Antialiasing Encoder Interface With Sub-Nyquist Sampling

Zachary D. Buckner, *Member, IEEE*, Michael L. Reed, *Senior Member, IEEE*, and James H. Aylor, *Fellow, IEEE*

Abstract—A technique is presented for reconstructing absolute position or angle from time-sampled outputs of incremental encoders. Unlike existing techniques, the new method is able to track analog quadrature outputs with frequency contents higher than the Nyquist frequency. The algorithm is described in detail, and both the existing and new methods are tested using a linear position sensor and a high-speed hydraulic actuator. The results from these tests demonstrate aliasing and incorrect tracking for the existing algorithm and correct tracking for the new algorithm. The technique is an excellent complement to resolution-enhancing encoder interfaces that sample raw analog encoder outputs.

Index Terms—Encoder, Nyquist, quadrature, reconstruction, sampling.

I. INTRODUCTION

ENCODERS translate rotary or linear motion into time-varying electrical signals, often of the form

$$s_1 = \text{amp}_1 \cdot \sin(\theta + \text{off}_1)$$
$$s_2 = \text{amp}_2 \cdot \cos(\theta + \text{off}_2) \tag{1}$$

where θ represents the angle with respect to a single tooth period or band period on the encoder target. For linear encoders,[1] θ can be expressed as

$$\theta = 2\pi \cdot \text{position/pitch} \tag{2}$$

where pitch is the distance that corresponds to a complete sinusoidal period in the output signal. For rotary encoders, θ can be expressed as

$$\theta = \theta_{\text{shaft}} \cdot n \tag{3}$$

where n is the number of encoder bands per complete shaft revolution.

Since the output signals s_1 and s_2 are periodic functions of the measurement variable, additional processing stages are necessary to translate the signals into an absolute phase or absolute position for use by control and measurement systems.

Many encoder interfaces perform this translation by passing each signal through a zero-crossing detector circuit [1]–[3]. The resultant pulse trains are then used to drive a digital counter circuit that accumulates state transitions. Many variations and improvements to this technique have been suggested [1], [2]. However, since only four discernable state transitions occur during each pitch distance, encoder resolution is limited to pitch/4.

To improve resolution, several techniques have been proposed [1], [4]–[8] for directly sampling the analog quadrature signals in (1) and interpolating to resolve finer phase values. These techniques rely on sample-and-hold circuits, analog-to-digital converters, and lookup tables or trigonometric functions to return a single measured phase value at each time step, i.e.,

$$\theta_m[t] = \theta \bmod 2\pi. \tag{4}$$

The second stage of the encoder interface typically tracks the instantaneous modulo phase angle in (4) to produce an absolute phase (2).

One approach for this second stage is to compare the difference between modulo phase values θ_m on successive time steps [5]. Thus, at each time step, the following difference equations are evaluated:

$$\theta[t] = \theta[t-1] + \omega[t] \tag{5}$$
$$\omega[t] = \angle\theta_m[t], \theta_m[t-1] \tag{6}$$

where $\angle$ is the "apparent motion" angular difference, which yields the smallest magnitude phase angle (positive or negative) that could account for the phase change from a_1 to a_2, i.e.,

$$\angle a_2, a_1 = a_2 - a_1 + k \cdot 2\pi$$
$$k = \arg\min_{\text{integer } k} |a_2 - a_1 + k \cdot 2\pi| \tag{7}$$

where argmin returns the value of k that minimizes the given expression. In closed form, the $\angle$ operator can be written as

$$\angle a_2, a_1 = \begin{cases} a_2 - a_1 - 2\pi, & \text{if } a_2 - a_1 > \pi \\ a_2 - a_1, & \text{if } -\pi \leq a_2 - a_1 \leq \pi \\ a_2 - a_1 + 2\pi, & \text{if } a_2 - a_1 < -\pi. \end{cases} \tag{8}$$

$\theta[t]$ and $w[t]$ in (5) and (6) relate to encoder position and velocity as follows:

$$\text{position}[t] = \theta[t] \times \frac{\text{pitch}}{2\pi} \tag{9}$$

$$\text{velocity}[t] = w[t] \times \frac{\text{pitch}}{\Delta t \cdot 2\pi}. \tag{10}$$

Z. D. Buckner is with Elder Research, Inc., Charlottesville, VA 29901 USA (e-mail: zach@datamininglab.com).

M. L. Reed and J. H. Aylor are with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904 USA (e-mail: reed@virginia.edu; jha@virginia.edu).

[1]Notation for linear encoders has been adopted for most equations. However, rotary encoders can be considered by simply replacing the linear measures of "pitch" and "position" with their angular counterparts.

This technique will correctly track absolute phase at sufficiently low angular velocities. However, since the angular difference operator can only produce values inside the range $(-\pi, \pi)$, $w[t]$ is constrained to

$$-\pi < w[t] < \pi. \tag{11}$$

By substituting (6), this becomes

$$|\text{velocity}[t]| < \frac{\text{pitch}}{2 \cdot \Delta t}. \tag{12}$$

This inequality identifies the well-known Nyquist sampling frequency. Thus, encoded velocity is limited by the Nyquist frequency of the sampled signal [5], [7]. If velocity exceeds this bound, aliasing will occur, causing erroneous results.

To overcome this drawback, several techniques have been proposed to augment the resolution enhancement method with counting circuitry [5], [6], [8]. However, such solutions introduce opportunities for error, as the counter and interpolation circuitry can become out of synchronization near "pitch" boundaries (i.e., the counting circuit may advance to the next count before the interpolation circuit does), yielding unacceptably large error.

II. ANTIALIASING TECHNIQUE

A new technique is presented for tracking absolute phase angle at higher encoder velocities without aliasing, using only the time-sampled analog values in (1). The following difference equations describe the approach:

$$\theta[t] = \theta[t-1] + w[t] \tag{13}$$

$$w[t] = w[t-1] + a[t] \tag{14}$$

$$a[t] = \angle (\theta_m[t] - \theta_{\text{predicted}}[t]) \tag{15}$$

$$\theta_{\text{predicted}}[t] = (\theta_m[t-1] + w[t-1]) \bmod 2\pi \tag{16}$$

subject to the initial conditions

$$\theta[0] = w[0] = a[0] = 0 \tag{17}$$

where $\theta_m[t]$ is the measured phase input to the system, as described in (4), and $\theta[t]$, $w[t]$, and $a[t]$ are calculated quantities that relate to encoder position, velocity, and acceleration as follows:

$$\text{position}[t] = \theta[t] \times \frac{\text{pitch}}{2\pi} \tag{18}$$

$$\text{velocity}[t] = w[t] \times \frac{\text{pitch}}{\Delta t \cdot 2\pi} \tag{19}$$

$$\text{acceleration}[t] = a[t] \times \frac{\text{pitch}}{(\Delta t)^2 \cdot 2\pi}. \tag{20}$$

For each iteration, $\theta[t-1]$, $\theta_m[t-1]$, and $w[t-1]$ are known, as they are the stored results from the previous iteration, and $\theta_m[t]$ is measured. The improved algorithm uses the angular difference between the predicted phase, labeled $\theta_{\text{predicted}}$, and the measured phase, labeled $\theta_m[t]$, to determine $a[t]$, a step factor due to acceleration. The difference represents the change in position that is in excess of the predicted change, where

the predicted change is based on the velocity from the previous iteration. In other words, the change in position for the current iteration is equal to the change in position for the previous iteration plus $a[t]$, a measurable step factor due to acceleration.

The same rationale that limits maximum velocity in the traditional encoder interfacing scheme can be applied to the new technique. However, under the new technique, the limited range of the angular difference function limits only $a[t]$, i.e.,

$$-\pi < a[t] < \pi. \tag{21}$$

By applying this constraint to (20), it becomes

$$|\text{acceleration}[t]| < \frac{\text{pitch}}{2 \cdot (\Delta t)^2}. \tag{22}$$

Thus, the original velocity constraint has been replaced with an acceleration constraint. In many applications, this acceleration limit may exceed any acceleration that the system could undergo.

III. GENERALIZED ALGORITHM WITH EXTENSION TO HIGHER ORDER DERIVATIVES

The technique outlined in Section II can be extended to an arbitrarily high degree, replacing the acceleration constraint in (22) with a higher order derivative constraint. The generalized algorithm for this extension is outlined as follows:

Step 1) Set initial conditions for each derivative to zero (or any other value) as follows:

$$\theta^{[0]}[0] = 0$$
$$\theta^{[1]}[0] = 0$$
$$\cdots$$
$$\theta^{[n]}[0] = 0. \tag{23}$$

Step 2) For each time step $t > 0$, assume that the highest order derivative is 0. Then predict the resultant phase by evaluating all n of the following difference equations in sequence:

$$\theta_p^{[n]}[t] = 0 \tag{24}$$

$$\theta_p^{[i]}[t] = \theta^{[i]}[t-1] + \theta_p^{[i+1]}[t], \qquad i = n-1 \ldots 0. \tag{25}$$

Step 3) Using the predicted phase value computed above, set the highest order derivative as

$$\theta^{[n]}[t] = \angle \left(\left(\theta_p^{[0]}[t] \bmod 2\pi \right), \theta_m[t] \right). \tag{26}$$

Step 4) Recompute the series of differential equations as follows:

$$\theta^{[i]}[t] = \theta^{[i]}[t-1] + \theta^{[i+1]}[t], \qquad i = n-1 \ldots 0. \tag{27}$$

This method tracks correctly as long as the following constraint is met:

$$\left| \theta^{[n]}[t] \right| < \frac{\text{pitch}}{2 \cdot (\Delta t)^n}. \tag{28}$$

TABLE I
ENCODER INTERFACE PROPERTIES

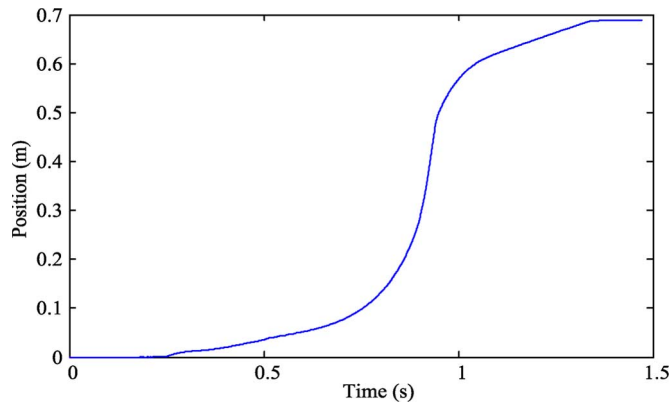|	
Encoder Pitch	1.27 mm
Interface Sampling Period	0.980 ms
Interface Sampling Frequency	1.02 kHz
Max Velocity Before Aliasing (n=1) Algorithm from (12)	0.648 m/s
Max Acceleration Before Aliasing (n=2) Algorithm from (22)	661 m/s²
Max Jerk Before Aliasing (n=3) Algorithm from	6.74 ×10⁵ m/s³
Max $\theta^{[4]}[t]$ Before Aliasing (n=4) Algorithm	6.88 ×10⁸ m/s⁴

Fig. 1.   Profile 1. The prescribed position profile for the linear actuator.

Note that the traditional technique for tracking absolute phase shown in (5) and (6) can be described using the above algorithm, where $n = 1$. The improved algorithm described in Section II can be described using the above algorithm, where $n = 2$. The algorithm is referenced in Section IV, where $n = 3$ and $n = 4$.

IV. EXPERIMENTS

A. Encoder Parameters and Properties

The generalized algorithm described in Section III was tested using a linear encoder manufactured by Visi-Trak Worldwide. This sensor has standard analog quadrature outputs, which were resolved to a phase angle using an arctangent method [1]. Absolute phase angle $\theta^{[0]}[t]$ was digitally output at each time step and recorded on a PC using a high-speed digital input board made by National Instruments. The generalized algorithm was implemented in Matlab and used offline to process the phase angle signals. The parameters and properties for this encoder and the interface hardware are summarized in Table I.

B. Test Configuration

The sensor was mounted on a hydraulic linear actuator with a Visi-Trak closed-loop controller. The controller was programmed to follow position profiles similar to those used in die casting applications. Two test profiles were used, i.e., Profile 1 (Fig. 1; Table II) and Profile 2 (Fig. 2; Table III).

TABLE II
PROFILE 1 PROPERTIES

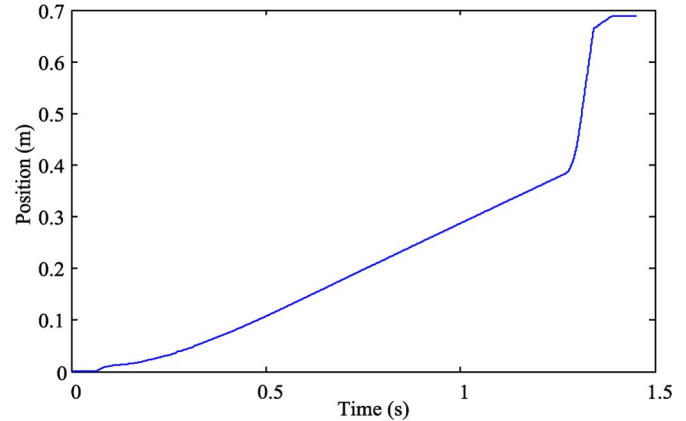|	
Max Velocity	5.99 m/s
Max Acceleration	376 m/s²
Max Jerk	6.47 10⁵ m/s³
Max $\theta^{[4]}[t]$	1.07 10⁹ m/s⁴

Fig. 2.   Profile 2. With lower peak velocity than profile 1 but higher peak acceleration and jerk. Note that the maximum acceleration is marginally faster than the $(n = 2)$ algorithm can handle.

TABLE III
PROFILE 2 PROPERTIES

Max Velocity	3.34 m/s
Max Acceleration	665 m/s²
Max Jerk	5.40 10⁵ m/s³
Max $\theta^{[4]}[t]$	9.28 10⁸ m/s⁴

The interface algorithm was tested in $n = 1$, $n = 2$, $n = 3$, and $n = 4$ configurations, representing the traditional tracking algorithm in (5) and (6), the improved tracking algorithm in (13)–(17), and two higher order configurations. The algorithm was deemed to have correctly tracked the profile by visually inspecting the measured profile's shape and checking to assure that the ending position matched the known ending position for the linear actuator, i.e., 0.690 m.

C. Results for Profile 1

Because of the high velocity developed during the regions between 0.5 and 1 s of Profile 1, the traditional absolute phase tracking algorithm $(n = 1)$ yielded aliasing (Fig. 3). This aliasing can be observed as sharp discontinuities at the 0.65 m/s velocity boundary, in accord with the constraint in (12) and calculated in Table I.

However, the same stream of sampled phase values was correctly tracked using the improved algorithm $(n = 2)$, yielding the velocity profile shown in Fig. 4. This algorithm's constraint was satisfied, as total acceleration, i.e., 376 m/s², was less than the maximum acceleration acceptable for the encoder interface, i.e., 661 m/s².

The aliasing introduced with the $(n = 1)$ algorithm is more clearly visible in the periodic banding in Fig. 5, a scatter plot showing velocities tracked with the $(n = 1)$ algorithm versus
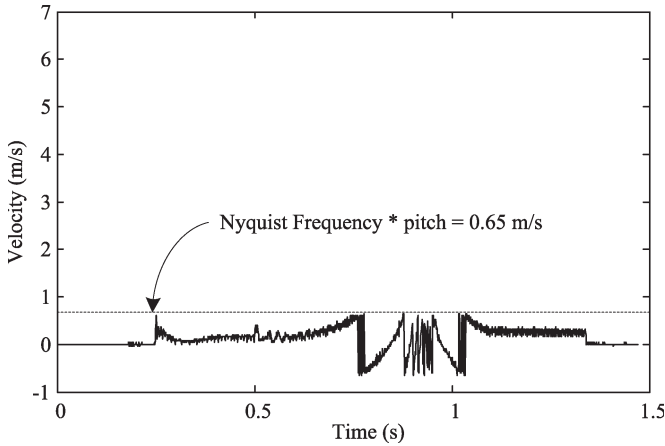
Fig. 3. Velocity $\theta^{[1]}[t]$ tracked using the $(n = 1)$ algorithm. Aliasing is evident for velocities exceeding 0.65 m/s.
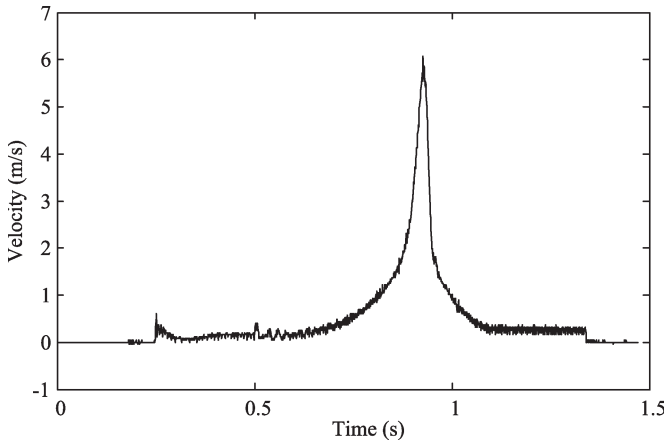

Fig. 4. Velocity $\theta^{[1]}[t]$ tracked using the $(n = 2)$ algorithm. No aliasing occurred, as maximum acceleration for the algorithm was not exceeded.
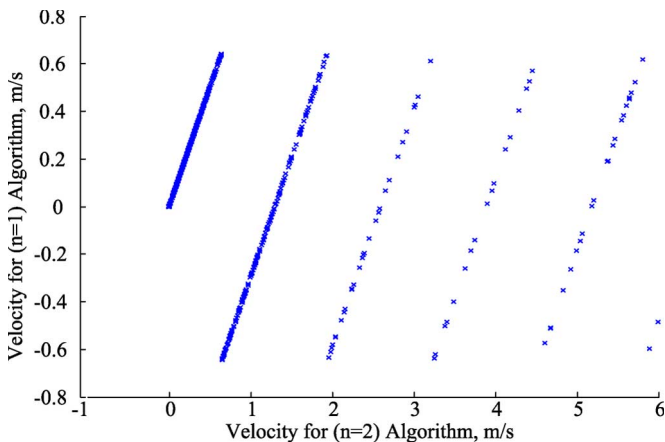

Fig. 5. Scatter plot showing the aliasing of high frequencies in the $(n = 1)$ algorithm.

the recorded velocities computed at the same instants with the $(n = 2)$ algorithm (Fig. 6).

The $(n = 3)$ algorithm also tracked correctly, as total jerk for the profile was less than the maximum acceptable jerk. The $(n = 4)$ algorithm did not track correctly, as $\theta^{[4]}[t]$ exceeded the maximum allowable for the $(n = 4)$ algorithm.
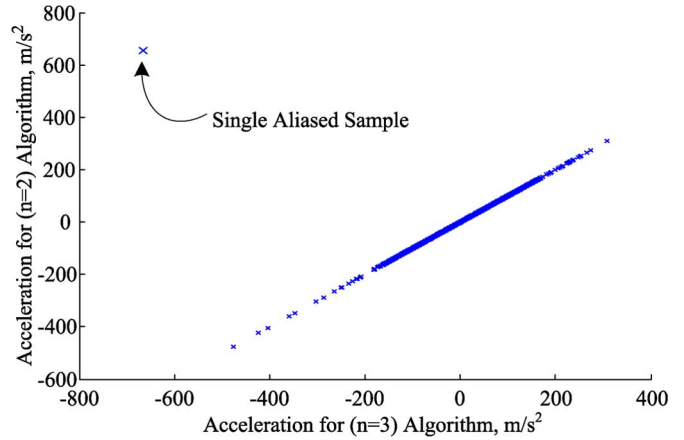

Fig. 6. Scatter plot showing the aliasing of a single sample in the $(n = 2)$ algorithm.

TABLE IV
TRACKING RESULTS

	Tracked Correctly	
	Profile 1	Profile 2
n=1	no	no
n=2	yes	no
n=3	yes	yes
n=4	no	no

D. Results for Profile 2

Profile 2 underwent velocities that were too high for the $(n = 1)$ algorithm, and a single sample was recorded that exceeded the acceleration limit for the $(n = 2)$ algorithm. This sample occurred around $t = 1.4$ s, during the period of rapid deceleration at the end of the profile. However, the $(n = 3)$ algorithm tracked correctly, as maximum jerk experienced in Profile 2 was less than the calculated maximum. The $(n = 4)$ algorithm produced erroneous results, as $\theta^{[4]}[t]$ exceeded the maximum allowable for the $(n = 4)$ algorithm.

E. Summary of Results

A complete summary of all tracking results is shown in Table IV.

V. CONCLUSION

The generalized algorithm presented in Section III is a useful improvement over existing techniques. This algorithm can be used in conjunction with a wide variety of resolution-enhancing encoder interface methods [1], [4]–[7], yielding finer resolution measurements at high speeds.

However, selection of the parameter n must be done with careful consideration to operating constraints. A general technique is to select the lowest n such that the constraint in (28) is satisfied for all measured profiles. As highlighted in Table IV, this optimum value was $(n = 3)$ for the test profiles. All $(n > 3)$ produced erroneous results, as the absolute magnitude of higher order derivatives tended to increase with the derivative order. This tended to prevent the constraint in (28) from being satisfied.

REFERENCES

[1] J. R. R Mayer, "Optical encoder displacement sensors," in *The Measurement, Instrumentation, and Sensor Handbook*, J. Webster, Ed. Boca Raton, FL: CRC, 1999, pp. 6.98–6.118.

[2] R. Pallás-Areny and J. G. Webster, *Sensors and Signal Conditioning*, 2nd ed. New York: Wiley, 2001, pp. 433–441.

[3] R. Frank, *Understanding Smart Sensors.*. Norwood, MA: Artech House, 2000, pp. 63–64.

[4] P. L. M. Heydemann, "Determination and correction of quadrature fringe measurement errors in interferometers," *Appl. Opt.*, vol. 20, no. 19, pp. 3382–3384, Oct. 1981.

[5] N. Hagiwara and H. Murase, "A method of improving the resolution and accuracy of rotary encoders using a code compensation technique," *IEEE Trans. Instrum. Meas.*, vol. 41, no. 1, pp. 98–101, Feb. 1992.

[6] J. R. R Mayer, "High resolution of rotary encoder analog quadrature signals," *IEEE Trans. Instrum. Meas.*, vol. 43, no. 3, pp. 494–498, Jun. 1994.

[7] K. K. Tan, H. X. Zhou, and T. H. Lee, "New interpolation method for quadrature encoder signals," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 5, pp. 1073–1079, Oct. 2002.

[8] Motion Engineering, *SIM4 Scale Interpolation Module*, Apr. 2000. Application Notes 206, Rev. C. [Online]. Available: http://www.motioneng.com/pdf/scale_intrp_c.pdf.

Michael L. Reed (S'79–M'80–SM'99) received the B.S. and M.Eng. degrees from Rensselaer Polytechnic Institute, Troy, NY, and the Ph.D. degree from Stanford University, Stanford, CA, all in electrical engineering.

He is a Professor of electrical engineering and biomedical engineering with the University of Virginia, Charlottesville. He has held appointments at Hewlett-Packard Laboratories, Carnegie-Mellon University, University of Twente, ETH Zürich, and the Albert Ludwigs Universität, Freiburg, Germany. His research interests center around micro- and nanofabrication technologies and microsystems applications.

Dr. Reed is a Fellow of the Institute of Physics. He was the Technical Chairman of the 1995 IEEE International Workshop on MEMS and the General Chairman of the 1996 Workshop. He also organized the 1996 Materials Research Society Symposium on Materials in Microsystems. He is an Editor of the journal *Sensors and Materials*. He currently holds 12 issued and seven pending patents related to microsystems technology and microfabricated medical devices. With R. Rohrer, he is the author of the textbook *Applied Introductory Circuit Analysis for Electrical and Computer Engineers* (Prentice-Hall, 1998) as well as numerous research publications. He is a recipient of the Hertz Foundation Prize and a Presidential Young Investigator Award.

James H. Aylor (S'66–M'68–SM'82–F'96) received the B.S., M.S., and Ph.D. degrees from the University of Virginia, Charlottesville, in 1968, 1971, and 1977, respectively, all in electrical engineering.

He is the Louis T. Rader Professor of Electrical and Computer Engineering and the Dean of the School of Engineering and Applied Science with the University of Virginia. Before taking the Dean position, he was an Interim Dean for one year, a Senior Associate Dean of Academic Programs for one year, and the Chair of the Department of Electrical and Computer Engineering for eight years. He has served as Principal Investigator or Co-Principal Investigator on more than 35 contracts. He has authored or coauthored one book, four book chapters, and more than 150 technical articles. He has been involved in all aspects of the digital design process for more than 25 years. He has directed work in the areas of system-level modeling, concurrent error detection, automatic test pattern generation, hardware description languages, very large scale integration system design, and computer technology for aged independence.

Dr. Aylor has been extremely active in professional activities both in technical and administrative capacities. He served as the President of the Electrical and Computer Engineering Department Heads Association during 2001–2002. He has also served in many administrative roles within the Institute of Electrical and Electronic Engineers and the IEEE Computer Society, including serving as the 1993 Computer Society President and Editor-in-Chief of *IEEE Computer*. From 1994 to 1996, he served as an IEEE Division Director and a member of the IEEE Board of Directors. He also serves as an ABET Visitor for electrical engineering, computer engineering, and computer science programs and is an International Evaluator.

Zachary D. Buckner (M'06) received the M.S. degree in electrical engineering from the University of Virginia, Charlottesville, in 2004.

He is currently a Consultant with Elder Research, Inc., Charlottesville, where he manages the development of several analytic software packages. In addition, he continues to lead sensor research for Visi-Trak Worldwide, Valley View, OH. He is the holder of two patents related to encoder interfaces.