuzziness seen in phase detector output due to filters used by Matlab's decimation feature. The refIn signal is the data input. The clockIn is the recovered VCO clock. The PD State is the state of the phase detector. This is passed through two different low pass filters. The LP PD Out is low passed with a 4 GHz bandwidth. The VLP PD Out is the phase detector output passed through a 400 MHz bandwidth filter. In the bottom plot, the input phase is θ_i and the recovered clock phase is θ_o.

locked methods would be a lot duller. Because accurate sinusoids are often difficult to produce in simple circuits and difficult to maintain across a circuit, deviations using non-sinusoidal shapes have arisen, and phase detectors beyond mixers (multipliers) have been employed. These circuits are even more nonlinear than the classical PLL, but they have clear advantages in simplicity and large scale reliability. A perfect example involves replacing the sinusoids with square waves and the mixer phase detector with an Exclusive-OR circuit. The signals are binary, but the baseband behavior of the output of the XOR follows that first and third quadrant nonlinearity [13].

The point of this discussion is not to teach a lot of phase detector circuits (see [13] for that), but to point out that phase-lock methods – and the ideas behind them – can be used in all manner of signals that do not match our typical control system signals. Understanding that even these pulse trains and/or binary signals can yield information allowing us to synchronize an oscillator (or some mixing signals for an IQ demodulator) allow us to extract clean signals from all manner of encodings. As strange a device as the so-called Bang-Bang phase detector of Figure 14, can produce results that, when averaged to reveal the demodulated low frequency behavior, yield understandable phase relationships as shown in Figure 15 [40], [41].

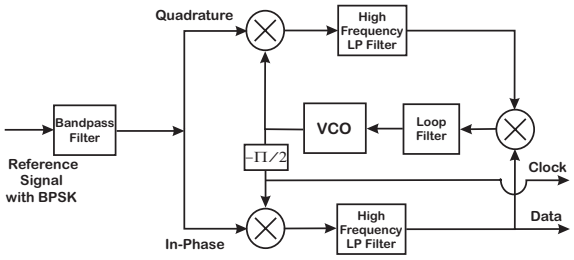

Fig. 16. A Costas loop combines a PLL and an IQ demodulation function.

It is worth understanding the Costas Loop, shown in Figure 16 because its progeny show up repeatedly in designs. We will see an example of one in the Laser Interferometers described in Section X. The upper branch of the loop, labeled as quadrature, acts like a PLL, driving the phase between the input and the mixing signal to 0. When the phase error in the upper branch is driven to near 0, the oscillation in the lower branch is in phase with the input signal. In the absence of modulation, the two in-phase signals multiplied together essentially form a $\sin^2(\cdot)$ quantity which would produce an always positive value, especially when integrated via any low pass filter. Any amplitude modulation may be extracted almost trivially at this point. While the Costas Loop is most commonly associated with communication signals using Binary Phase Shift Keying (BPSK) this basic idea can be extremely useful in speeding up precision IQ demodulation (Section VII) or in laser interferometers (Section X).

Phase-lock methods allow us to synchronize an internal oscillator with some external reference signal. In doing so we use feedback to get phase alignment between signals, which simplifies a lot of other signal extraction. In some cases, we will see that the alignment provided by phase-lock methods allows a trivial extraction of other signal information, greatly reducing the computational load.

VII. Precision Integration Lock-In

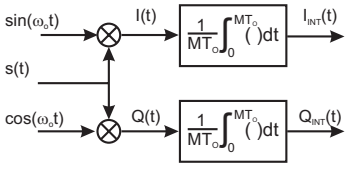

Fig. 17. Lock-in amplifier with precision integration over an integer number of periods.

The lock-in amplifier methods described in Section V had the disadvantage – from a controls perspective – of having a fairly long delay, because their use models were not affected by delay. However, as we will see in the examples that follow

this section, many of the uses of demodulation in control systems require a minimization of that delay.

One of the sources of delay is the long time constant in the LPF needed to minimize the effects of the higher harmonics produced in the mixing operation. In a pure circuit, this would only be the $2\omega_0$ harmonic, but as many circuits have small nonlinearities in practice, they had produce a wide set of higher harmonics that are hard to predict in advance. Returning to Figure 4, we can replace the generic LPF with a precision integral of Figure 17.

Mixing with in-phase and quadrature signals as shown in Figure 17 yields

$$\begin{aligned}\int I(t)dt &= \int s(t)\sin(\omega_0 t)dt \\ &\approx \int A\sin(\omega_0 t+\theta)\sin(\omega_0 t)dt \\ &\quad + \int n(t)\sin(\omega_0 t)dt\end{aligned} \tag{11}$$

and

$$\begin{aligned}\int Q(t)dt &= \int s(t)\cos(\omega_0 t)dt \\ &\approx \int A\cos(\omega_0 t+\theta)\cos(\omega_0 t)dt \\ &\quad + \int n(t)\cos(\omega_0 t)dt.\end{aligned} \tag{12}$$

Here $n(t)$ is the noise in $s(t)$. Coherent demodulation (a.k.a. lock-in amplification) is based on the idea that if one sets the mixing signal to the same fundamental period as the drive signal, $T_0 = \frac{1}{f_0} = \frac{2\pi}{\omega_0}$, and integrates, then most of the terms drop out, leaving only a signals at baseband and at $2f_0$. The higher frequency signal can be removed with a post-integration notch or low-pass filter. Often, this low pass filter effect is achieved just by integrating over a large number of periods. With analog circuits, the difficulty in precisely knowing the fundamental frequency, f_0, means that it is difficult to place an analog notch at the exact $2f_0$ frequency (or those of any other harmonics). For this reason, the use of analog Lock-In-Amplifiers for coherent demodulation is usually accompanied by a broad low pass filter. The negative phase effects of using such a filter limits the closed-loop bandwidth of any system using such a demodulator.

Ideally, we will want to integrate over an integer, M, number of periods of the frequency that we wish to demodulate. Making the integrals definite and using well known trigonometric identities, yields:

$$\begin{aligned}\tfrac{1}{MT_0}\textstyle\int_0^{MT_0} I(t)dt &= \tfrac{A}{2}\Big(\cos\theta\tfrac{1}{MT_0}\textstyle\int_0^{MT_0} dt \\ &\quad -\tfrac{1}{MT_0}\textstyle\int_0^{MT_0}\cos(2\omega_0 t+\theta)dt \\ &\quad +\tfrac{1}{MT_0}\textstyle\int_0^{MT_0} n(t)\sin(\omega_0 t+\theta)dt\Big)\end{aligned} \tag{13}$$

and

$$\begin{aligned}\tfrac{1}{MT_0}\textstyle\int_0^{MT_0} Q(t)dt &= \tfrac{A}{2}\Big(\sin\theta\tfrac{1}{MT_0}\textstyle\int_0^{MT_0} dt \\ &\quad -\tfrac{1}{MT_0}\textstyle\int_0^{MT_0}\sin(2\omega_0 t+\theta)dt \\ &\quad +\tfrac{1}{MT_0}\textstyle\int_0^{MT_0} n(t)\cos(\omega_0 t+\theta)dt\Big).\end{aligned} \tag{14}$$

Equations 13 and 14 both have the properties that the second term on the right hand side goes to 0 for all positive M. The third term goes to 0 for increasing MT_0 as long as $n(t)$ is uncorrelated with the mixing sinusoids.

Such precise control of the integration period is difficult in an analog circuit but straightforward in a digital operation. As MT_0 gets large the contribution of $n(t)$ goes to 0, yielding the familiar relationships

$$I_{int} = \frac{1}{MT_0}\int_0^{MT_0} I(t)dt \approx \frac{A}{2}\cos(\theta) \tag{15}$$

and

$$Q_{int} = \frac{1}{MT_0}\int_0^{MT_0} Q(t)dt \approx \frac{A}{2}\sin(\theta). \tag{16}$$

There are several issues with standard methods of demodulation. The first is that imperfections in the integration approximation and noise in the signal require that MT_0 be large, relative to the period of the frequency at which demodulation is to take place, T_0, so M must be large.

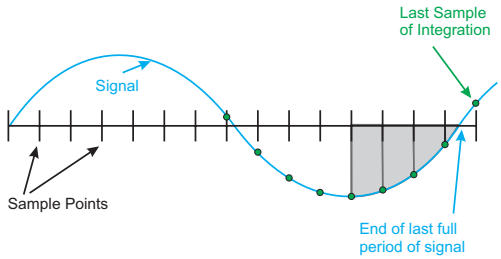

Fig. 18. Integrating the partial sample of a sampled sinusoid.

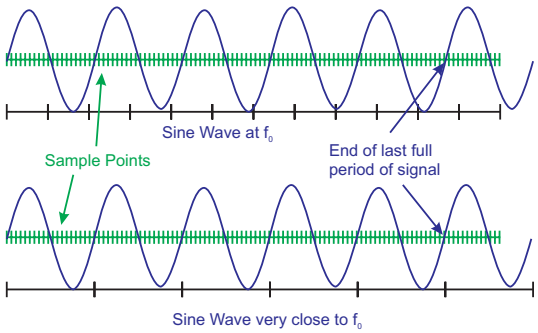

Fig. 19. The top drawing shows a sine wave which doesn't end up on an integer number of sample points. Adjusting f_0 slightly allows an integer number of periods to line up with an integer number of samples, as shown in the bottom drawing.

The second is that with a digital controller, we have to be careful if we want to honor our desire to integrate over an integer number of periods of oscillation. We want

$$NT_S = MT_0, \tag{17}$$

where N are are the number of samples in the integration, T_S is the sample period, M is the number of periods of oscillation, and T_0 is the period of oscillation. As illustrated in Figure 19 the data sample rate is rarely an integral multiple of the oscillation frequency, so it is difficult to make Equation 17 hold. Most digital systems are run at a fixed sample rate, $f_S = \frac{1}{T_S}$. The oscillation frequency, $f_0 = \frac{1}{T_0}$, comes from the frequencies at which we want to measure the FRF. That means f_0 will vary but f_S will not. A solution is that for any desired f_0 and M, we can pick N such that:

$$NT_S \le MT_0 = N_{Real}T_S \le (N+1)T_S. \tag{18}$$

We then round N_{Real} to the nearest integer. We don't want to change T_S or M, so we have two options. There is a fundamental difference between these methods when we have a fixed fundamental frequency, f_0 and when we can adjust it. In the most common case when equality does not hold, the last period of the integration is a partial one, as shown in Figure 18. This will require $N+1$ samples where the first N samples of the integral integrate over the complete sample period and the last one is interpolated over a partial sample. If we are trying to precisely match a frequency, such as the resonant frequency of an AFM cantilever [6], [7], then we round N_{Real} down to the closest integer below and then integrate over a partial fraction of an interval (Figure 18). The length of the partial fraction of an interval changes with every sample period and oscillation frequency, but would be fixed during any one measurement.

One example of a practical implementation of these integrals was described in the author's earlier work on a low latency, high fidelity demodulation for atomic force microscopes (AFMs) [6], [7].

In those papers, a trapezoidal rule integration is used. This seems to provide a reasonable compromise between minimizing latency and integration accuracy. Consider the trapezoidal rule implementation of our integral:

$$\int_{t_0}^{t_N} y(t)dt \approx \sum_{k=0}^{N-1} \left(\frac{y_{k+1} + y_k}{2} \right) T_S, \tag{19}$$

where T_S and N are defined as in Equation 18. Between N and $N+1$, we will have a partial interval integral that must be computed

$$\int_{t_N}^{t_k+T_S h} y(t)dt \approx \left(\frac{y_{N+1} + y_N}{2} \right) hT_S, \tag{20}$$

where $0 \leq h \leq 1$ and

$$h = \frac{MT_0 - NT_S}{T_S}. \tag{21}$$

Note that hT_S is the integration time needed to complete the M^{th} period of oscillations at f_0, so the fraction of a sample period that this represents is given by h. Putting these together and looking back in time rather than forward, we get

$$\int_{kT_S - MT_0}^{kT_S} y(t)dt \approx S_k \text{ where} \tag{22}$$

$$\frac{S_k}{T_S} = \sum_{j=0}^{N-1} \left(\frac{y_{k-j} + y_{k-(j+1)}}{2} \right) + \left(\frac{y_{k-N} + y_{k-(N+1)}}{2} \right) h. \tag{23}$$

$$\frac{S_k}{T_S} = \frac{y_k}{2} + \sum_{j=0}^{N-1} y_{k-j} + \frac{y_{k-N}}{2} + h \left(\frac{y_{k-N} + y_{k-(N+1)}}{2} \right). \tag{24}$$

Equation 24 is very instructive because it shows us that the integral can be simply constructed as a FIR filter. We can factor out a single scale factor, T_S, and then we have a main integral corresponding to the terms before the term scaled by h and the fractional portion, scaled by h. It is also instructive that very little about this formula is dependent upon the sample interval, f_S, and the oscillation frequency, f_0. Basically, a change in f_0, f_S, and/or the number of periods in the integral, M, changes only N and h. For a given M, T_0, and T_S, we pick N from Equation 18 and h from Equation 21.

Eventually, after a lot of algebra, we got to the remarkably simple incremental computation for the integral:

$$S_k = \frac{T_S}{2} \left[I_k + h \left(y_{k+1} + y_k \right) \right]. \tag{25}$$

where

$$I_k = \Delta I_k + I_{k-1}, and \tag{26}$$

$$\Delta I_{k+1} = y_{k+1} + y_k - (y_{k+1-N} + y_{k-N}). \tag{27}$$

This relationship was relatively straightforward to program into an FPGA, with the fractional portion of the integral removed from the iteration. Thus, the additions and subtractions from the incremental sum in Equation 27 are exact, preventing the possibility of small errors in h accumulating in the recursion. Some will recognize that this form is essentially the same form as a Cascaded Integrator-Comb (CIC) Filter [42].

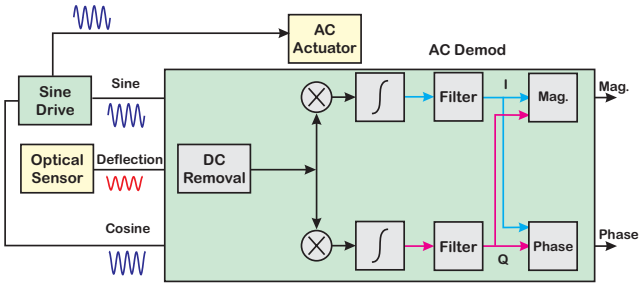

Fig. 20. Coherent demodulation for AFM. DC removal and post integration filtering included.

The partial interval integral is necessitated in some cases by needing to precisely match a predetermined oscillation frequency. This is often due to a physical parameter – such as the cantilever resonance in the AFM demodulator. Note that if the exact frequency match is not critical, as with a built in sine-dwell [23], then we can adjust T_0 so that $h = 0$ in Equation 21. This simplifies the sum terms above.

$$\tilde{N}T_S = M\tilde{T}_0. \tag{28}$$

In that case, M was generally assumed to be at least 8, as implemented in the HP 3562A Dynamic Signal Analyzer [20]. Furthermore, the measurement frequencies in that use were servo system frequencies, generally significantly lower than the frequency of a cantilever tip oscillation. Finally, the generation of an accurate frequency response function measurement did not hinge on maintaining a single frequency. This allowed the set of oscillation frequencies to be adjusted slightly, so that for each measurement frequency, $f_0 = \frac{1}{T_0}$, equality in Equation 28 held.

The adjustments to T_0 to make equality hold in Equation 28 can be kept small if N and M are made large. While this approach may be feasible for an off-line measurement described above, or for producing signal processing results that are not used in the feedback loop, this choice will add to latency in the integral calculation, so we are better off integrating over the fractional interval as described above.

Digital quadrature is documented in many numerical computation texts [43], [44]. Generally, the algorithms for quadrature will make use of a polynomial fit over some number of sample points to approximate the function. The fit of a L^{th} order polynomial will involve $L+1$ points. In applications where latency (time delay) is not an issue, one can achieve higher accuracy by conducting the integral between samples k and $k+1$ using samples on either side of this interval.

For example, the Hewlett-Packard 3562A computes the integral of its mixed sinusoids by using a fifth order polynomial fit over 6 points [20]. It uses 3 points on either side of the interval in question. As the interval of integration slides forward in time, points to the left and right of it are used to give a more accurate approximation of the function being integrated. Note that the interval over which the integral is done is delayed by two and a half sample intervals (compared with simply using only the latest sample point).

For use in generating the error signal for a feedback controller, we want to minimize the latency of the integral and for this the simplest discrete integral approximations are the forward and backward rectangular rule approximations, and the trapezoidal rule approximation. Because of their small amount of delay, these are often used in generating discrete equivalents of analog controllers [45]. The forward rectangular rule has a single period delay. The backwards rectangular rule has zero delay. Finally, the trapezoidal rule has a half sample period delay.

Standard practice in digital lock-in amplifiers is to use one of the rectangular rule approximations and rely on integrating over many periods of oscillation to drive the error to 0. However, by using a higher-order approximation and a partial sample integral, we can cut the error down with significantly fewer periods of integration.

For the amplitude modulation of the AFM cantilever, we are mostly concerned about the magnitude of the signal. In the case of a frequency-response function (FRF), we need both the magnitude and phase. However, while the integrals must be computed in real-time to keep up with the signals, these latter quantities can be computed off line on the results of the integrals, as they comprise a finite set of frequency results.

The other major difference between the precision IQ demodulation used in real-time feedback and that used in say a stepped-sine calculation is in extracting the magnitude and phase from the output of the precision integrators. In Figure 20, the standard rectangular to polar computations are done.

To extract the magnitude in real-time for a feedback calculation requires the computation of Equation 6. The difficulty

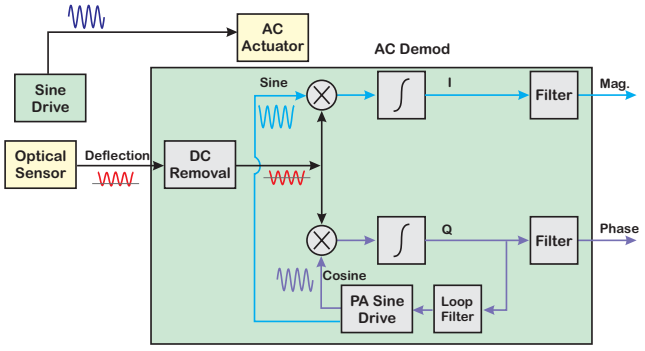

Fig. 21. Coherent demodulation for AFM using a PLL. DC removal and post integration filtering included.

comes in the resources needed to compute these relationships in real time with high sample rates. For example, a highly efficient algorithm is the so called CORDIC algorithms [26], [27]. This algorithm computes magnitude and phase by rotating the frame of reference until the frame of reference and the signal have a matching magnitude and phase. The CORDIC algorithm is computationally simple, and is at the heart of the trigonometric calculations in the original HP-35 calculator [27]. However, to compute magnitude and phase, a CORDIC algorithm requires one computational cycle per bit of accuracy, so a 16-bit accuracy would require an extra computational delay (on top of that done by the integral itself) of 16 clock cycles. In a standard computer, this might be considered fast, but in a DSP or FPGA which typically complete table lookup operations, additions, and multiplies in one or two cycles, this is considered slow.

A more time efficient calculation involves a table lookup. However with two different numbers to look up and the high precision desired in these calculations, the tables can become huge. A 16-bit quantity would nominally require $2^{16} = 64\text{K}$ values. A few well placed adjustments to and restrictions of the calculation make it possible to use a relatively small table to give reasonably good estimates. A fuller discussion is found in [7].

A a faster way is to use the knowledge of phase-lock techniques to simplify the calculation. The quadrature branch (Q) of the integral is very close to a PLL, if we allow the mixing oscillator to have its phase adjusted in response to the phase error in that branch. With the mixing signal phase-locked, the in-phase (I) branch is aligned so that the output of that integral is proportional to the cosine of the phase difference. As that phase difference is driven to the vicinity of 0, the cosine is approximately 1, and the magnitude drops out trivially.

Some examples of the demodulator from Figure 21 in action are shown in Figures 22 and 23. We see in Figure 22, the original oscillation signal on top, with the I and Q branch integrals below. Because of the PLL structure of the lower, Q branch, we see that the Q integral converges to 0 as the loop locks, leaving the I branch locked to the magnitude. In Figure 23, we examine the behavior of post integration filtering on the result, and we can see (especially in the lower

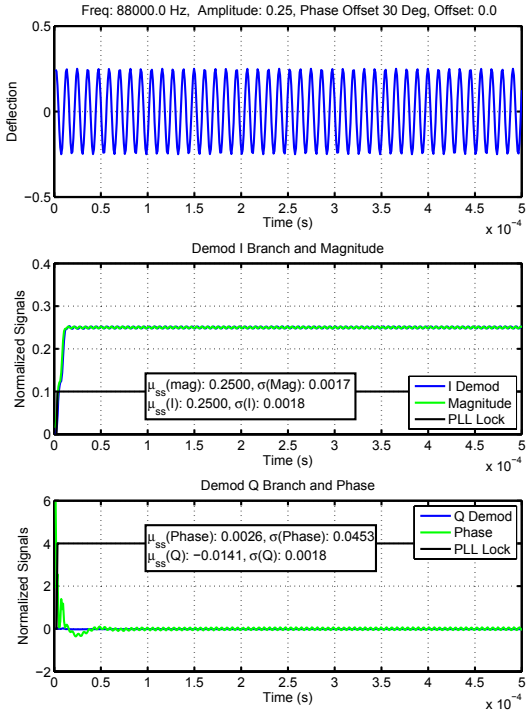

Fig. 22. Output of ModelSim Simulation of FPGA based demodulator. The oscillation frequency is 88 kHz. The normalized deflection amplitude is 0.25. There is no offset in the signal level, but the phase of the signal driving the deflection is 30° ahead of the in-phase (sine) mixing signal at the beginning of the simulation. Note how the I and Q phases converge to the magnitude and instantaneous phase, respectively.

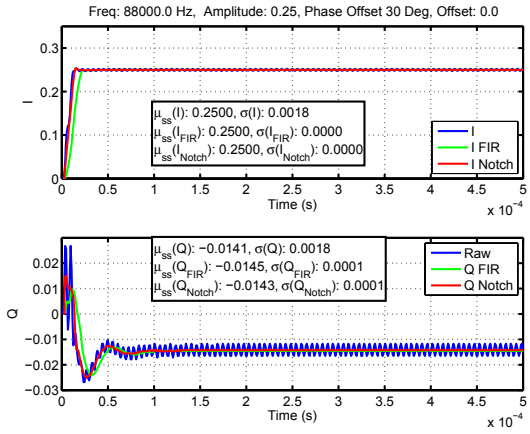

Fig. 23. Output of ModelSim Simulation of FPGA based demodulator. The oscillation frequency is 88 kHz. The normalized deflection amplitude is 0.25. There is no offset in the signal level, but the phase of the signal driving the deflection is 30° ahead of the in-phase (sine) mixing signal at the beginning of the simulation. This plot shows the effect of adding post post integration filtering to the demodulator.

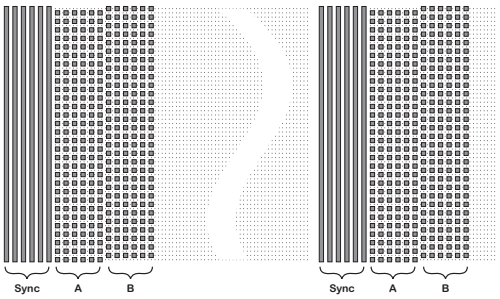

Fig. 24. The layout of an HDD track. The sync field allows the PLL to recover its timing so that the fields can be detected at the right time. In this simplified form, there are just A and B fields offset by half a track from the track center. After the A and B fields comes the user data, which typically comprises a much larger proportion of the track.

Q plot) that a little bit of filtering removes a residual $2f_0$ frequency left over from imperfect integration.

VIII. Example: Servo Signal Demodulation in Hard Disk Drives

Hard disk drives (HDD) provide a breadth of examples of modulation and demodulation schemes used to encode position. We can trace an evolution of demodulation methodologies from looking at different methods we might apply to the same basic signals. Modern disk drives use what is called sectored servo, which means that the position information is multiplexed with user data in periodic samples along the track known as servo bursts. The servo burst itself will consist of several fields: a clock sync field, servo position information, and an edit gap (Figure 24). The clock sync field is merely a pattern of alternating magnetic polarities along the track. These fields are consistent across the cross-track direction. This allows the PLL to minimize any phase drift that may have occurred during the data portion of the track. After that, there are magnetic patterns offset from each other on opposite sides of the track center. A highly simplified view of this for our discussion is shown in Figure 25. The top drawing shows the relative position of the read/write head relative to the offset position information. The goal is to track the center of the track and if the head is too far over one field (A or B), then the return signal is larger for that portion, as shown in the lower drawing.

With the clock recovered, the servo processing portion of the chip opens up windows for the A field and the B field, separately. The output signals from the electronics are illustrated in the lower drawing. We can see that when the readback head is off to one side of track center (in this illustration more towards A), that the corresponding readback will produce a higher amplitude in the signals illustrated in the lower drawing. Now, assuming that the clocking allows us to separate the A "signal" from the B "signal", we have some options on how to extract their relative amplitude.

Figure 26 illustrates different methods for extracting the amplitude of each field during its active window. The first two methods, are the oldest used in HDD, and while they are only active in the open clock window for the particular field, are not coherent themselves. They both use a rectifier to make the two-sided, zero-mean signal a one-sided signal. The first (line (c)) uses **peak detection**, employing a circuit that simply holds the highest value found in the window. While this method largely ignores low level baseline noise, it is highly susceptible to any noise that manifests itself at the top of the peak. The next (line (d)) improves on

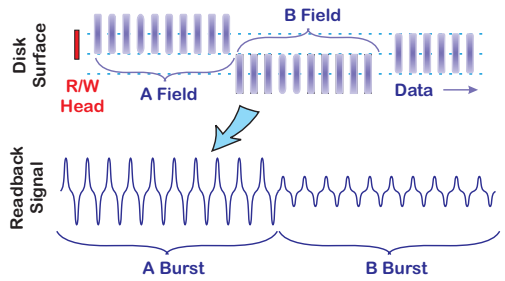

Fig. 25. Disk drive position fields and the resulting signals.

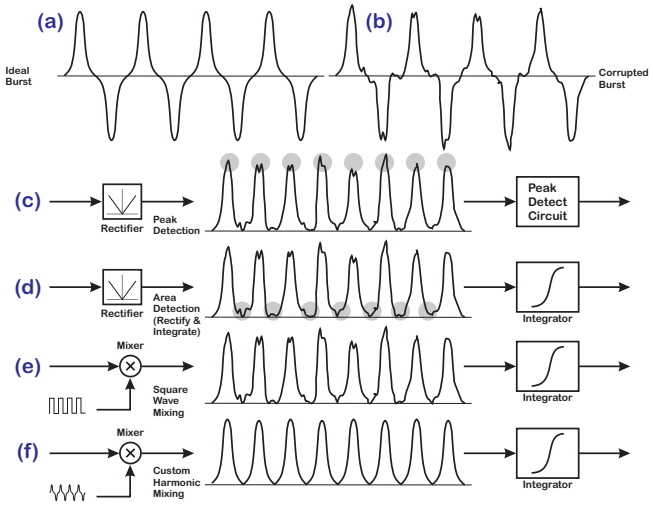

Fig. 26. Common demodulation methods used in decoding disk drive position bursts.

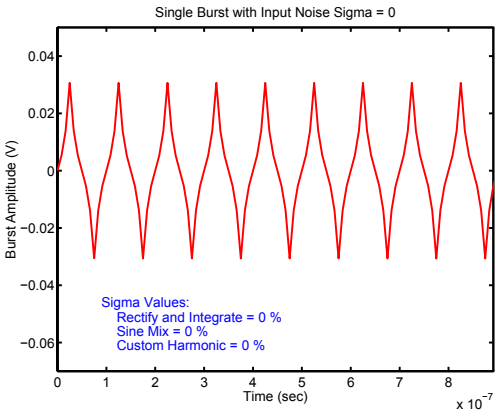

Fig. 27. Ideal servo burst with no noise. Note that beyond the fundamental, the signal shape contains third and fifth harmonics.

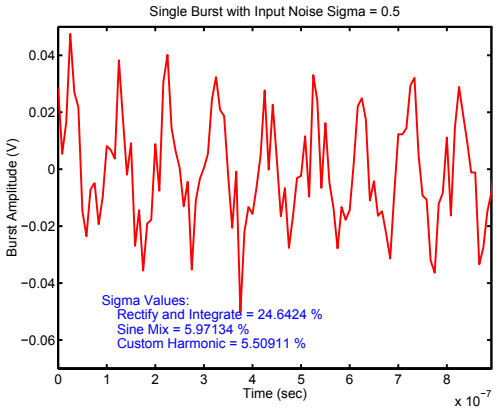

Fig. 28. deal servo burst with significant noise. Note that beyond the fundamental, the signal shape contains third and fifth harmonics. We see how coherent demodulation adds significant immunity to noise in the servo position signal.

the noise immunity by integrating the signal over the time window. The idea of this **rectify and integrate** method is that the effect of random noise is reduced compared to that of the signal. However, since the signal was rectified, noise that was previously zero mean now has a DC bias rather than being averaged out. Rectify and integrate was generally considered an improvement on peak detection, but required a more complex circuit.

We can remove the rectification of noise by multiplying the signal in the window with a square wave aligned with the pulses. We see (in line (e)) that this rectifies the servo signal while the noise can still average towards its zero mean level. Besides requiring a coherent square wave, the main downside of this is that the square wave mixing admits all harmonics of returned servo burst signal. Finally (in line (f)) we see that if we are selective about which harmonics we use in the mixing signal, we can remove some of the nonlinear distortion while capturing the best features of the servo signal [46], [47], [48]. The ideal signal shapes might be sinusoidal, but the noise free actual signals resemble those of Figure 27 which includes first, third, and fifth harmonics. The addition of significant noise reveals the tremendous difference in the performance of the different schemes (Figure 28). The effect on the noise admitted into the system can be seen in Figure 29, where moving from rectify and integrate down to mixing with the fundamental sine wave dramatically drops the noise passed into the system. Adding extra harmonics (the custom harmonic curve) tweaks this result to be a little bit better.

IX. Example: Optical Disk Precision Clocking: DVD+RW

On optical drives such as CDs and DVDs, there are two main servo loops, one to maintain focus of the optical beam and the second to keep that beam over the correct track. However, because the tracks are distinguished by physical features on the disk, the tracking loop has the potential for much higher sample rates than are found in the HDD loops described in Section VIII. In these problems the more difficult problem is that of establishing sub-bit accurate timing down the track for making rewritable optical disks that are compatible with the ubiquitous DVD ROM format. The latter has none of the edit gaps mentioned in the discussion of HDDs in Section VIII, and so being able to write new data in the correct down-the-track location requires precise synchronization with sub-bit accuracy.

The essential technology is a high frequency, high fidelity reference signal embedded into the disk surface itself, as

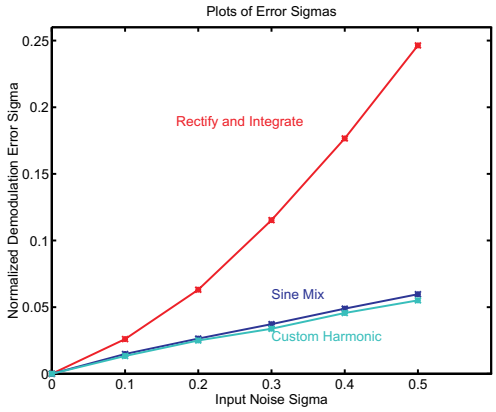

Fig. 29. Advanced demodulation can reduce the noise of the signal, and therefore minimize the noise circulating in the feedback loop. Rectify and integrate admits far more noise than either demodulating with the first harmonic (sine wave) or multiple matched harmonics.

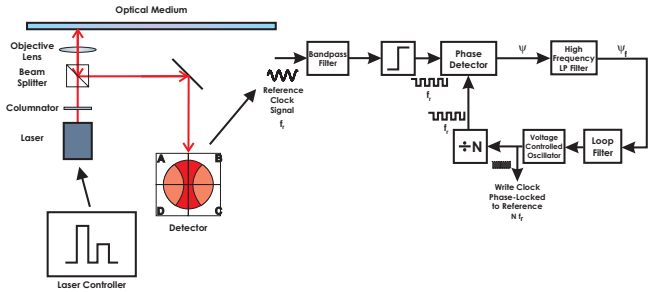

Fig. 30. Rewritable optical disk drive system with a Harmonic Locking PLL to generate the write clock [49].

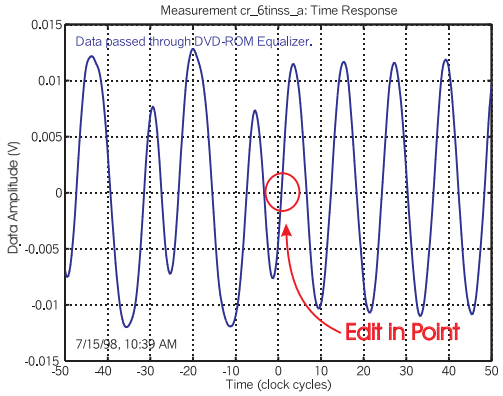

Fig. 32. DVD+RW, gapless edit. A 6T pattern spliced into a 4T-8T pattern. This represents the time response at the edit-in point.

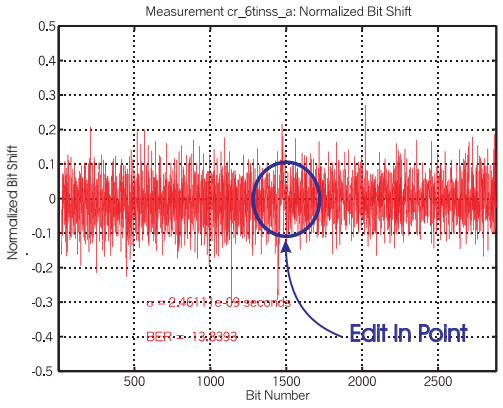

Fig. 33. DVD+RW, gapless edit. A 6T pattern spliced into a 4T-8T pattern. This represents the phase error for a data clock generated from the data. (Note the absence of any phase jumps.)

illustrated in Figure 31. With this so-called high frequency wobble, a digital phase-locked loop can correct the timing to enable read/write operations without edit gaps. A gapless edit of a 6T pattern into a 4T-8T pattern is shown in Figures 32–34. The lack of jumps in phase errors in a clock derived from the data (Figure 33) as well as the lack of any 5T or 7T patterns in the histograms of Figure 34 indicate a bit perfect edit, where T is the bit clock period. The continuous nature of the high fidelity reference signal simplified the PLL design which enabled the DVD+RW (and DVD+R) formats [49], [50], [51], [52]. By having a continuous, high fidelity reference clock signal across the entire length of the track, the internal timing of the read

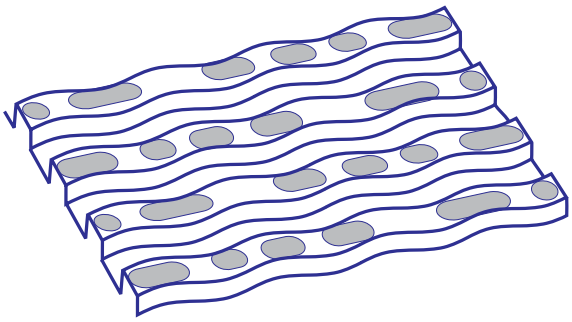

Fig. 31. High frequency wobbles used in the DVD+RW optical disk format.

and write circuits was within fractions of a bit (allowing the desired read/write functionality with no edit gaps. By the year 2013 the DVD+RW and DVD+R businesses enabled by this amounted to 25% of a $559M market (according to a website called Storage.com which now has been re-purposed for finding self-storage units).

X. Example: Laser Interferometry

This section is largely excerpted from the author's *A Tutorial on Laser Interferometry for Precision Measurements* [53]. The full equation derivations are in that reference, but a few key points will be used here to describe laser interferometers in the context of demodulation systems. The operation of an interferometer depends upon optics performing some of the same equations that we have been discussing above. It is in understanding the relationship of the optics equations that we see that the optics are performing a demodulation of two signals at the same frequency interfering with each other and therefore producing a phase difference that we can detect with something akin to a Costas loop.

The basic Michelson interferometer (Figure 35) uses a half silvered mirror to split a monochromatic light source into two beams. Each beam reflects off of a mirror, to be recombined at the half-silvered mirror. The recombined beam contains an interference pattern that changes when either of the mirrors

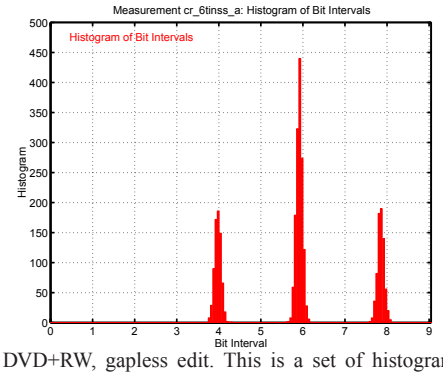

Fig. 34. DVD+RW, gapless edit. This is a set of histograms of the bit intervals. The absence of any 5T or 7T bits is an indication that no bit errors have occurred.

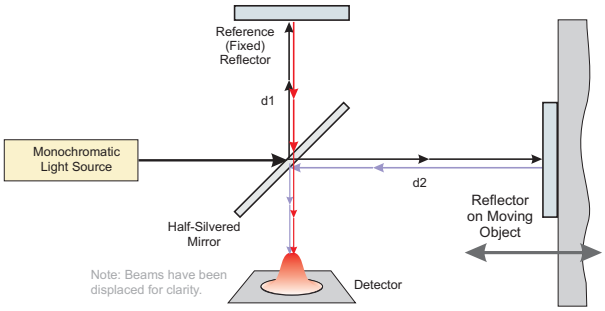

Fig. 35. Basic Michelson Interferometer

move. Keeping one mirror fixed allows one to attribute all of the interference pattern changes to motion of the other mirror.

Many texts show the an interference pattern such as the one in the far left of Figure 36. However, in the absence of the beam being cropped, the detector will see a collimated beam on its center axis. Typically, this is modeled as a Gaussian beam and as the measure mirror moves the intensity of this pulse will vary, as shown in the right three figures of Figure 36. The detector then acts to integrate the intensity of the beam over its spatial extent, and – assuming the integration is faster than the change in the interference pattern – this integrated intensity can be used to measure distance, modulo the wavelength of the laser used.

The Michelson interferometer is one of the most basic models of interferometry available. It is not a practical interferometer, in that there are significant issues with the actual implementation. However, it provides an easy to understand conceptual model for understanding precision measurement

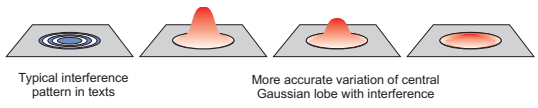

Fig. 36. Effects of interference on detector. On the far left is the typical diagram one sees in books. However, the banding is typically caused by the beam being cropped and not the effect of the interferometry. A better picture comes from the three diagrams on the right, in which the intensity of the central Gaussian lobe is modulated by the interference pattern.

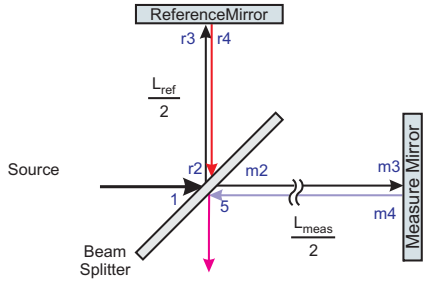

Fig. 37. Some details on the beams of a Michelson interferometer

interferometers. Generally, for every imperfection of the Michelson interferometer, there is a practical fix that expands the range of usefulness of the interferometer [53]. Each of these fixes essentially returns the interferometer back to a more ideal Michelson behavior.

We start our analysis of the Michelson IF equations by looking at Figure 37. For our purposes, the source beam can be considered to originate at position 1, right before contact with the half silvered mirror. At the mirror, half of the beam is reflected to the reference mirror (path r2-r3-r4) where it is reflected back towards the half silvered mirror. At this interface, half of the beam is passed through to position 5, while half reflects back to the source. Meanwhile, the transmitted portion of the beam goes to the measurement mirror (path m2-m3-m4) and reflects back. At the half silvered mirror, half of the measure beam is reflected to position 5, while half passes back to the source. We are concerned with the two beams that meet at position 5 and are imaged on the detector.

A few things are important to understand interference as it is used in our measurements. First, since both the reference beam and the measure beam originate from the same laser, they are coherent with each other. Second, every time a beam goes through a reflection, it undergoes a $180°$ phase shift. A look at the diagram of Figure 37 indicates that each beam at position 5 has gone through $360°$ in phase shifts and thus they are still in phase with each other. Third, by the time both beams reach position 5, their amplitude has been reduced to $\frac{1}{4}$ of their original amplitude. (This is fixed in practical interferometers via use of polarizing beam splitters, quarter wave plates, corner cubes, and a second frequency so that a far greater percentage of the beam power hits the detector [53].) Conceptually, if the reflection/transmission is exactly 50/50 and if the mirrors are perfectly aligned, then both beams add through linear superposition and have the same amplitude. Thus, we can attribute the variation at the detector to interference.

The equations for the interference pattern are derived in classic optical texts [54], [55] from application of the vector electromagnetic wave equations [56], [57]. Consider the electric field of the source beam at position 1:

$$E_{z,source}(z,t) = A\cos(kz - \omega t + \phi) \tag{29}$$

where z is the direction of travel, $k = \frac{2\pi}{\lambda}$ is the wave number, λ is the wavelength of the light, and A is the amplitude of the

beam. From position 1, the reference beam travels a distance $L_{ref} = 2d_1$ to get back to position 5, while the measure beam travels a distance $L_{meas} = 2d_2$ to get back to position 5. If we consider position 1 to be $z = 0$, then the two beams are thus,

$$E_{ref}(t) = \frac{A}{4}\cos(kL_{ref} - \omega t + \phi) \tag{30}$$

for the reference beam and

$$E_{meas}(t) = \frac{A}{4}\cos(kL_{meas} - \omega t + \phi) \tag{31}$$

for the measure beam. Through linear superposition, the beams add, so that the electric field of the combined beams at position 5 is

$$\begin{aligned} E_{tot}(t) &= \tfrac{A}{4}\left[\cos(kL_{ref} - \omega t + \phi)\right. \\ &\quad \left. + \cos(kL_{meas} - \omega t + \phi)\right]. \end{aligned} \tag{32}$$

A word about notation is useful here. Normally, when one is working with wave equations [56], [57], the equations are set up as vector quantities along some frame of reference. This works very well in analysis of plane equations, point sources, etc. but in an interferometer, the direction of the beams are switched so often that keeping track of all the vector frames becomes confusing. For this tutorial, we will assume that the source electric field is in the X-Y plane, and the source magnetic field is rotated $90°$ in that plane. This means that the Poynting vector which describes the energy density is in the Z direction. Every time we go through a reflection, polarizer, or beam splitter, the reference frame is changed, but our signals will end up so that the Poynting vector is normal to the detector plane. For the sake of simplicity, we will leave off the unit vector designations on the equations.

The detector is sensitive to signal intensity, not amplitude, and we can calculate this from the Poynting vector. If we assume that the electric field is in the x direction and the magnetic field is in the y direction, then

$$\begin{aligned} H_{tot}(t) &= \tfrac{A}{4}\sqrt{\tfrac{\epsilon}{\mu}}\left[\cos(kL_{ref} - \omega t + \phi)\right. \\ &\quad \left. + \cos(kL_{meas} - \omega t + \phi)\right]. \end{aligned} \tag{33}$$

We now have two choices to simplify this: proceed with trigonometric identities or switch gears to saying that Equations 32 and 33 are the real parts of a complex exponential notation. For pedagogical purposes, we will plug through the trigonometric equations here. With the polarizations we have assumed, the Poynting vector, $\mathcal{P}(t)$, will be in the direction normal to the detector with

$$\begin{aligned} \mathcal{P}_{tot}(t) &= \vec{E}_{tot}(t) \times \vec{H}_{tot}(t) \\ &= \frac{A^2}{16}\sqrt{\frac{\epsilon}{\mu}}\left[\cos\alpha + \cos\beta\right]^2 \qquad (34) \\ &= \frac{A^2}{4}\sqrt{\frac{\epsilon}{\mu}}\left[\cos^2\alpha + \cos^2\beta + 2\cos\alpha\cos\beta\right] \qquad (35) \end{aligned}$$

where $\alpha = kL_{ref} - \omega t + \phi$ and $\beta = kL_{meas} - \omega t + \phi$. With this and some trigonometric identities, we end up with

$$\cos^2\alpha = \frac{1 + \cos 2(kL_{ref} - \omega t + \phi)}{2}, \tag{36}$$

$$\cos^2\beta = \frac{1 + \cos 2(kL_{meas} - \omega t + \phi)}{2}, \text{ and} \tag{37}$$

$$\begin{aligned} 2\cos\alpha\cos\beta &= \cos\left(k(L_{meas} + L_{ref}) - 2\omega t + 2\phi\right) \\ &\quad \times \cos\left(k(L_{meas} - L_{ref})\right). \end{aligned} \tag{38}$$

Putting these all together,

$$\begin{aligned} \mathcal{P}_{tot}(t) &= \frac{A^2}{32}\sqrt{\frac{\epsilon}{\mu}}\left[1 + \cos 2(kL_{ref} - \omega t + \phi)\right. \\ &\quad +1 + \cos 2(kL_{meas} - \omega t + \phi) \\ &\quad +2\cos\left(k(L_{meas} + L_{ref}) - 2\omega t + 2\phi\right) \\ &\quad \left. + 2\cos\left(k(L_{meas} - L_{ref})\right)\right] \end{aligned} \tag{39}$$

If we average over an integer number of periods, $T = \frac{1}{f} = \frac{2\pi}{\omega}$ then the time varying portion integrates out, leaving only the DC portion:

$$\mathcal{P}_{tot,avg} = \tfrac{A^2}{16}\sqrt{\tfrac{\epsilon}{\mu}}\left[1 + \cos\left(k(L_{meas} - L_{ref})\right)\right] \tag{40}$$

This rationale should remind the reader of the high precision IQ demodulator of Section VII. As a practical matter, the laser frequency is so much faster than the integration time of our detector that we are always getting the "DC portion". Thus, the relationship that is most commonly used for this type of interferometer is that for the intensity:

$$I \sim K\left[1 + \cos\left(k(L_{meas} - L_{ref})\right)\right] \; W/m^2 \tag{41}$$

This is often rewritten in terms of the wavelength, λ, as

$$I \sim K\left[1 + \cos\left(\tfrac{2\pi}{\lambda}\left(L_{meas} - L_{ref}\right)\right)\right] \; W/m^2 \tag{42}$$

This provides the power density at the detector in $Watts/m^2$. The detector integrates the energy density (intensity) over the detector surface. Thus, it is not the pattern on the surface that matters so much as the amount of intensity on that surface. Equation 41 gives the density at a given point. In fact for a highly collimated beam, the distribution is likely Gaussian and effect of a change in L_{meas} is to cause the height of this Gaussian distribution to rise and fall. If all other variables are held constant, one can measure a change in distance by counting the passing of these light and dark times. From Equation 41 we see that we are still missing an ability to discern direction of motion. This is fixed in a single frequency IF by splitting the the beam and adding a phase delay to one portion, thus allowing for in-phase and quadrature demodulation (IQ), again reminiscent of Section VII.

We start with the reminder that all position measurements with an interferometer are relative. The fringes give a change in position from some starting position. This is analogous to trying to measure position from velocity measurements: one must assume a starting position.

IF measurements rely on knowing the wavelength of light, λ, and the wavelength being stable. This is why commercial interferometers did not emerge until lasers were invented. It is important to know that as pressure, temperature, humidity,and gas composition change, so does λ. Thus, an IF system is making measurements with a somewhat elastic ruler.

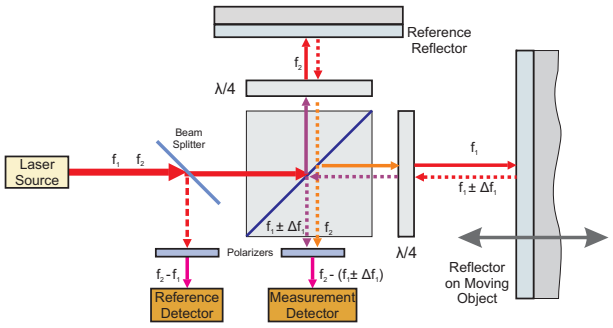

Fig. 38. Two frequency (heterodyne) Michelson interferometer configuration.

All the modifications to the Michelson interferometer discussed thus far essentially are designed to desensitize the interferometer to non-ideal behavior and restore the accuracy of Equation 42. However, even when things are properly aligned, the interferometers described so far operate in the baseband. They use a single frequency of light, also known as homodyne interferometry, and the "difference" between measure and reference only shows up as a baseband phase and Equation 42 is a variation away from DC. DC detection is slow and suffers from $\frac{1}{f}$ and other noise in the detectors, mainly signal intensity variations (due to air turbulence or accumulated contaminants on mirror and optic surfaces) being indistinguishable from position changes.

Borrowing from the world of radio communications, it is more advantageous if the interference shows up at some intermediate frequency. To achieve this, modern IF measurement systems typically operate with multiple wavelengths [58], where the interference pattern is not a baseband signal, but in fact an AC signal, as diagrammed in Figure 38. Thus, distance becomes a measurement of the difference between two signals, one of which (known as the measurement signal) is modulated by the moving object, while the other (known as the reference signal) is generally fixed. The reference signal is usually composed of the difference between the two frequencies before one of them has been modulated, but can also be another modulated signal to create a differential measurement between two moving mirrors.

Assuming the two laser frequencies are ω_1 and ω_2, we get the equation for the Poynting vector [53]. If the measurement mirror is moving, then that movement will appear as a Doppler Shift in $\omega_1 = 2\pi f_1$, so that $\omega_1 \Longrightarrow \omega_1 + \Delta\omega_1$ becomes:

$$\mathcal{P}_{IF,LP}(t) \approx \frac{A^2}{2}\sqrt{\frac{\epsilon}{\mu}}\,[1+ \cos\left(k_1 L_{meas} - k_2 L_{ref2} - (\omega_1 + \Delta\omega_1 - \omega_2)t\right)]\,. \tag{43}$$

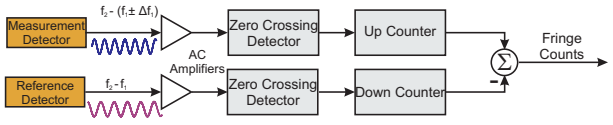

Fig. 39. Generating distance from AC frequency differences.

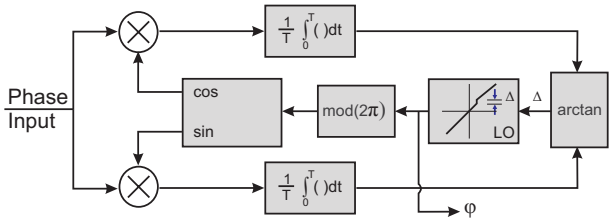

Fig. 40. Phase generation from interference pattern input. Note the structural similarity to the Costas loop of Figure 16.

In the end, the optics have generated an intensity that is proportional to a $1+cos(\theta)$ where θ is periodic and related to the distance between the reference reflector and the moving reflector. The fundamental accuracy is inversely proportional to the laser wavelength (λ).

In Equation 43, θ becomes a shift in the frequency difference between the two signals, and results in new oscillatory signal that once again needs to be demodulated. To be an instrument for measuring distance and velocity we need devices for counting either peaks in intensity (a peak finding demodulator, Figure 39) or to accurately and quickly measure the phase of the intensity signal (Figure 40). Note how Figure 40 closely matches the Costas loop of Figure 16. The peak counter (peak finding demodulator) has an accuracy limited to half the wavelength, while the IQ demodulator is capable of much finer resolution.

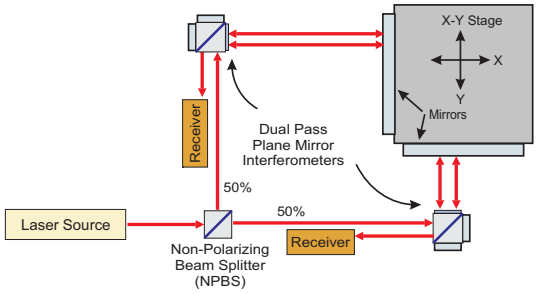

Fig. 41. Two-axis plane mirror interferometer configuration

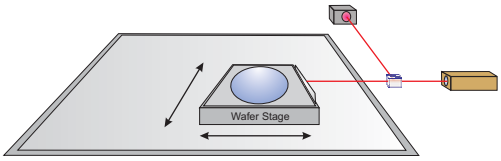

Fig. 42. Wafer stage system measured with interferometer.

One of the great benefits of precision interferometry for position measurement is that because the measurements are done at a distance, multiple axes can be measured with the same system, by splitting the laser beam and directing it off of different surfaces and back to multiple receivers. This can be seen in the two-axis configuration shown in Figure 41, where a single beam is split and directed at polarizing beam splitter based interferometers. Each of these beams is reflected off of a planar mirrors on the side of a moving stage, resulting in position measurements for the x and y axes.

The bulk of interferometers systems are used in the IC photolithography industry [59], [60]. Here, very precise machines move an X-Y stage under an optical column. What is critical is the location of the stage relative to the optical column, and the repeatability of this measurement. So these systems use laser interferometers to measure the X and Y positions of the stage and the column, as well as the pitch and yaw of these items. Some even measure the vertical direction of the stage. Figure 42 shows the basic setup without the optical column which would obscure the stage.

This section has gone through more math than most of the others combined, but it shows that at the end, to a clean position and velocity signal, we again need to understand modulation and demodulation. A huge amount of the improvement in the interferometry measurements are made by an understanding of the optical paths, but at the end, the final bit of accuracy is limited by signal conversion done in the electronics. Mixing (multiplying) and digitally integrating/low-pass filtering a signal usually requires at least 10 samples per second. If we consider the upper range of FPGA fabric circuitry to be around 500 MHz, then we can consider processing a 50 MHz signal, but only if we use only one clock cycle per sample. More reasonably, we might expect a heavily pipelined algorithm to have 10 cycles of signals up to 5 MHz. Faster signals would require either custom signal processing circuitry or analog processing for the front end.

XI. Summary

Modulated signals are rarely considered a main part of control design. It is rare that the carrier itself has significant information for the loop. It is probably for this reason that this subject is not well studied by control engineers. However, the form of modulation, and the method by which we demodulate it can have a significant effect on the quality of the demodulated signal returned to the loop.

This tutorial has given a brief, non-exhaustive overview of demodulation methods for modulated signals found in control applications. The emphasis was placed on a conceptual understanding, but with a view to how to execute the different demodulation computations. There may well be a wide variety of cases for which the simplest modulation/demodulation – such as amplitude or pulse encoding demodulated with a simple rectifier and low-pass filter are sufficient. This is really a requirements question: when the speed and accuracy of the demodulation scheme is an order of magnitude above the needs of the loop, there is no reason to do anything more sophisticated. However, in understanding the more advanced methods, we have the option to speed and clean up the sensor signals before they get into the feedback loop, thus avoiding the limitations imposed by Bode's Integral Theorem [11], [61].

The demodulation methods we see often have a lot of commonality. An understanding of Fourier integrals is extremely helpful, but to make them practical, we have to seriously look at the information content of the signals and the computational structures needed to extract them. The payoff here is the potential to dramatically lower the sensor noise (and nonlinearities) injected into a loop, as well as dramatically speeding up the acquisition of the signal. As we know from our earliest controls principles that sensor noise goes right through to the output [62], [63], the extra work has a direct payoff.

References

[1] W. H. Calvin, "Normal repetitive firing and its pathophysiology," in *Epilepsy: A Window to Brain Mechanisms*, J. Lockard and A. A. Ward, Eds. New York: Raven Press, 1980, pp. 97–121.

[2] B. Widrow, Y. Kim, and D. Park, "The Hebbian-LMS learning algorithm," *IEEE Computational Intelligence Magazine*, pp. 37–53, November 2015.

[3] Q. Zhong, D. Inniss, K. Kjoller, and V. Ellings, "Fractured polymer/silica fiber surface studied by tapping mode atomic force microscopy," *Surf. S. Lett.*, vol. 290, no. 1-2, pp. L688–L692, Jun. 1993.

[4] V. B. Elings and J. A. Gurley, "Jumping probe microscope," Digital Instruments, Santa Barbara, CA USA, United States Patent 5,266,801, November 30 1993.

[5] D. Y. Abramovitch, S. B. Andersson, L. Y. Pao, and G. Schitter, "A tutorial on the mechanisms, dynamics, and control of atomic force microscopes," in *Proceedings of the 2007 American Control Conference*, AACC. New York, NY: IEEE, July 11–13 2007, pp. 3488–3502.

[6] D. Y. Abramovitch, "Low latency demodulation for atomic force microscopes, Part I: Efficient real-time integration," in *Proceedings of the 2011 American Control Conference*, AACC. San Francisco, CA: IEEE, June 29–July 1 2011.

[7] ——, "Low latency demodulation for atomic force microscopes, Part II: Efficient calculation of magnitude and phase," in *Proceedings of the IFAC 18th World Congress*, IFAC. Milan, Italy: IFAC, August 28–September 2 2011.

[8] D. M. Harcombe, M. G. Ruppert, and A. J. Fleming, "A review of demodulation techniques for multifrequency atomic force microscopy," *Beilstein Journal of Nanotechnology*, vol. 11, pp. 76–91, January 2020.

[9] M. R. P. Ragazzon, S. Messineo, J. T. Gravdahl, D. Harcombe, and M. G. Ruppert, "Generalized lyapunov demodulator for amplitude and phase estimation by the internal model principle," in *Conference: 8th IFAC Symposium on Mechatronic Systems*, IFAC. Vienna, Austria: IFAC, September 2019.

[10] 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.

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

[12] 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*, IEEE. Montreal, Canada: IEEE, August 2020.

[13] ——, "Phase-locked loops: A control centric tutorial," in *Proceedings of the 2002 American Control Conference*, AACC. Anchorage, AK: IEEE, May 2002.

[14] Wikipedia. (2020) Amplitude modulation. [On line; accessed April 12, 2020]. [Online]. Available: https://en.wikipedia.org/wiki/Amplitude_modulation

[15] ——. (2020) Phase modulation. [On line; accessed April 12, 2020]. [Online]. Available: https://en.wikipedia.org/wiki/Phase_modulation

[16] ——. (2020) Frequency modulation. [On line; accessed April 12, 2020]. [Online]. Available: https://en.wikipedia.org/wiki/Frequency_modulation

[17] ——. (2020) Quadrature amplitude modulation. [On line; accessed April 12, 2020]. [Online]. Available: https://en.wikipedia.org/wiki/Quadrature_amplitude_modulation

[18] C. Kitchin and L. Counts, *RMS-to-DC Conversion Application Guide*, 2nd ed. Analog Devices, Inc., 1986.

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

[20] R. C. Blackham, J. A. Vasil, E. S. Atkinson, and R. W. Potter, "Measurement modes and digital demodulation for a low-frequency analyzer," *Hewlett-Packard Journal*, vol. 38, no. 1, pp. 17–25, January 1987.

[21] *HP 3563A Control Systems Analyzer*, Hewlett-Packard, 1990.

[22] Wikipedia. (2012) Lock-in amplifier. [Online; accessed December 30, 2014]. [Online]. Available: http://en.wikipedia.org/wiki/Lock-in_amplifier

[23] 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.

[24] K. S. Karvinen and S. O. R. Moheimani, "A high-bandwidth amplitude estimation technique for dynamic mode atomic force microscopy," *Review of Scientific Instruments*, vol. 85, no. 2, p. 023707, 2014.

[25] M. G. Ruppert, K. S. Karvinen, S. L. Wiggins, and S. O. R. Moheimani, "A Kalman filter for amplitude estimation in high-speed dynamic mode atomic force microscopy," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 1, pp. 276–284, January 2016.

[26] J. E. Volter, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computation*, vol. 8, pp. 330–334, 1959.

[27] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems – I:Regular Papers*, vol. 56, no. 9, pp. 1893–1907, September 2009.

[28] F. M. Gardner, *Phaselock Techniques*, 2nd ed. New York, NY: John Wiley & Sons, 1979, iSBN 0-471-04294-3.

[29] A. Blanchard, *Phase-Locked Loops*. New York, NY: John Wiley & Sons, 1976.

[30] A. J. Viterbi, *Principles of Coherent Communication*, ser. McGraw-Hill Series in Systems Science. New York, NY: McGraw-Hill, 1966.

[31] D. H. Wolaver, *Phase-Locked Loop Circuit Design*, ser. Advanced Reference Series & Biophysics and Bioengineering Series. Englewood Cliffs, New Jersey 07632: Prentice Hall, 1991.

[32] R. E. Best, *Phase-Locked Loops: Design, Simulation, and Applications*, 3rd ed. New York: McGraw-Hill, 1997.

[33] J. A. Crawford, *Frequency Synthesizer Design Handbook*. Norwood, MA 02062: Artech House, 1994.

[34] D. Y. Abramovitch, M. C. Fischer, J. N. Hogan, and C. P. Taussig, "Harmonic correction in phase-locked loops," Hewlett-Packard, Palo Alto, CA USA, United States Patent 6,646,964, November 11 2003.

[35] D. Abramovitch, "Lyapunov Redesign of analog phase-lock loops," in *Proceedings of the 1989 American Control Conference*, AACC. Pittsburg, PA: IEEE, June 1989, pp. 2684–2689.

[36] D. Y. Abramovitch, "Lyapunov Redesign of analog phase-lock loops," *The IEEE Transactions on Communication*, vol. 38, no. 12, pp. 2197–2202, December 1990.

[37] D. Abramovitch, "Lyapunov Redesign of classical digital phase-lock loops," in *Proceedings of the 2003 American Control Conference*, AACC. Denver, CO: IEEE, June 2003, pp. 2401–2406.

[38] H. De Bellescize, "La réception synchrone," *L'onde électrique*, vol. 11, pp. 225–240, May 1932.

[39] P. C. Parks, "Liapunov redesign of model reference adaptive control systems," *IEEE Trans. on Automatic Control*, vol. AC-11, no. 3, July 1966.

[40] D. Y. Abramovitch, "Efficient and flexible simulation of phase locked loops, Part I: Simulator design," in *Proceedings of the 2008 American Control Conference*, AACC. Seattle, WA: IEEE, June 11–13 2008, pp. 4672–4677.

[41] ——, "Efficient and flexible simulation of phase locked loops, Part II: Post processing and a design example," in *Proceedings of the 2008 American Control Conference*, AACC. Seattle, WA: IEEE, June 11–13 2008, pp. 4678–4683.

[42] Wikipedia. (2020) Cascaded integratorcomb filter. [On line; accessed April 19, 2020]. [Online]. Available: https://en.wikipedia.org/wiki/Cascaded_integrator-comb_filter

[43] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge: Cambridge University Press, 1988.

[44] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. London: Academic Press, 1981.

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

[46] 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.

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

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

[49] D. Abramovitch, D. Towner, C. Perlov, J. Hogan, M. Fischer, C. Wilson, I. Çokgör, and C. Taussig, "High Frequency Wobbles: A write clock generation method for rewritable DVD that enables near drop-in compatibility with DVD-ROMs," in *Technical Digest of ISOM/ODS 1999: Joint International Symposium on Optical Memory and Optical Data Storage 1999 Conference*, IEEE/LEOS,OSA,SPIE,JSAP. Koloa, HI: SPIE, July 1999, pp. 56–58.

[50] D. Y. Abramovitch, "Turning the tracking problem sideways: Servo tricks for DVD+RW clock generation," in *Proceedings of the 2000 American Control Conference*, AACC. Chicago, IL: IEEE, June 2000, pp. 2615–2620.

[51] ——, "Magnetic and optical disk control: Parallels and contrasts," Agilent Labs, Palo Alto, CA 94304, On-Line Full Draft of Paper, June 2001, http://www.labs.agilent.com/personal/-Danny_Abramovitch/pubs/hd_vs_od_servo.pdf.

[52] ——, "*References for:* Magnetic and Optical Disk Control: Parallels and Contrasts," Agilent Labs, Palo Alto, CA 94304, On-Line Bibliography of Paper, June 2001, http://www.labs.agilent.com/personal/-Danny_Abramovitch/pubs/hd_vs_od_servo_refs.pdf.

[53] 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.

[54] E. Hecht and A. Zajac, *Optics*, ser. Addison-Wesley Series in Physics. Reading, MA: Addison-Wesley, 1979.

[55] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference, and Diffraction of Light*, 6th ed. Oxford, England: Pergamon Press, 1980.

[56] S. Ramo, J. R. Whinnery, and T. V. Duzer, *Fields and Waves in Communication Electronics*, 2nd ed. New York, NY: John Wiley & Sons, 1984.

[57] J. William H. Hayt, *Engineering Electromagnetics*, 4th ed., ser. Electrical & Electronic Engineering. McGraw-Hill Inc., 1981.

[58] J. N. Dukes and G. B. Gordon, "A two-hundred-foot yardstick with graduations every microinch," *Hewlett-Packard Journal*, pp. 2–9, August 1970.

[59] H. Butler, "Position control in lithographic equipment: An enabler for current-day chip manufacturing," *IEEE Control Systems Magazine*, pp. 28–47, October 2011.

[60] R. M. Schmidt, G. Schitter, and J. V. Eijk, *The Design of High Performance Mechatronics: High-Tech Functionality by Multidisciplinary System Integration*. Delft, NL: IOS Press (Delft University Press), September 15 2011.

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

[62] K. Ogata, *Modern Control Engineering*, 3rd ed., ser. Prentice-Hall Instrumentation and Controls Series. Englewood Cliffs, New Jersey: Prentice-Hall, 1970.

[63] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed. Upper Saddle River, New Jersey: Prentice Hall, 2006.