

Digital Emulation of Pulse Frequency Modulation for Neuroprosthetic Sensory Feedback

Jake J. Abbott, *Member, IEEE*, and Sanford G. Meek

Abstract—Pulse frequency modulation (PFM) is a method of encoding information where the instantaneous frequency of a pulse train carries the signal's information. PFM is of particular interest to those working towards interfacing prosthetic devices directly with the human nervous system. In this paper, we consider the effects of directly implementing PFM with a digital microprocessor. We consider three digital PFM algorithms: two are deterministic, and the third has a probabilistic nature that has desirable time-averaged and ensemble behavior. For each algorithm, we analytically bound the error between the desired pulse frequency and the actual frequency output by the microprocessor. We aim to provide tools for the design and analysis of closed-loop neuroprosthetic systems containing PFM.

Index Terms—Peripheral nervous system, prosthesis, pulse frequency modulation (PFM).

I. INTRODUCTION

PULSE FREQUENCY MODULATION (PFM) is a method of encoding information where the instantaneous frequency of a pulse train carries the signal's information. PFM is currently of particular interest to those researchers working towards communicating directly with the human central nervous system, since the nervous system uses PFM to transmit information [1], [2]; the neural "spikes" arise from chemical action potentials at the cellular level [3]. We are particularly interested in PFM for the direct neural control of a prosthetic arm and hand through the peripheral nervous system (PNS).

Direct cortical control of robot manipulators has been achieved by utilizing large numbers of neural signals [4], making use of the stochastic nature of neural spikes. Alternatively, researchers have worked toward the design of prostheses to be interfaced to the PNS [5]. A naturally functioning prosthesis controlled by the PNS will require both efferent and afferent communication. Prior work has considered the ability to control muscle forces by inputting pulses trains directly into the PNS [6], as well as the detection of individual neural spikes from noisy data [7]. An active area of research is the design of devices for stimulating the PNS [8], [9]. Work has begun to consider the input of PFM sensor data back into the nervous



Fig. 1. University of Utah experimental neuroprosthetic hand.

system through afferent nerves [10], [11]. A thorough review of prior work in neuroprosthetics is given in [2].

Recently, Dhillon and Horch [12] presented the first demonstration of direct neural control of an artificial arm with direct sensory feedback into the PNS. They use longitudinal intrafascicular electrodes [9], which excite a small number of neurons in the local area of the electrode. Consequently, there is a need for accurate individual neural stimulation since the spatial averaging of multiple neurons is not available. The experimental prosthetic hand being considered at the University of Utah, which is equipped with position/velocity sensing and three-degrees-of-freedom force sensing, is shown in Fig. 1; the prototype was presented in [13].

PFM was first analyzed decades ago [14], [15], and was later applied to electromechanical control systems [16]. The analysis and design of systems containing PFM is difficult because of the nonlinear and time-varying nature of PFM. The stability of PFM control systems has been investigated, typically taking the form of analysis and prediction of potential limit cycles [17]. Previous work on PFM control systems has assumed an ideal (analog) implementation of PFM, but the design of neuroprostheses will likely involve digital microprocessors implementing high-level control systems. Prosthetic design will also put size and power consumption at a premium.

In this paper, we consider the direct implementation of PFM with a digital microprocessor. We consider the error that is incurred when we attempt to output a pulse train of a desired frequency from a microprocessor with its own constant sampling frequency. This error has the effect of limiting the resolution of afferent signals being fed into the PNS, with the error generally increasing with pulse frequency. We consider three digital PFM algorithms: the first two are deterministic (i.e., there is a single-valued mapping from the reference pulse frequency to the actual pulse frequency), and the third has a probabilistic nature that we demonstrate. For each algorithm, we analytically

Manuscript received June 6, 2006; revised October 27, 2006. This work was supported in part by the National Institutes of Health under Grant ROI NS27371 and in part by the Center for Engineering Design at the University of Utah.

J. J. Abbott was with the Department of Mechanical Engineering, University of Utah, Salt Lake City, UT 84112 USA. He is now with the Institute of Robotics and Intelligent Systems, ETH Zurich, 8092 Zurich, Switzerland (e-mail: jabbott@ethz.ch).

S. G. Meek is with the Department of Mechanical Engineering, University of Utah, Salt Lake City, UT 84112 USA (e-mail: meek@mech.utah.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNSRE.2007.891380

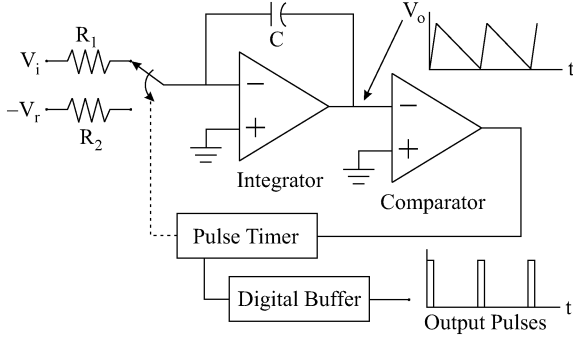


Fig. 2. V/F converter. The output pulse frequency is linearly proportional to the input voltage V_i .

quantify the error bounds. These errors impact on the resolution of sensory feedback, and will likely impact on the stability of closed-loop neuroprosthetic systems.

II. PULSE FREQUENCY MODULATION

An integral pulse frequency modulator is a device that emits a pulse whenever the magnitude of the integral of the input reaches a threshold value [14], [15]. For a system with input u and output y , the device is described by

$$\frac{dx}{dt} = u - \alpha \operatorname{sgn}(x) \delta(|x| - \alpha) \quad (1)$$

$$y = \operatorname{sgn}(x) \delta(|x| - \alpha) \quad (2)$$

where α is the threshold integral value to emit a pulse. This device can be realized with an analog circuit (with the exception of the ideal δ of unit area, infinite height, and infinitesimal duration). There is no single definition of this circuit, but Fig. 2 shows an example capturing the salient details, synthesized from two sources [18], [19]. This circuit is known as a “voltage-to-frequency converter.”

Neuroprostheses will likely make use of digital microprocessors to implement high-level control systems. Any sensory feedback calculated by the control system must be converted to a pulse train to be fed into the nervous system, and there are two basic methods to accomplish this. In the first, the microprocessor outputs a binary number, which is then converted to an analog voltage by a digital-to-analog (D/A) converter, which is then converted to a digital pulse train by a voltage-to-frequency (V/F) converter. In the second method, we remove the D/A and V/F converters, and output the digital pulse train directly from the microprocessor. In both methods, the digital pulse train is finally converted to an analog neural spike train and fed into an afferent nerve. It may be desirable to implement many parallel channels of feedback information, but, using the first method, this would also require many D/A and V/F converters, which may be prohibitive under power and space restrictions. In comparison, a single n -bit binary number could be used to generate n digital pulse trains directly. The resolution of this second method is ultimately limited by the sampling frequency of the microprocessor, and is the focus of this paper.

In the idealized analog version of PFM, the integrand is reset to zero at the occurrence of an output pulse. However, in a discrete controller, this simple notion of resetting the system becomes ambiguous, due to inherent limitations in numerical integration with a constant sampling frequency. Different numerical integration schemes can be used to emulate PFM, none of which perfectly recreate the ideal circuit. An additional fundamental limitation of digital PFM is that no pulse frequency

begin loop $I = I + T_s$ if $I \geq T_r$ Output pulse $I = 0$ end if end loop	begin loop $I = I + T_s$ if $T_r - I < 0.5T_s$ Output pulse $I = 0$ end if end loop	begin loop $I = I + T_s$ if $I \geq T_r$ Output pulse $I = I - T_r$ end if end loop
Algorithm 1	Algorithm 2	Algorithm 3

Fig. 3. Three digital PFM algorithms to output a pulse train with desired pulse period T_r using a constant sampling period T_s .

faster than the microprocessor sampling frequency can possibly be realized. We will assume that the desired frequency is always less than the sampling frequency.

Integral PFM is capable of outputting both positive and negative pulses. However, the human nervous system is only capable of transmitting pulses of a single sign—joint control is accomplished by antagonistic muscles, each only capable of pulling. This led to the development of parallel-path single-signed integral PFM [14]. Consequently, in this paper, we only consider the generation of single-signed pulses. In addition, for the purpose of analysis, we only consider constant reference frequencies. In general, there is a fundamental loss of information during pulse frequency modulation of a time-varying signal that we will not address. However, the insights gained here are applicable to time-varying reference frequencies.

For the remainder of this paper, we use the following terminology. The microprocessor has a constant sampling frequency f_s and sampling period $T_s = 1/f_s$. The desired pulse frequency is given as a reference f_r , and has an associated reference period $T_r = 1/f_r$. The actual output pulse period T_δ is updated at the occurrence of a pulse, and is measured as the time that has passed since the occurrence of the previous pulse. The output pulse frequency is defined as $f_\delta = 1/T_\delta$.

III. DIGITAL PFM ALGORITHMS

The first digital PFM algorithm that we consider is given in Fig. 3 as Algorithm 1. In this naive approach, the algorithm numerically integrates time until the value of the integrand crosses the threshold T_r . At this sample, a pulse is output and the integrand is reset to zero. Fig. 4 shows the output pulse frequency f_δ versus the reference pulse frequency f_r , both normalized by the sampling frequency f_s , for Algorithm 1. We observe that significant errors in pulse frequency occur for reference frequencies as low as 10% of the sampling frequency. It is also clear that the output frequency is always less than the reference frequency. The magnitude of the bound on the error in f_δ is given by

$$\frac{|f_\delta - f_r|}{f_r} = \frac{f_r}{f_s + f_r}. \quad (3)$$

For reference-frequency values that are very small relative to the sampling frequency ($f_r \ll f_s$), the bound of (3) can be approximated by the simple expression

$$\frac{|f_\delta - f_r|}{f_r} \approx \frac{f_r}{f_s}. \quad (4)$$

The second digital PFM algorithm that we consider is given in Fig. 3 as Algorithm 2. This algorithm looks forward in time,

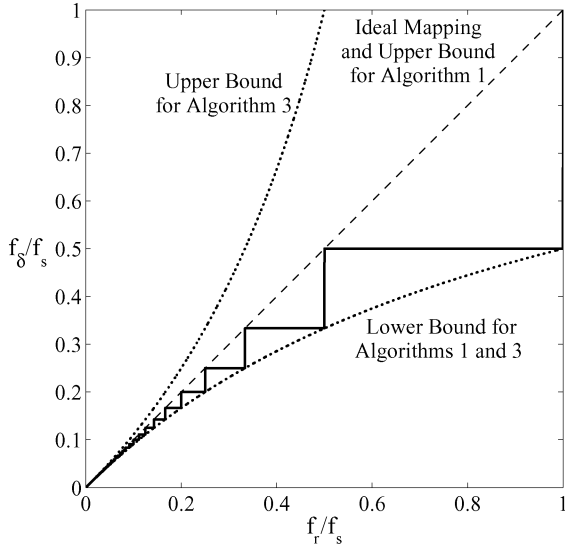


Fig. 4. Input/output mapping for Algorithm 1. Error bounds, which also apply to Algorithm 3, are also shown.

in an attempt to remove the bias (i.e., constant under-representation of f_δ) exhibited by Algorithm 1. The algorithm emits a pulse, even if the integrand has not yet reached the threshold value, if that pulse would tend to minimize the error in T_δ . This works under the assumption that the reference pulse frequency will remain constant over the next sampling period. Fig. 5 shows the output pulse frequency f_δ versus the reference pulse frequency f_r , both normalized by the sampling frequency f_s , for Algorithm 2. The nonlinear nature of the input/output mapping remains, but the bias has been largely removed. The magnitude of the bound on the error in f_δ for over-representation of f_δ (i.e., the upper bound) is

$$\frac{|f_\delta - f_r|}{f_r} = \frac{f_r}{2f_s - f_r}. \quad (5)$$

The magnitude of the upper error bound is always slightly larger than that of the lower error bound. Thus, the error bound (5) can be used as an overall error bound for Algorithm 2. However, note that this error bound becomes very conservative for $f_r/f_s > 2/3$. For reference-frequency values that are very small relative to the sampling frequency ($f_r \ll f_s$), the bound of (5) can be approximated by the simple expression

$$\frac{|f_\delta - f_r|}{f_r} \approx \frac{f_r}{2f_s}. \quad (6)$$

Algorithms 1 and 2 are both deterministic, in the sense that there is a single-valued mapping from the reference frequency f_r to the output pulse frequency f_δ . We now discuss a third digital PFM algorithm that outputs the correct reference frequency in a probabilistic sense. The algorithm we consider is given in Fig. 3 as Algorithm 3. Here, time is numerically integrated until a threshold value is crossed, but unlike the two previous algorithms, the integrand is not reset to zero. The result is an output pulse train that toggles back and forth between two values of T_δ , sometimes overrepresenting and sometimes under-representing the reference value, at a ratio appropriate for the value of T_r .

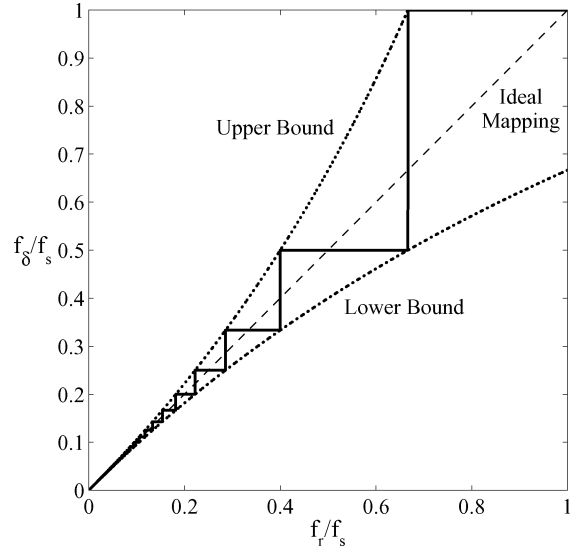


Fig. 5. Input/output mapping for Algorithm 2. Error bounds are also shown.

To quantify this notion of outputting the correct pulse period in a probabilistic sense, let us consider any value of T_r that is rational and greater than the positive rational sampling period T_s . Let n be the smallest positive integer such that $m = nT_s/T_r$ is an integer. Starting with any initial value of the integrand, after n cycles of the microprocessor, the integrand returns to the initial value, and a total of m pulses is emitted during that time. Let p be the positive integer such that $T_r = (p + \xi)T_s$, where $\xi \in [0, 1)$. Let q be the integer number of pulses giving $T_\delta = (p + 1)T_s$ within the n cycles; there are consequently $m - q$ pulses giving $T_\delta = pT_s$ within the n cycles. From the definition of n , we may write the equation

$$q(p + 1)T_s + (m - q)pT_s = nT_s. \quad (7)$$

We are concerned with the ratio q/m , which is the ratio of pulses of duration $T_\delta = (p + 1)T_s$ to the total number of pulses that occur within the n cycles. It can easily be shown that the solution to the above equations is $q/m = \xi$. During a sequence of n cycles, the output pulse train may appear chaotic, but repeats after the n th cycle. The above proof holds true for any value of T_r that is rational, but the underlying system behavior holds true for any value of T_r ; however, an irrational T_r will never repeat ($n \rightarrow \infty$).

Let us solidify this concept with a numerical example. Consider $T_s = 1$ and $T_r = 1.7$. The digital PFM algorithm will output two distinct pulse periods: $T_\delta \in \{1, 2\}$. We have $n = 17$, $m = 10$, and $\xi = 0.7$, which gives $q = 7$. Thus, for every ten pulses that are emitted, seven give $T_\delta = 2$, and three give $T_\delta = 1$. The value of $\xi = 0.7$ corresponds to the fact that $T_r = 1.7$ is 70% of the way between $T_r = 1$ and $T_r = 2$, and results in 70% of the pulses emitted corresponding to $T_r = 2$.

Next, we demonstrate that the underlying behavior is not dependent on T_r or T_s being rational. Fig. 6 shows three runs with $f_r = 0.43f_s$ ($T_r \approx 2.33T_s$), with the initial value of the integrand randomly chosen between 0 and T_s . We see in this figure that the pulse period does take on two distinct values: $T_\delta \in \{2, 3\}$. We cannot compute n or m for this value of f_r , but we know that $\xi \approx 0.33$. From Fig. 6, we find that of the 63

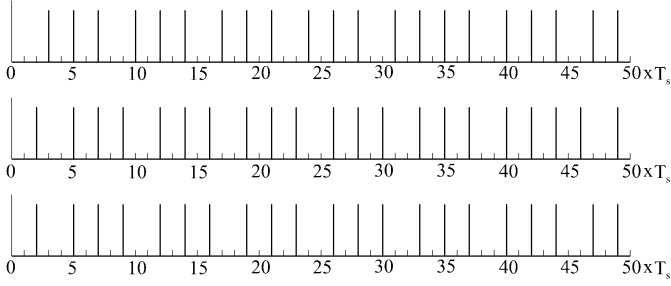


Fig. 6. Three runs of Algorithm 3 for $f_r = 0.43 f_s$, with the initial value of the integrand randomly chosen between 0 and T_s .

pulses emitted, 21 resulted in $T_\delta = 3$ (that is, one in three). This is expected from our value of ξ .

It is possible to bound the instantaneous error resulting from this algorithm, but unlike with Algorithms 1 and 2, with this method it is possible to instantaneously over- and under-represent nearly every value of f_r . The instantaneous error bounds are shown in Fig. 4. The lower bound from Algorithm 1, given by (3), applies as the lower bound here as well. However, we now have an addition upper bound

$$\frac{|f_\delta - f_r|}{f_r} = \frac{f_r}{f_s - f_r}. \quad (8)$$

The magnitude of this upper bound is always larger than the magnitude of the lower bound, so it can be used as the overall error bound for this algorithm. Note that this error bound becomes very conservative for $f_r/f_s > 1/2$. The simple bound (4) can be used to approximate (8) when $f_r \ll f_s$.

For the two deterministic algorithms, it is simple to quantify the output pulse frequency on average, since it is constant. However, for this third algorithm, it is not obvious how best to describe the “average” output pulse frequency. This is ultimately a function of a chosen pulse frequency demodulation (PFD) method; various PFD methods are discussed in [17]. A simple PFD method is a windowing method, where one simply counts the pulses that occur inside a fixed time window, and then divides the window time by the number of pulses, resulting in an “average” pulse period. If we again consider the set of rational reference periods T_r discussed above, and make use of the definition of m , we compute the time-average output pulse period for the this algorithm as

$$T_{\delta,av} = \frac{nT_s}{m} = T_r. \quad (9)$$

In other words, for a rational T_r , for each window of width nT_s there occurs m pulses, which gives an average output pulse period of T_r , and consequently, an output pulse frequency of f_r . As before, this underlying system behavior holds true even when T_r is not rational (consider the limit as the PFD window expands to infinity).

IV. DISCUSSION

A comparison of the error bounds obtained with the three digital PFM algorithms is given as Fig. 7. The data is shown at two different scales. For $f_r/f_s < 0.5$, the error bound from Algorithm 2 is smaller than that from Algorithm 1, and the

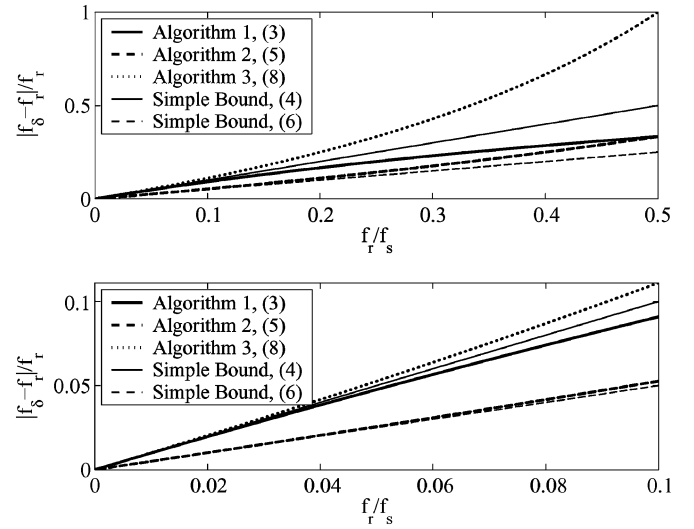


Fig. 7. Comparison of error bounds obtained using three digital PFM algorithms, shown at two scales.

error bound from Algorithm 3 is larger than that of the first. For $f_r/f_s < 0.1$, the true error bounds are closely approximated by the simple error bounds. For these relatively low reference frequencies, the error bounds resulting from Algorithm 1 and Algorithm 3 converge, and are roughly twice as large as those from Algorithm 2. Each of the algorithms presented are essentially equivalent in terms of computation cost. Algorithm 2 is always desirable over Algorithm 1.

It is interesting to note that Algorithm 3 provides the correct frequency when averaged over time, but simultaneously generates the largest instantaneous errors. The nervous system tends to demodulate PFM signals using stochastic methods that tend to average many spike trains, making information transfer robust to errors in any single spike train. A collection of spike trains like those in Fig. 6, where each channel has a unique integrator initialization, would have an ensemble average that has the desired pulse frequency. Consequently, Algorithm 3 may prove to have desirable properties for afferent sensory feedback. Multiple digital PFM channels using Algorithm 2 would have an ensemble average that retains the error seen in the individual channels. Ultimately, only human experiments will reveal which of Algorithms 2 and 3 is the most desirable for interacting with the human nervous system. The number of available feedback channels for a given sensation will likely factor in as well.

There are two other physiological aspects that indicate that digital emulation of PFM may be particularly useful for afferent sensory feedback. First, in recent experiments it was found that human subjects can discriminate neural feedback pulse frequencies up to 500 Hz [12]. This value roughly corresponds to the absolute physiological upper bound on neural spike frequencies imposed by the duration of individual spikes and the absolute refractory period that follows each spike. Second, there is evidence of a logarithmic relationship between afferent pulse frequency and perceived sensation [10]. This indicates a decreasing sensitivity to errors in pulse frequency as the pulse frequency increases. Both of these properties tend to mitigate the impact of errors incurred with digital PFM—mitigate, but not alleviate.

The upper bounds on possible reference frequencies can be used in conjunction with the analysis presented here in the design of prosthetic control systems. For example, if it was determined that a 5% error in output pulse frequency was acceptable, a microprocessor implementing Algorithm 2 must be capable of running at a minimum of approximately 5000 Hz (assuming a 500-Hz upper limit on f_r). Implementing this sampling rate in a clinical arm is nontrivial, due to computation and A/D time. Our current system runs at 2000 Hz—this rate has errors of up to 14.3% using Algorithm 2.

This paper is not a presentation of what pulse frequencies should be used to generate a given sensation. The analysis presented here applies regardless of how the desired pulse frequency is obtained—whether the mapping from pulse frequency to perceived sensation is linear or logarithmic, and whether an absence of sensation corresponds to a zero or non-zero baseline pulse frequency. Another aspect of digital PFM not explored in this paper is time-varying reference frequencies, which may become important once digital PFM is included as part of a closed-loop feedback system. These feedback systems are known to demonstrate undesirable limit cycles [17]. For relatively low-bandwidth systems, such as the human body, consideration of resolution errors of digital PFM to constant reference frequencies should provide a good approximation for closed-loop system behavior. The fact that the nervous system naturally makes use of PFM indicates that it should also be sufficient for neuroprostheses. Human trials with the neuroprosthesis at the University of Utah are ongoing, and the sensitivity of a closed-loop system to the errors discussed will be explored experimentally in the future.

V. CONCLUSION

We have presented three simple algorithms for implementing PFM directly with a microprocessor with its own constant sampling frequency, reducing the complexity of the analog electronics necessary for afferent sensory feedback. Two methods give a deterministic mapping from the desired pulse frequency to actual pulse frequency. We quantified the errors incurred with each algorithm, and showed that a smart choice of algorithm can reduce errors incurred by a factor of two. A third algorithm presented results in a pulse frequency that is close to the desired frequency in a probabilistic sense. That is, the actual pulse period is overrepresented and underrepresented at a ratio that is proportional to the desired pulse period, resulting in an optimal time-average output frequency. Using knowledge of the errors incurred with digitally implementing PFM directly, as well as the maximum pulse frequency that can be discriminated by the human nervous system and allowable system errors, it is possible to find a lower bound on the necessary microprocessor sampling frequency such that PFM can be implemented directly with a microprocessor.

REFERENCES

- [1] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press, 1997.

- [2] K. W. Horsch and G. S. Dhillon, Eds., *Neuroprosthetics: Theory and Practice*. New York: World Scientific, 2004.
- [3] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, 4th ed. New York: McGraw-Hill, 2000.
- [4] S. I. Helms Tillery, D. M. Taylor, and A. B. Schwartz, "The general utility of a neuroprosthetic device under direct cortical control," in *Proc. IEEE Int. Conf. Eng. Med. Biol. Soc.*, 2003, pp. 2043–2046.
- [5] X. Navarro, T. B. Krueger, N. Lago, S. Micera, T. Stieglitz, and P. Dario, "A critical review of interfaces with the peripheral nervous system for the control of neuroprostheses and hybrid bionic systems," *J. Peripheral Nervous Syst.*, vol. 10, pp. 229–258, 2005.
- [6] D. McDonnell, G. A. Clark, and R. A. Normann, "Interleaved, multi-site electrical stimulation of cat sciatic nerve produces fatigue-resistant, ripple-free motor responses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 12, no. 2, pp. 208–215, Jun. 2004.
- [7] P. T. Watkins, G. Santhanam, K. V. Shenoy, and R. R. Harrison, "Validation of adaptive threshold spike detector for neural recording," in *Proc. IEEE Int. Conf. Eng. Med. Biol. Soc.*, 2004, pp. 4079–4082.
- [8] A. Branner, R. B. Stein, and R. A. Normann, "Selective stimulation of cat sciatic nerve using an array of varying-length microelectrodes," *J. Neurophysiol.*, vol. 85, no. 4, pp. 1585–1594, 2001.
- [9] S. M. Lawrence, G. S. Dhillon, and K. W. Horsch, "Fabrication and characteristics of an implantable, polymer-based, intrafascicular electrode," *J. Neurosci. Methods*, vol. 131, no. 1–2, pp. 9–26, 2003.
- [10] G. S. Dhillon, T. B. Krüger, J. S. Sandhu, and K. W. Horsch, "Effects of short-term training on sensory and motor function in severed nerves of long-term human amputees," *J. Neurophysiol.*, vol. 93, pp. 2625–2633, 2005.
- [11] J. Ohta, K. Kagawa, T. Tokuda, and M. Nunoshita, "Retinal prosthesis device based on pulse-frequency-modulation vision chip," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 2923–2926.
- [12] G. S. Dhillon and K. W. Horsch, "Direct neural sensory feedback and control of a prosthetic arm," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 4, pp. 468–472, Dec. 2005.
- [13] M. B. Colton and S. G. Meek, "An experimental neuroelectric prosthetic arm," in *Proc. IFAC Conf. Mechatronic Syst.*, 2002, pp. 336–341.
- [14] C. C. Li and R. W. Jones, "Integral pulse frequency modulated control systems," in *Proc. 2nd Congress IFAC*, 1963, pp. 186–195.
- [15] T. Pavlidis and E. I. Jury, "Analysis of a new class of pulse-frequency modulated feedback systems," *IEEE Trans. Autom. Control*, vol. AC-10, pp. 35–42, 1965.
- [16] S. Tzafestas and G. Frangakis, "Design and implementation of pulse frequency modulation control systems," *Trans. Inst. Measurement Control*, vol. 2, no. 2, pp. 65–78, 1980.
- [17] J. J. Abbott, "Design tools for pulse-frequency-modulated control systems: Error analysis and limit-cycle prediction," Master's thesis, Dept. Mechan. Eng., Univ. Utah, Salt Lake City, 2001.
- [18] D. T. Horn, *Oscillators Simplified, With 61 Projects*. Blue Ridge Summit, PA: Tab Books, 1987.
- [19] P. Horowitz and W. Hill, *The Art of Electronics*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1989.



Jake J. Abbott (M'05) received the M.S. degree from the University of Utah, Salt Lake City, in 2001 and the Ph.D. degree from The Johns Hopkins University, Baltimore, MD, in 2005, both in mechanical engineering.

He is now a Postdoctoral Research Associate with the Institute of Robotics and Intelligent Systems at ETH Zurich, Switzerland. He is interested in the design and control of novel robotic systems.



Sanford G. Meek received the B.S. degree from Purdue University, West Lafayette, IN and the Ph.D. degree from the University of Utah, Salt Lake City.

He is currently an Assistant Professor in Mechanical Engineering at the University of Utah. His research interests are the control and signal processing for neuro-prosthetic arms, as well as biologically inspired compliance for gait stability in quadruped robots.