A timing impairment module for electrical synchrometrology

D. M. Anand*, C. Freiheit,‡ M. Weiss*, K. Shenoi†, H. Ossareh‡
*National Institute of Standards and Technology
{and4, mweiss}@nist.gov
†Qulsar Inc.
kshenoi@qulsar.com
‡University of Vermont
cfreihei, hamid.ossareh@uvm.edu

***Abstract*—Accurate time is frequently cited as an enabling requirement for precisely coordinated control systems used in the electrical power system. Methods and technologies to evaluate the impact of impaired time accuracy on these control systems are frequently expensive to build and confined to a laboratory setting. Our focus in this paper is to develop a system to apply timing impairments to electrical sensors in the field. We expect that the value of such a system would be in elucidating potentially negative interactions between interconnected and interdependent measurement and control components as deployed. As such, this paper outlines the design of a new hardware-in-the-loop tool for applying timing impairments to sensors that require accurate time. In addition to the design of the tool, we discuss the challenges and one approach for implementing realistic impairment scenarios that may be comprised of stochastic variations, systematic offsets and accumulating errors.**

***Index Terms*—Timing subsystems, Testing, Reference Governor**

I. Introduction

The National Institute of Standards and Technology (NIST) designs and develops test procedures and calibration equipment for high precision measurements of the electrical power system. In recent years a focus of these development efforts has been on sensors designed for synchrometrology [1]. Synchrometrology refers to time synchronized point-on-wave (time domain) and vector (phasor domain) measurements made across multiple sites within an electrical network. By combining these tightly synchronized measurements, control systems are able to estimate relative phase differences in voltage and current waveforms at different points on the network to infer dynamics in the electrical power flows and to predict the onset of potentially unstable dynamics. The desired timing accuracy for such sensors are reported to be in the order of 10^{-5} seconds [2].

More recently, synchronized measurements of electromagnetic transients are being used as early indicators of electrical faults. Here the desired timing accuracy is determined by the propagation velocity of traveling waves induced by the transients. As computed in [3], these velocities are approximately $250\ m/\mu s$ for distribution lines, whereas the spatial resolution required to localize a fault in a distribution circuit is on the order of tens of meters [4]. This spatial resolution is an abstract requirement for a system comprised of several individual components that detect transient features, timestamp them, analyze the time of arrival at various locations and compute modal reflections. These emergent applications of synchrometrology would require synchronization accuracy well below 10^{-7} seconds.

Given the variety of algorithms, software, and hardware used for synchrometrology; sensors are often evaluated in a Hardware-In-the-Loop (HIL) fashion where models of operational scenarios are simulated and applied to networks of real sensors, controllers, and power converters. The benefit of this testing approach is in providing end users confidence in the performance of an end-to-end synchrometrology system that combines interconnected equipment from different vendors. So far, these HIL tests have been performed in a laboratory setting [1] for one class of sensors called Phasor Measurement Units (PMU). The experimental setup for these tests uses specialized calibration equipment, making it expensive to replicate and not well suited for field testing sensors.

During a workshop conducted by NIST to gain input on timing challenges in power systems, power system operators and substation engineers expressed an interest in field testing tools to test the impact of impaired timing on a more general class of synchrometrology systems [5]. Following this feedback, NIST has been working to design a Timing Impairment Module (TIM) that can:

1) Generate timing impairment scenarios as part of a hardware-in-the-loop test system.
2) Test devices already installed in the field by producing timing signals compatible with standard commercial interfaces.
3) Reproduce impairment scenarios comprised of stochastic variations, systematic offsets and accumulating errors in the time reference values.
4) Replay timing impairments recorded in the field with sufficient fidelity and repeatability.

This paper presents an overview of the proposed Timing Impairment Module (TIM) and documents results from an early prototype of the device. In Section II, we document the hardware and software design for the TIM. In Section II-A, we present our planned approach for implementing impairment scenarios including stochastic and systematic deviations from reference time. In Section III we show data experiments performed on the TIM prototype and discuss our planned approach for ensuring that the output phase from the device

978-1-5386-7607-3/19/$31.00 ©2019 IEEE

is appropriately governed using a reference shaping scheme. The proposed reference governor design is applied to a model of the TIM's phase tracking circuitry to show performance improvements in the device's ability to accurately output the impairment scenarios.

A. Effect of impaired timing

The most widely deployed synchrometrology sensor in the power system is a PMU. PMUs estimate the phase, frequency, frequency modulation and amplitude of the fundamental grid frequency (60 Hz in the U.S.) in relation to a reference cosine wave synchronized to the second counter in Coordinated Universal Time (UTC). As a result, erroneous clock offsets in the PMU clock manifest as phase error of the vector measurement. The prevailing synchrophasor standard C37.118.1-2011 [6] and its amendment bound the total vector error for synchrophasors to a threshold (typically $\pm 1\%$ under steady state conditions). This threshold effectively bounds the desired steady state clock offset to $\leq$ 26.5 μs. The impact of dynamics in the PMU clock offset is more difficult to analytically evaluate. Simulation studies shown in [7] demonstrate the impact of phase steps and phase drift on grid control functions. The simulations show a case where a drift in phase directly results in false activation of circuit protection devices and a case where phase steps degrade a wide-area damping controller to an extent where the regulation function of the controller has a negative impact on system stability.

NIST tested the impact of the leap second on PMUs. These timing tests were implemented on the NIST calibrator for conformance calibration systems. The results of these tests (see [8]) showed that the application of a leap second could manifest as phase errors if the power system frequency is not exactly nominal. The analysis of such outcomes is challenging since several algorithms may be used to estimate frequency and phase and implementations of these algorithms tend to be proprietary.

Computing the sensitivity of synchrometrology sensors to stochastic timing impairment is non-trivial for similar reasons as discussed above. For the sake of analysis, let us assume that a majority of the algorithms utilize the Discrete Fourier Transform (DFT) or a similar algorithm [9] applied to point on wave samples of a sinusoidal signal ω_0 sampled at ω_s rad/s. Then the variance in the Fourier series $F(\omega_0) = \sum f_{n\in N} = \frac{R}{N}(\omega_0) + \mathrm{j}\frac{I}{N}(\omega_0)$, where R and I are the real and imaginary parts, respectively, can be expressed in terms of first order effects introduced by phase noise in the clock generating ω_s, as illustrated in Figure 1. Root-mean squared (RMS) timing jitter introduced by the power-law phase noise is given by (see [10]):

$$T_{jitter} = \frac{\sqrt{2 \cdot 10^{A/10}}}{\omega_s} \tag{1}$$

where A is the integrated noise power over a frequency domain of interest (typically $2\omega_s$). If T_{jitter} is a Gaussian distribution over the DFT window, then the standard deviation of the measurand $U_{n\in N} = \alpha_n \cdot T_{jitter}$ where α_n is the first derivative of ω_0 at sampling instant n. With these assumptions enforced,

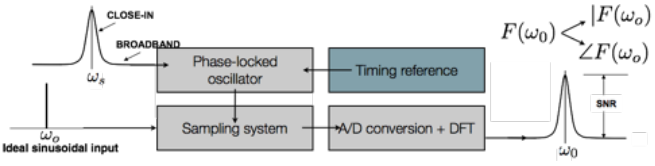

Fig. 1. A schematic diagram showing the components of a synchrometrology sensor impacted by phase noise in the sampling process.

the variance of the Fourier series as derived by [11] can be written as:

$$U^2_{R(\omega_0)}|T_{jitter} = \sum_{n=0}^{N-1} f_n \cos^2(\frac{\omega_0 n}{N}) \cdot U_n^2 \tag{2}$$

$$U^2_{I(\omega_0)}|T_{jitter} = \sum_{n=0}^{N-1} f_n \sin^2(\frac{\omega_0 n}{N}) \cdot U_n^2$$

B. System overview

Beyond the assumptions on the implementation of the DFT algorithm, the analysis presented in Section I-A assumes stationary, uncorrelated phase noise and Gaussian deviates for jitter. In practice, the algorithms used by the sensor being tested are unknown, however the design of the TIM is intended to ensure reproducible stationary, uncorrelated implementations of U_n and T_{jitter} in addition to systematic impairments to drift and steady state offset.

With this capability, the TIM can be integrated with existing commercial HIL test systems for synchrometrology sensors. The schematic diagram in Figure 2 illustrates the anticipated test setup where an existing grid emulator is used to generate validation scenarios for power system monitoring, protection and control [12], while the TIM provides timing impairments via three time transfer channels, i.e. an appropriately modulated 10 MHz sinusoid, a 1 Hz pulse train bearing the necessary jitter/wander/drift, and lastly a time code transmitted via standard digital protocols [13], [14].

TIM's software interfaces were designed to be consistent with the framework outlined by the North American Synchrophasor Initiative - PMU Application Requirements Task Force's methodology for examining data quality impacts to synchrophasor applications [15].

Specifically, their report outlines systematic experiment design for *'benchmarking'* synchrophasor systems by combinatorially testing an application with impaired datasets relative to a clean dataset(s), effectively highlighting interactions via a multi-dimensional error analysis. The report also stresses *'standardization'* of test scenarios in order to develop an application-specific performance envelope that shows the characteristics and magnitude of impairment which can be tolerated and yet deliver sufficiently accurate output (based on users or developers acceptability requirements), as well as which impairments can render the applications output inaccurate or untrustworthy. While the scope of this report extends beyond sensors, to networks and data aggregators, our

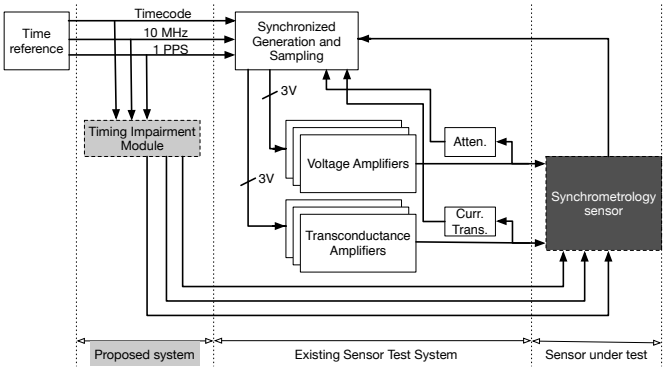

Fig. 2. An illustration showing the layout of a HIL test for synchrometrology sensors. The Timing Impairment Module can be integrated with existing HIL test components.

design presented in Section II directly addresses the need for combinatorial testing and repeatable test vectors highlighted in the report.

II. TIM Design

The TIM comprises a Numerically Controlled Oscillator (NCO) with a clock control servo tracking a reference 10 MHz source. This provides a stable and accurate frequency source that may be shared with other components in a HIL test setup. This frequency source is utilized in a phase-locked loop (PLL) arrangement that tracks the reference clock's phase. Phase bias can be inserted into the PLL tracking loop by adding an offset ($\Delta\phi$) to the phase detector of the servo as depicted in Figure 3.

The output of the NCO is processed by a clock engine that can provide multiple low jitter (low phase noise) clock outputs. For example, the clock engine generates a 1Hz output by first up-converting the 10 MHz to a suitably high frequency such as 125 MHz and then dividing this down to 1 Hz. Note that when the oscillator is locked to the reference input, the noise floor is determined by the phase-detector granularity. The granularity of the current design is 8 ns (based on a 125 MHz clock signal).

Some synchrometrology sensors only accept network based timing inputs, requiring impaired timing signals to be transmitted over a network. The TIM is based on a Qulsar M88 Managed Clock Engine [16] which includes all the functions that are necessary for a SyncE [17] or IEEE 1588v2 [18] based network clock synchronizer. Leveraging these features, the clock outputs can also be routed to the available network synchronizers allowing the TIM to function as a Master Clock for network based timing.

The core technical contribution discussed in the following sections is our work in developing specialized algorithms to accurately reproduce impairment scenarios into modulo 2π phase bias values. These algorithms address challenges in simulating stochastic impairments in finite time and address the need to automatically compensate for tracking dynamics in the PLL.

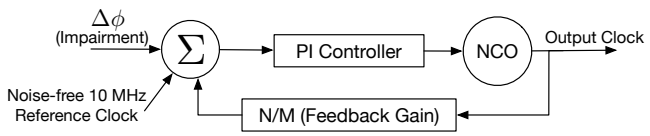

Fig. 3. A diagram of the clock control loop showing an input for phase bias inputs from an impairment simulator. The control loop is comprised of a Proportional-Integral (PI) Controller and a rational feedback gain N/M.

A. Implementation of impairment scenarios

We hope to garner feedback from the timing community for timing impairment scenarios that might reflect operational concerns. We expect that most scenarios would be comprised of stochastic series consistent with the usual power-law noise processes found in timing systems [19] such as white, flicker and random walk phase and frequency modulation. In addition, we can add systematic offsets in the form of a frequency bias, a frequency drift, bounded frequency modulation and discrete time or frequency jumps. Any combination of these can be added to provide a large variety of controlled timing impairments. These combinations of effects are compiled offline into sequences of phase offset set points ($\Delta\phi$).

The finite-time realizations of stochastic impairment functions are non-trivial and require care for a number of issues. A first concern is that the random number functions in computer systems are often not sufficiently random, but generally have sequential correlations. In order to address this issue, we follow the development in [20]. Section 7.1 in [20] discusses how to improve a system-supplied routine, and Section 7.2 discusses how to generate Gaussian Normal deviates. Gaussian deviates alone can create stochastic white phase modulations (WhPM) or white frequency modulations (WhFM). Integrating a white noise process, or in our discrete case, summing the white impulses of phase or frequency modulation produces a random walk in phase or frequency (RWPM) or (RWFM). However, producing flicker phase or frequency modulation requires a more complex filter of white noise.

Much work has been done on simulating various power-law noise processes. We include references to discussions based on the autocorrelation function using z-transform techniques [21] and [22] and a method based on spectral properties using the Fourier Transform [23]. Our method of generating stochastic noise is to use the latter method to produce a discrete-time sequence consistent with a model of a power-law noise process where the exponent of the power law, $Sx(f) = f^{-\beta}$, is arbitrary for $0 \leq \beta \leq 4$. This includes using techniques from [20] to ensure the sequences are sufficiently random.

As described above, the TIM applies changes to the phase of the output signal by accepting a discrete-time sequence of offsets ($\Delta\phi$ in Figure 3). Without compensation, the transfer function of the clock control loop in Figure 3 would modulate the random sequences generated by our power law model resulting in spurious correlations and tracking artifacts at the output of the TIM. Since we use spectral methods to generate stochastic noise, we are particularly concerned about parasitic

resonance in the clock control loop. These resonant modes in the output, referred to as spurs in PLL literature, void the assumptions about phase noise and Gaussian deviates for jitter used in (1) and (2). Therefore, proper characterization and compensation of the clock control loop is critical to the implementation of impairment scenarios.

B. Optimized compensation of reference

To support the characterization of the PLL, we provided a sequence of constant step offsets of magnitude $.64\pi$, $.48\pi$, $.96\pi$, $.64\pi$ and $.32\pi$ updated every 14 seconds. We measured the phase of the output 10 MHz signal with respect to the input noise-free reference clock, 1000 times per second. The phase steps commanded and the actual output from are shown in Figure 5 labeled "$\Delta\phi$" and "$y\ ungoverned$" respectively.

Note the under-damped response of the clock control loop expressed as oscillations in $y\ ungoverned$ following each step change in $\Delta\phi$. Also note that the magnitude of the overshoot is affected by the magnitude of the step change commanded. Tuning of the Proportional-Integral (PI) controller and feedback gains in the PLL may reduce overshoot but typically also introduces trade-offs in the form of increased rise and settling times. Further, manual compensation is not always possible for impairment functions with arbitrary magnitudes. In order to ensure that the TIM can implement as wide a set of scenarios as possible, we will consider the use of a reference governor (described in Section III-B) as an add-on scheme for enforcing pointwise-in-time output constraints. This approach differs from methods that rely on tuning closed loop gains by automatically (and minimally) modifying the reference command to the PLL system to meet output constraints. The value of this approach is in our ability to directly enforce tolerances associated with impairment scenarios on the output of the TIM.

III. Initial results

A. System identification

Analog PLLs have been traditionally modeled by second-order linear dynamic equations. The linearity of the transient response of the PLL is based on a small angle assumption for the phase detector and knowledge of the closed-loop gains for the NCO. Our design calls for large phase steps and uses commercial PLL and NCO subcomponents, requiring us to first validate the closed loop response of the system via system identification in order to test the limit of the linear model and to determine the dynamic coefficients that may alter the actual output. Feed-forword system inversion was not used simply because the system model is not known precisely and has characteristics including a limited slew rate. Model uncertainty can be addressed in the reference governor to form a robust constraint management system.

A discrete-time model of the system is needed for the operation of a reference governor. Data from the PLL system was collected and a step response was analyzed. The percent overshoot and rise time were found and used to calculate the natural frequency (ω_n) and damping ratio (ζ) of a second-order linear time-invariant (LTI) system. ω_n and ζ were found to be 1.184 rad/s and 0.55 respectively.

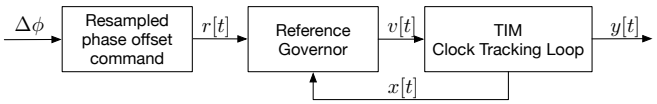

Fig. 4. Block diagram showing a scalar reference governor applied as an add-on to the clock control loop.

B. Reference Governor

1) Review: Reference Governor (RG) [24], [25] is an add-on scheme for enforcing pointwise-in-time state and control constraints by modifying, whenever required, the reference to a well-designed stable closed-loop system. A block diagram of a RG is shown in Figure 4, where $y[t]$ is the constrained output from the TIM, $r[t]$ is the pointwise-in-time reference signal generated from the commanded phase offset $\Delta\phi$, $v[t]$ is the governed reference, and $x[t]$ is the system state. To compute $v(t)$, a RG employs the so-called maximal admissible set (MAS) [26], which is defined as the set of all inputs and states that are constraint-admissible. By solving a simple linear program over this set, the RG selects a $v[t]$ that is as close as possible to $r[t]$ such that the constraints are satisfied for all time.

To provide a review, consider the discrete-time LTI system given by:

$$\begin{aligned} x[t+1] &= Ax[t] + Bv[t] \\ y[t] &= Cx[t] + Dv[t] \end{aligned} \tag{3}$$

where $x[t] \in \mathbb{R}^n$ is the state vector, $v[t] \in \mathbb{R}$ is the input, and $y[t] \in \mathbb{R}^m$ is the constrained output vector. Over the output the following constraints are imposed: $y[t] \in \mathbb{Y}, \forall t \in \mathbb{Z}_+$, where $\mathbb{Y}$ is a polytopic set. The input $v[t]$ is computed by the RG as a convex combination of the previous input $v[t-1]$, and the current reference $r[t]$. That is:

$$v[t] = v[t-1] + \kappa(r[t] - v[t-1]) \tag{4}$$

where κ is the solution of the following linear program:

$$\begin{aligned} \underset{\kappa\in[0,1]}{\text{maximize}} \quad & \kappa \\ \text{s.t.} \quad & v[t] = v[t-1] + \kappa(r[t] - v[t-1]) \\ & (x[t], u[t]) \in O_\infty \end{aligned} \tag{5}$$

where O_∞ is the Maximal Admissible Set (MAS) discussed below. Note that $\kappa = 0$ means that in order to keep the system safe, $v[t] = v[t-1]$, and $\kappa = 1$ means that no violation is detected and, therefore, $v[t] = r[t]$. This RG formulation ensures closed-loop stability and recursive feasibility.

The MAS is the set of all safe initial conditions and inputs, defined as:

$$O_\infty := \{(x_0, u_0) : x[0] = x_0, v[t] = v_0, y[t] \in \mathbb{Y}, \forall t \geq 0\}$$

To generate the MAS, we assume that $v[t] = v_0$ is held constant for all time. Computation of the MAS is possible,

as $y[t]$ can be expressed explicitly as a function of $x[0] = x_0$ and v_0:

$$y[t] = CA^t x_0 + (C(I - A^t)(I - A)^{-1}B + D)v_0 \tag{6}$$

The MAS can be computed using the above, and can be shown to be a polytope of the form:

$$O_\infty = \{(x_0, v_0) : H_x x_0 + H_v v_0 \leq h\} \tag{7}$$

Conditions for O_∞ to be finitely determined (i.e., matrices H_x, H_v, h to be finite dimensional) are discussed in [27]. Basically, to ensure that O_∞ is finitely determined, the steady-state constraint is first tightened:

$$H_0 v_0 \in (1 - \epsilon)\mathbb{Y}$$

where H_0 is the DC gain of system and ϵ is a small positive number. This constraint (i.e., set of inequalities) is introduced in O_∞ to ensure finite determinism. In the sequel, with some abuse of notation, we assume that O_∞ includes the tightened steady-state constraint and is, hence, finitely determined.

An efficient solution to (5) can be found by combining (4) and (7) to form

$$\kappa H_v(r[t] - v[t-1]) \leq h - H_x x[t] - H_v v[t-1] \tag{8}$$

By iteratively looping through (8), the maximum $\kappa \in [0, 1]$ that satisfies all the inequalities can be found. If no solution exists, $\kappa := 0$.

2) Application of the reference governor to the PLL system model: Modifications to the RG were made in order to apply it to the PLL system. While the previously mentioned RG sets a fixed constraint on the output, the modified RG continuously updates the constraint to match the reference. This modification was chosen because it allows for the under-damped PLL system to nearly maintain its rise time while eliminating overshoot and decreasing settling time.

To implement the modified RG, we set the output constraint to be: $\mathbb{Y} = \{y : y \leq r[t]\}$ whenever $y[t] \leq r[t]$, and $\mathbb{Y} = \{y : y \geq r[t]\}$ whenever $y[t] \geq r[t]$, where t denotes the current timestep. This is to ensure that overshoot on the output is eliminated both when the reference signal rises and when the reference falls.

To implement the above in an RG setting, the h matrix is initialized to $[1 - \epsilon; 1; ...; 1; 1]$ before creating the O_∞ set. We denote this matrix as h_{ones}. At each time-step, the governor reads the current reference and scales the h_{ones} matrix by the reference:

$$h = h_{ones} r[t] \tag{9}$$

The matrices H_v and H_x are not modified. The parameter κ is then calculated by maximizing $\kappa \in [0, 1]$ over the constraints imposed by the following logic:

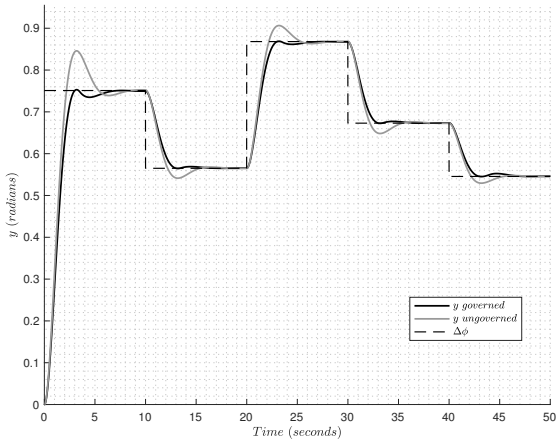

Fig. 5. Governed and ungoverned responses of PLL system model to step references. The ungoverned response shows the closep loop response of the TIM. The governed response shows significantly reduced overshoot.

$$\begin{aligned}
&if\ y[t-1] < (1-\epsilon)r[t] \\
&\quad \kappa H_v(r[t] - v[t-1]) \leq h - H_x x[t] - H_v v[t-1] \\
&else\ if\ y[t-1] > (1+\epsilon)r[t] \\
&\quad \kappa H_v(r[t] - v[t-1]) \geq h - H_x x[t] - H_v v[t-1] \\
&else \\
&\quad \kappa := 0
\end{aligned} \tag{10}$$

Note that hysteresis ($1 - \epsilon$ and $1 + \epsilon$ terms) are included to prevent toggling in the presence of noise. Also, instead of comparing $y[t]$ with $(1 - \epsilon)r[t]$ and $(1 + \epsilon)r[t]$, we compare $y[t - 1]$ with them to account for cases in which $D \neq 0$.

Simulation results of the modified RG operating at a sample frequency of 0.5 Hz and subject to the second order PLL system model can be found in Figures 5, 6, 7 and 8.

The governed step responses in Figure 5 have similar rise times and reduced settling times compared to the ungoverned step responses. Most notably, the overshoot, apparent in the ungoverned system response, is eliminated via implementation of the modified RG.

Figures 6 and 7 demonstrate that the ability of the PLL system to track systematic impairments such as a linear and quadratic drift in phase is nearly identical for the governed and ungoverned systems. It is expected that the responses should be the same because the ungoverned response does not overshoot the reference. However, the reason for the slight discrepancy between the two responses is due to a relatively slow RG sample time compared to the PLL sample time.

Lastly, we evaluate the performance of the governed system when tracking a sinusoidal reference, as shown in Figure 8. Observe that the ungoverned system has positive steady state gain. This is an artifact of local resonance in the ungoverned PLL system as illustrated in the numerically computed fre-

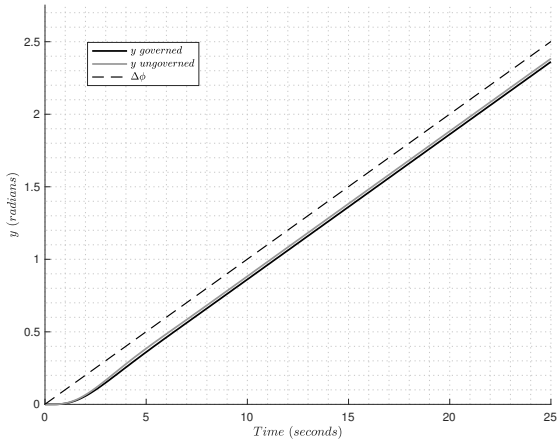

Fig. 6. Governed and ungoverned responses of PLL system model to a linear phase drift.

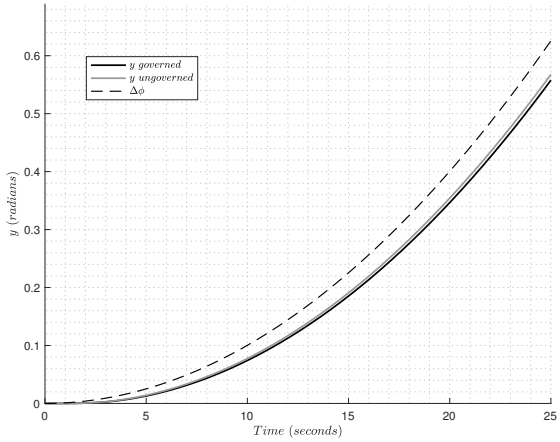

Fig. 7. Governed and ungoverned responses of PLL system model to a quadratic phase drift.

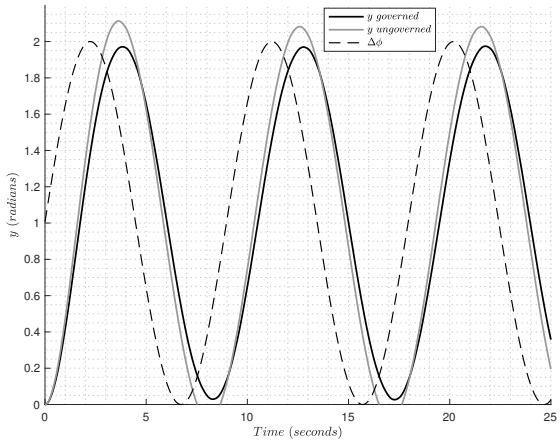

Fig. 8. Governed and ungoverned responses of PLL system model to a 0.11 Hz sinusoidal phase offset.

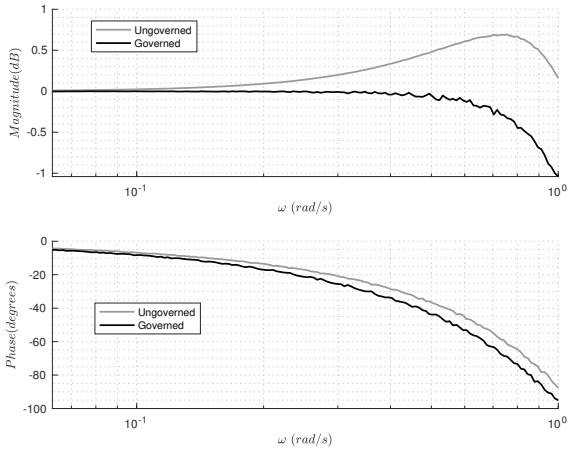

Fig. 9. Frequency response of the governed and ungoverned systems.

quency response plots for the governed and ungoverned systems in Figure 9. The figure shows that the governed system maintains a gain of 0 dB up to about 0.5 rad/s and eliminates the resonance at 0.8 rad/s.

IV. Conclusion

This paper presents an overview of a device under development to generate timing impairment scenarios as part of a hardware-in-the-loop test system for synchrometrology sensors. This Timing Impairment Module is comprised of hardware components that apply commanded phase offsets to the output of a numerically controlled oscillator.

We also discuss two software components of the system: The first is an approach to generate finite time implementations of stochastic functions including power-law noise while ensuring output sequences are sufficiently random. The other major contribution is the application of a Reference Governor algorithm to pre-condition the commanded phase reference in order to enforce output and control constraints. These capabilities allow the system to accurately reproduce real-world impairment scenarios while compensating for dynamics in the clock control system.

Disclaimer

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States. Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

Parts of this paper may have been presented in technical seminars and included in government publications. Recorded versions of those seminars and copyright free versions of publications are available through the National Institute of Standards and Technology at https://www.nist.gov/publications.

REFERENCES

[1] A. Goldstein, "2014 NIST assessment of phasor measurement unit performance," NIST, NISTIR 8106, Feb 2016.

[2] NASPI Time Synchronization Task Force, "Time synchronization in the electric power system," NASPI, Technical Report TR-001, March 2017.

[3] E. J. S. Leite, F. V. Lopes, F. B. Costa, and W. L. A. Neves, "Closed-form solution for traveling wave-based fault location on non-homogeneous lines," *IEEE Transactions on Power Delivery*, vol. 34, no. 3, pp. 1138–1150, 2019.

[4] A. Borghetti, M. Bosetti, M. Di Silvestro, C. A. Nucci, and M. Paolone, "Continuous-wavelet transform for fault location in distribution power networks: Definition of mother wavelets inferred from fault originated transients," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 380–388, 2008.

[5] D. Anand, A. Goldstein, Y.-S. Li-Baboud, C. Nguyen, and M. Weiss, "Timing challenges in the smart grid," NIST, Special Publication 1500-08, Jan 2017.

[6] Power System Relaying Committee, "IEEE standards for synchrophasor measurements for power systems-IEEE std c37. 118.1-2011," *New York, USA*, 2011.

[7] M. S. Almas, "Synchrophasor applications and their vulnerability to time synchronization impairment," Ph.D. dissertation, KTH Royal Institute of Technology, 2017.

[8] A. Goldstein, D. Anand, and Y.-S. Li-Baboud, "Investigation of pmu response to leap second: 2015," NIST, NISTIR 8077, Aug 2015.

[9] A. G. Phadke and B. Kasztenny, "Synchronized phasor and frequency measurement under transient conditions," *IEEE Transactions on Power Delivery*, vol. 24, no. 1, pp. 89–95, 2008.

[10] W. Kester, "Converting oscillator phase noise to time jitter," *Tutorial MT-008, Analog Devices*, p. 2009, 2009.

[11] G. Betta, C. Liguori, and A. Pietrosanto, "Propagation of uncertainty in a discrete fourier transform algorithm," *Measurement*, vol. 27, no. 4, pp. 231–239, 2000.

[12] J. A. Momoh, *Electric power distribution, automation, protection, and control*. CRC press, 2017.

[13] K. Martin, G. Brunello, M. Adamiak, G. Antonova, M. Begovic, G. Benmouyal, P. Bui, H. Falk, V. Gharpure, A. Goldstein *et al.*, "An overview of the IEEE standard C37. 118.2 synchrophasor data transfer for power systems," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1980–1984, 2014.

[14] F. Steinhauser, C. Riesch, and M. Rudigier, "IEEE 1588 for time synchronization of devices in the electric power industry," in *2010 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE, 2010, pp. 1–6.

[15] NASPI PMU Applications Requirements Task Force, "PMU data quality: Methodology for examining data quality impacts to synchrophasor applications," NASPI, Technical Report TR-002, March 2017.

[16] *M88 Managed Clock Engine (MCE) Module*, 2nd ed., Qulsar Inc., 90 Great Oaks Blvd., Suite 204 San Jose, CA 95119 USA, Jan 2016.

[17] ITU-T, "G.8261-timing and synchronization aspects in packet networks," *ITU Std.*, 2008.

[18] J. C. Eidson, M. Fisher, and J. White, "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std*, pp. 1588–2002, 2002.

[19] D. Sullivan, D. Allan, D. Howe, and F. Walls, "Characterization of clocks and oscillators," NIST, Technical Report TN-1337, Jan 1990.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C*. Cambridge University Press, 1990, ch. 7, pp. 204–217.

[21] N. J. Kasdin, "Discrete simulation of colored noise and stochastic processes and $1/f^{\alpha}$ power law noise generation," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 802–827, 1995.

[22] N. Kasdin and T. Walter, "Discrete Simulation of Power Law Noise," in *Proceedings of the IEEE Frequency Control Symposium*, Stanford, CA, 1992.

[23] N. Ashby, "Probability distributions and confidence intervals for simulated power law noise," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 62, no. 1, pp. 116–128, 2015.

[24] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *2014 American Control Conference*. IEEE, 2014, pp. 226–241.

[25] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[26] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transactions on Automatic control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[27] E. G. Gilbert and I. Kolmanovsky, "Discrete-time reference governors for systems with state and control constraints and disturbance inputs," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 2. IEEE, 1995, pp. 1189–1194.