Model-free iterative control of repetitive dynamics for high-speed scanning in atomic force microscopy

Yang Li and John Bechhoefer[a)]

Department of Physics, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada

(Received 4 November 2008; accepted 14 December 2008; published online 8 January 2009)

We introduce an algorithm for calculating, offline or in real time and with no explicit system characterization, the feedforward input required for repetitive motions of a system. The algorithm is based on the secant method of numerical analysis and gives accurate motion at frequencies limited only by the signal-to-noise ratio and the actuator power and range. We illustrate the secant-solver algorithm on a stage used for atomic force microscopy. © *2009 American Institute of Physics.*

[DOI: 10.1063/1.3065093]

I. INTRODUCTION

Many physics measurements and techniques require periodic alteration of a parameter. The desired temporal modulation may be sinusoidal (e.g., lock-in techniques), square wave (switching), or triangular (raster scanning). If the modulation frequency is low, then a quasistatic approximation, with input simply proportional to the desired output, often suffices. For faster modulation, however, the dynamics of the experimental apparatus will distort the input.[1] To the extent that dynamics are well-approximated by a linear, well-characterized, and time-invariant system, one can design a prefilter that will alter the desired command signal to one that compensates for the dynamics by (approximately) inverting them.[2,3] However when dynamics are nonlinear, poorly characterized, and change over time, it is difficult to make such feedforward approaches work robustly. In particular, for weakly damped mechanical systems such as the scanning stage for atomic force microscopes (AFM) discussed here, it is difficult to generate an accurate waveform at frequencies exceeding 10%–20% of the first mechanical resonance frequency. Feedback methods are also limited by mechanical resonances, whose effect can be mitigated only to the extent that the frequencies of the resonances are known and stable.[4]

In a different approach, repetitive dynamics may be learned iteratively. Such "iterative learning control" (ILC) approaches have mostly been formulated in the time domain. (For recent reviews, see Refs. 5 and 6). A successful application to scanning stages has very recently been presented.[7] Approaches involving Fourier transforms of successive signal blocks, with computations done in the frequency domain, were also proven quite successful,[8,9] and the method presented here falls within this overall class. As we shall see, one advantage of working in the frequency domain is that there are straightforward algorithms that do not require prior measurement of system properties. In addition, the frequency-domain implementation of ILC can be simpler than the corresponding time-domain algorithm.[7]

For frequency-domain ILC, the general strategy, at step k, is to apply a control signal $u_k(t)$ for one or several periods while measuring the system response $y_k(t)$ relative to that desired, $y_d(t)$. One then Fourier transforms all signals, updates the next output u_{k+1} in the frequency domain, and finally transforms back to the time domain to output the new $u_{k+1}(t)$. See Fig. 1 for a block-diagram summary of the approach. These techniques have been applied to the control of scanning stages for AFM. The first such application, by Tien *et al.*,[9] used an experimentally obtained estimate of the linear transfer function $G(\omega)=y(\omega)/u(\omega)$ in the update rule. It works well for nearly linear systems with well-characterized dynamics. At high frequencies, where the error in the transfer-function estimate is large, one simply truncates the Fourier series. This gives an upper limit to the frequencies that can be tracked.

The updated rule, applied to each frequency component, is of the form $u_k=F(u_{k-1},y_{k-1};G)$. That is, $u_k=u_k(\omega_n)$, where $\omega_n=2\pi n/T$, with T as the duration of each block and $n=\{1,2,\ldots,N\}$. Each block duration T equals one or several periods of the desired motion. The highest frequency is set by $N=T/(2\Delta t)$, with Δt as the sampling interval. Once the new output Fourier coefficients are calculated via the update rule, the new signal $u_k(t)$ is calculated by summing the finite approximation to the Fourier series and then is output to the system.

More recently, we[10] and Kim and Zou[11] independently presented a model-free variant of the iterative method described above that estimates the transfer function simultaneously with the updates. As we shall discuss, the performance of that algorithm is limited by nonlinearities in the dynamical system being controlled. The basic problem is that the model-free algorithm considers each frequency independently, while nonlinearities couple different frequencies.

Here, we introduce a different model-free iterative method that correctly compensates for arbitrary nonlinearities, as long as the output remains periodic, or nearly so. Previous work by Cai and co-workers[12,13] also considered frequency coupling but used a different control algorithm, discussed below. The method presented here is simple and is limited only by noise and the physical limitations of sensors

[a)] Electronic mail: johnb@sfu.ca.

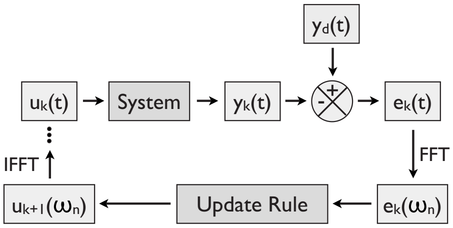

FIG. 1. Block diagram illustrating iterative learning control in the Fourier domain. At the kth iteration, a control signal $u_k(t)$ is sent to the system. The response $y_k(t)$ is compared with the desired signal $y_d(t)$, giving the error $e_k(t)$ at the kth iteration. To update the control signal, one Fourier transforms the error, obtaining a set of N Fourier coefficients, $e_k(\omega_n)$. One applies the update rule to these coefficients, giving a new set of Fourier coefficients for the control, $u_{k+1}(\omega_n)$. These are then inverted to form the new input $u_{k+1}(t)$, which is sent to the system, repeating the cycle.

and actuators—not the properties of the system to be controlled (e.g., mechanical resonances). As an application, we show that our new algorithm improves the speed and accuracy achievable by an AFM scanner.

II. DESCRIPTION OF THE ALGORITHM

A. Linear solver

We begin by describing the existing model-free iterative algorithm for calculating repetitive motion.[10,11] For reasons discussed below, we refer to this algorithm as the "linear solver." The linear-solver routine is initialized by using a naive estimate of the transfer function, $u_0 = y_d$. (We assume unity dc gain. A nonunity gain can be absorbed into the definition of u). Subsequent inputs are complex amplitudes $u_k(\omega_n)$ $(k \geq 1)$, which are calculated as

$$u_k = u_{k-1}\frac{y_d}{y_{k-1}}, \tag{1}$$

where we have dropped the explicit frequency dependence. We note that, in Eq. (1), $u_k(\omega_n)=0$ if $y_d(\omega_n)=0$. Also, since Eq. (1) is complex, it implies separate relationships for the magnitude and phase of each Fourier component (i.e., for $n=1,\ldots,N$).

The application of the linear-solver algorithm to an AFM scanning stage is illustrated in the left column of Fig. 2. The piezoelectric stage (Mad City Laboratories H100, with 100 μm range) has a mechanical resonance at 440 Hz (with a sample holder mass of 113 g) and a position-measuring sensor whose output is used by an internal feedback loop that linearizes the stage's performance at low frequencies. (The feedback algorithm has a two-pole low-pass filter with a bandwidth of 27 Hz, and a 10 Hz triangle wave already shows noticeable phase lag and rounding of the corners). We note that our work complements other efforts to increase AFM scan speed by raising the frequency of mechanical resonances.[14–16] The desired signal [Fig. 2(b)] here is a 1 μm (peak-to-peak amplitude) sinusoid at 500 Hz. Because the frequency of the desired signal is above both the mechanical resonance and the low-pass filter bandwidth, the control signal must be ≈100 times larger and must lead by almost a 2π phase shift. Figure 2(c) shows the residuals between the desired and measured signals in Fig. 2(b), and Fig. 2(e) shows that the residual error converges to a relatively small value (3.7% in this case), after just a few iterations, even though the initial guess u_0 is poor. Note that the linear-solver routine naturally leads to a zero-phase-delay, noncausal filter. Because the desired motion is periodic, the dynamics are known in advance and can be used to anticipate the required changes in the stage's motion.

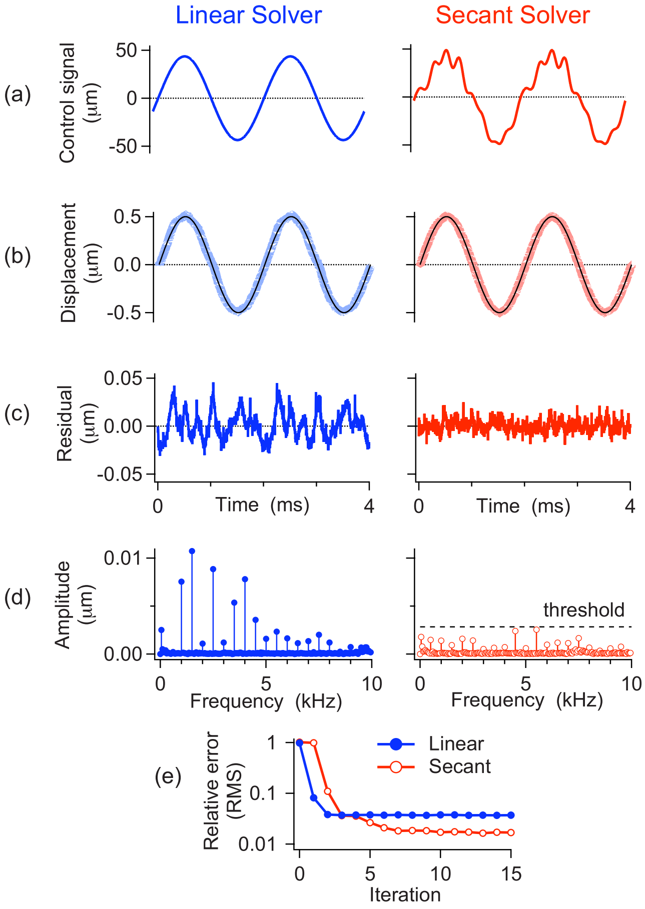

FIG. 2. (Color online) Comparison of the linear and secant solvers (left and right columns, respectively). (a) Control signal sent to stage. (b) Stage displacement in response to control. Thin line is the desired sinusoidal waveform (500 Hz, 1 μm peak-to-peak amplitude). (c) Residual error between desired and measured waveforms. (d) Fourier amplitudes of residuals. (e) Root-mean-square error, normalized by the waveform itself, as a function of iterations of the solver.

Despite the reasonable accuracy and fast convergence of the linear solver, the method has one flaw: examining the amplitude spectrum of the residuals [Fig. 2(d)] shows significant amplitudes at integer harmonics of the command signal of 500 Hz. These arise because the linear-solver algorithm [Eq. (1)] implies that a Fourier coefficient not present in the desired waveform will not be present in the control signal. Since the desired waveform has only one nonzero Fourier component, at 500 Hz, the control signal must also be a 500 Hz sine wave, with only its amplitude and phase relative to the desired sinusoid available to be set. However, the nonlinear response of the piezoelectric stage generates higher harmonics in the output, which the linear-solver algorithm cannot remove. Its performance is thus limited by the strength of the nonlinearities in the dynamics being controlled.

B. Secant solver

We now introduce an alternative model-free iterative algorithm that overcomes this defect of the linear solver. We begin by noting that at each iteration k, we seek a set of complex Fourier amplitudes u_k that leads to an output y_k equal to the desired signal y_d. Thus, we need $e(u)=0$, where the error $e=y_d-y$ is the difference between the desired and measured signals. At a given iteration k, this is a set of N complex functions (one for each Fourier component), which each depend on the N complex amplitudes u. The functions $e(u)$ are nonlinear, unknown, and subject to change. The responses can be measured: a signal $u_k(t)$ input to the system implies a set of Fourier coefficients $u_k(\omega_n)$. The measured output $e_k(t)$ then implies a set of Fourier coefficients $e_k(\omega_n)$, with $n=1,\ldots,N$ the set of frequencies of the Fourier series. Our problem is then to find a root u of the nonlinear, N-component, frequency-domain function $e(u)$. Although root finding is a standard problem of numerical analysis,[17] the problem here has three special features.

(1) We do not know the function $e(u)$, although we can evaluate it experimentally for a given u. Neither do we know the Jacobian (derivative) of e, although we can estimate it by finite differences.

(2) Noise in both the measurement of the stage's position and in the output amplifiers going to the stage imply that e is stochastic.

(3) While many classical numerical techniques seek to minimize the number of numerical calculations, our rate-limiting step is the evaluation of e given a particular input u. Thus, we seek a numerical method that minimizes the number of function evaluations.

Such problems of stochastic search and optimization are also well known and have numerical techniques that are well adapted to the kinds of constraints discussed above.[18] Here, we choose a relatively simple strategy that nonetheless performs well.

We first note that we cannot use algorithms based on Newton's method since we do not have an explicit way to evaluate the Jacobian. We next note that algorithms, such as Broyden's method,[17] that solve the N equations simultaneously take a long time to converge because they need N^2 finite-difference evaluations for the initial estimate of the Jacobian. We shall show that a method based on the iterative solution for each frequency component of u works well, in practice.

Our approach is based on the secant method of numerical analysis,[17] applied separately to each Fourier component. (Although normally applied to real equations, the secant method also works well on complex equations). Because we apply the method to each component separately, the nonlinearities in e must be small enough that cycling sequentially through each component in a given iteration still leads to convergence. This works in the experimental example considered here.

Explicitly, our "secant solver" is formulated for each Fourier component by initializing u_0 to a random value, $u_1=0$, and then updating u_k for $k\geq 2$ by

$$\delta u_k = - J_{k-1}^{-1} e_{k-1}, \quad k \geq 2,$$

where $\delta u_k \equiv u_k - u_{k-1}$, $e_{k-1}=y_d-y_{k-1}$, and J_{k-1}^{-1} is the inverse of the finite-difference approximation to the Jacobian

$$J_{k-1} = \left(\frac{e_{k-1}-e_{k-2}}{u_{k-1}-u_{k-2}}\right) = -\left(\frac{y_{k-1}-y_{k-2}}{u_{k-1}-u_{k-2}}\right). \tag{2}$$

Replacing the finite-difference approximation in Eq. (2) with the analytic Jacobian would lead to Newton's method.

Because $e(u)$ is a stochastic function, we supplement Eq. (2) with the rule that if e_{k-1} is less than some threshold value, we keep $u_k=u_{k-1}$. This rule is necessary because when the iterated control signals u_k become too close to each other, the errors in the finite-difference approximation to J become large and can lead to instability. Choosing a threshold value ten times the typical noise level of e measurements ensures convergence in practice. Note that even after several iterations where u_k is held constant, any error exceeding the threshold automatically restarts the updating of u.

We illustrate the secant solver in Fig. 2 (right column), under conditions identical to those used for the linear solver. After the two initial guesses, the method converges to an error of 1.7%, one-half that of the linear method. While the improvement is already significant, we note that the error can be decreased arbitrarily by averaging several spectra each iteration. The reduced noise level in the spectral components allows one to reduce the threshold accordingly. Thus, while the performance of the linear solver is limited by the size of nonlinearities, the secant method is not. Indeed, its only real limitations are those of the actuator. Here, we already have a signal that is only 1% of the input value. We can only go to higher frequencies by further reducing the scanning range.

We conclude with several comments.

(1) We typically measure for a block period that is ten times the period of the desired waveform. While, in principle, one could choose the block period equal to the waveform period, there are advantages to choosing an integer multiple. First, the longer period effectively averages over noise. Second, nonlinearities may generate subharmonic response, which can be compensated if finer frequency intervals are measured. Third, sustained external disturbances, whose frequencies will in general differ from the block period, are also canceled in this scheme, and finer frequency resolution allows for more accurate cancellation.

(2) While the linear method uses only one initial guess, the secant method requires two to estimate the local derivative. We choose a common random amplitude and random phases for the Fourier components of the initial output signal, u_0. (If the output is close to the input, setting $u_0=y_0+\text{rand}$ will speed convergence). We next set $u_1=0$. The threshold algorithm will then keep coefficients zero unless an error signal needs to be corrected. This choice of initial conditions improves the convergence rate and accuracy.

(3) We can relate the secant to the linear solver by noting that we can rewrite the update law in Eq. (1) in the form of Eq. (2), replacing the Jacobian of Eq. (2) by

$$J_{k-1} = \left(\frac{e_{k-1}}{u_{k-1}}\right) = -\left(\frac{y_{k-1}}{u_{k-1}}\right). \tag{3}$$

The new secant solver thus reduces to the old linear solver if we set $u_{k-2}=y_{k-2}=0$. This works if the output y is a linear function of the control signal u, with no offset, but fails when $u=0$ produces a nonzero output y.

(4) We have also examined Muller's method, which generalizes the secant method by fitting a parabola through the last three observations and using the root of that curve as the next iterate. (The secant method uses a straight line through the last two observations). In principle, the higher order of Muller's method should enlarge the basin of stability, implying that a lower accuracy is required of the initial guess. We found that the performance in practice was similar. Since the algorithm is more complicated and needs an extra initial guess, we prefer the secant solver.

(5) The method introduced by Tang *et al.*[12] is also based on the estimation of the Fourier coefficients of the system input $u(t)$. In a first approach, they also give an algorithm that applies separately to each frequency component. Specifically, they use a proportional-derivative feedback scheme on each error coefficient $e_k(\omega_n)$ combined with an integral-feedback estimate of the feedforward term needed to cancel the remaining dynamics. Such a feedback algorithm will converge more slowly than the secant method. Since it is based on a linear dynamical system, the convergence is exponential: $e_{k+1} \sim a e_k$, with the constant $a<1$. (The value of a depends on the feedback parameters, which need to be tuned). By contrast, the secant method has no parameters and converges superexponentially, with $e_{k+1} \sim a(e_k)^{1.618}$ (Ref. 17). In subsequent work, Qing and Cai[13] extended their method to handle the corrections to each Fourier coefficient simultaneously rather than individually. In this case, the range of convergence may be larger than with the method presented here.

III. APPLICATION TO AFM SCANNERS

We illustrate the secant method on a waveform useful for AFM scanners. The standard desired waveform for scanning probe microscopes is a triangle wave, corresponding to constant left and right scanning velocities. (Occasionally, a sawtooth waveform is used). One problem with such waveforms is that no physical scanner can achieve the infinite accelerations required to instantaneously reverse the velocity. Here, we propose a slightly different waveform, the "rounded triangle," which consists of straight-line segments smoothly joined to parabolic ends [Fig. 3(a)]. By replacing the instant turnaround with a finite acceleration at either end of the range, one has a motion that can be achieved. The cost is a small "overscan" at each end in which the stage reverses its velocity. It is a simple matter to trim this region from the image. Of course, the result trying to achieve a triangle waveform will also be a waveform with rounded ends; however, the difference here is that the turnaround region is controlled, with a predictable response. A slightly more mathematical view of the situation is to note that a triangle wave is $\mathcal{C}^1$ (discontinuity in the first derivative), which leads to Fourier coefficients n that decrease as $1/n^2$. The rounded triangle is $\mathcal{C}^2$, with Fourier coefficients that decrease as $1/n^3$. The more rapid decrease of Fourier coefficients for a rounded triangle allows one to increase the control bandwidth.

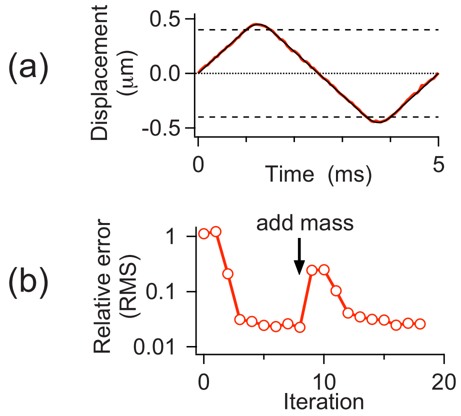

FIG. 3. (Color online) (a) Comparison between the desired and achieved rounded-triangle waveform. Frequency=200 Hz. The amplitude of linear motion is 800 nm peak-to-peak, and there is a 50 nm overscan on either side. The limits to the linear part of the waveform are indicated by the two dashed lines. (b) Root-mean-square error, normalized by the waveform itself, as a function of iterations of the solver. After iteration eight, a large (200 g) mass was placed on the stage.

Figure 3 shows the response to a desired signal that is a rounded triangle. The linear part of the waveform is contained between the two dashed lines (800 nm, peak to peak). The turnaround region is 50 nm on either side, corresponding to an overscan of 12.5%. The tracking bandwidth is 10 kHz. At iteration 8, we add a large (200 g) mass to the stage. After a brief transient, the desired waveform is recovered, illustrating the ability to adaptively track changes in system dynamics.

We also explore the effects of scan size. By reducing the scan frequency from 200 to 40 Hz, we can increase the linear part of the scan to 20 μm, peak to peak. Figure 4 shows the control signal calculated by the secant method (a), desired waveform and response (b), residuals between desired waveform and response (c), and magnitude of the Fourier spectrum of the residuals (d). Note that the size of the residuals in (c) is very similar to that seen in Fig. 2(c). (The error threshold is the same in both cases). Because the scan is 20 times larger than in Fig. 2, the relative error is correspondingly smaller (0.12% versus 1.7%)—despite the increase in hysteresis and nonlinearity expected for the larger scan range. In other words, the large-scan results support the claim that the performance of the algorithm is not limited by nonlinearities (as is that of the linear algorithm) but rather by the sensor noise level and actuator range. [In Fig. 4(a), we see that the control signal to the actuator has reached its maximum 100 μm amplitude range and can go no higher].

The results presented here were all obtained "offline," with no phase coherence between the waveforms obtained in each iteration. However, it is straightforward to achieve "pseudo-real-time" operation, where each iteration of the

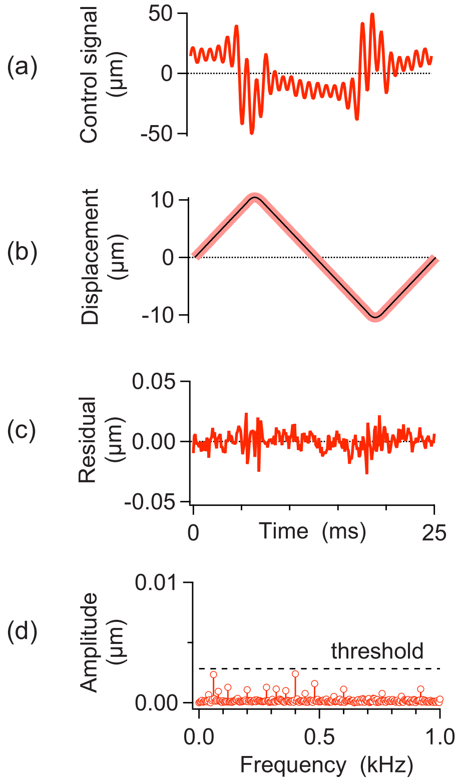

FIG. 4. (Color online) Performance of the secant algorithm for a large scan. (a) Control signal sent to stage. (b) Stage displacement in response to control. Thin line is the desired rounded-triangle waveform (40 Hz, 20 μm peak-to-peak linear range). (c) Residual error between desired and measured waveforms. (d) Fourier amplitudes of residuals.

output is updated coherently, with no phase slip. Such a version would have real-time adaptivity and could compensate for periodic disturbances, as well as changes in the dynamics. Both the offline and pseudo-real-time methods are easy to implement with high bandwidth because one digitizes the signal in blocks. Thus, any ordinary data acquisition card can be used, with acquisition rates that are at 1 MHz for standard cards and can exceed 1 GHz for specialized high-speed cards with onboard memory. The actual interaction occurs at the rate that different blocks are acquired, which can be orders of magnitude slower. The update rate can always be chosen longer than the calculation times, which, on a current computer, are not significant.

Finally, we emphasize that while the hardware limitations of our scanner (resonance at 440 Hz, internal low-pass filter on the hardware feedback loop) limit scan frequencies to a few hundred hertz, the methods we propose may be easily implemented on the faster scanners that have been recently demonstrated, whose resonances are in the 5–20 kHz range.[14–16] One can anticipate scan rates at such resonant frequencies. These would be 5–10 times faster than the rates reported in the above references. The recently published time-domain ILC method of Bristow *et al.*[7] also demonstrates tracking above the primary resonance but depends on prior measurements of the transfer function.

IV. CONCLUSION

We have presented an iterative method for calculating the input to an unknown, nonlinear dynamical system for repetitive output motions. The algorithm allows control of any system where a periodic input leads to a periodic output (i.e., excluding chaos) and generates an output that has no phase delay relative to the input. It requires no offline measurements for system characterization. Deviations from periodicity that occur infrequently or at frequencies less than the update rate can be compensated adaptively. The algorithm allows operation at frequencies that are limited only by the noise in measurements (which can be reduced by averaging spectra) and the physical limitations of the actuators used. In particular, we have demonstrated that one may control motion above the primary resonance frequency of a piezoelectric stage used for AFM imaging. The algorithm will be useful for increasing the speed of instruments such as AFMs and may also find industrial applications, for example in speeding the motion of robots in factories.[19]

ACKNOWLEDGMENTS

This research was supported by NSERC Discovery Grant (Canada).

[1] S. Devasia, E. Eleftheriou, and S. O. R. Moheimani, IEEE Trans. Control Syst. Technol. **15**, 802 (2007).
[2] D. Croft and S. Devasia, Rev. Sci. Instrum. **70**, 4600 (1999).
[3] Y. Li and J. Bechhoefer, Rev. Sci. Instrum. **78**, 013702 (2007).
[4] A. C. Shegaonkar and S. M. Salapaka, Appl. Phys. Lett. **91**, 203513 (2007).
[5] D. A. Bristow, M. Tharayil, and A. G. Alleyne, IEEE Control Syst. Mag. **26**, 96 (2006).
[6] H.-S. Ahn, Y.-Q. Chen, and K. L. Moore, IEEE Trans. Syst. Man Cybern., Part C Appl. Rev. **37**, 1099 (2007).
[7] D. A. Bristow, J. Dong, A. G. Alleyne, P. Ferriera, and S. Salapaka, Rev. Sci. Instrum. **79**, 103704 (2008).
[8] C. Kempf, W. Messner, M. Tomizuka, and R. Horowitz, IEEE Control Syst. Mag. **13**, 48 (1993).
[9] S. Tien, Q. Zou, and S. Devasia, IEEE Trans. Control Syst. Technol. **13**, 921 (2005).
[10] Y. Li and J. Bechhoefer, Proceedings of the American Control Conference, Seattle, WA, 2008 (unpublished), pp. 2703–2709.
[11] K.-Y. Kim and Q. Zou, Proceedings of the American Control Conference, Seattle, WA, 2008 (unpublished), pp. 2710–2715.
[12] X. Tang, L. Cai, and W. Huang, IEEE Trans. Rob. Autom. **16**, 1042 (2000).
[13] W. Qin and L. Cai, Proceedings of the IEEE/ASME Conference on Advanced Intelligent Mechatronics, 2001 (unpublished), p. 482.
[14] G. Schitter, K. J. Aström, B. E. DeMartini, P. J. Thurner, K. L. Turner, and P. K. Hansma, IEEE Trans. Control Syst. Technol. **15**, 906 (2007).
[15] K. K. Leang and A. J. Fleming, Proceedings of the American Control Conference, Seattle, WA, 2008 (unpublished), pp. 3188–3193.
[16] T. Fukuma, Y. Okazaki, N. Kodera, T. Uchihashi, and T. Ando, Appl. Phys. Lett. **92**, 243119 (2008).
[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, UK, 2007).
[18] J. C. Spall, *Introduction to Stochastic Search and Optimization* (Wiley, Hoboken, NJ, 1993).
[19] K. L. Moore and J.-X. Xu, Int. J. Control **73**, 955 (2000).