

# Variable-Rate Data Sampling for Low-Power Microsystems Using Modified Adams Methods

Geoff B. Irvine, Lei Wang, Peter Dickman, *Member, IEEE*, and David R. S. Cumming, *Member, IEEE*

**Abstract**—A method for variable-rate data sampling is proposed for the purpose of low-power data acquisition in a small footprint microsystem. The procedure enables energy saving by utilizing dynamic power management techniques and is based on the Adams–Bashforth and Adams–Moulton multistep predictor-corrector methods for ordinary differential equations. Newton–Gregory backward difference interpolation formulae and past value substitution are used to facilitate sample rate changes. It is necessary to store only  $2m + 1$  equispaced past values of  $t$  and the corresponding values of  $y$ , where  $y = g(t)$ , and  $m$  is the number of steps in the Adams methods. For the purposes of demonstrating the technique, fourth-order methods are used, but it is possible to use higher orders to improve accuracy if required.

**Index Terms**—Adams methods, dynamic power management, interpolation, low-power, variable-rate sampling.

## I. INTRODUCTION

LOW-power techniques have become increasingly important in recent years due to the burgeoning number of consumer, military, and medical remote sensing devices requiring long battery life combined with good performance. Some of these techniques employ dynamic power management (DPM) methods by, for instance, altering supply voltages or reducing clock speeds [1], [2].

In the simplest microsystems with digital signal processing (DSP) capability, the user will configure the device to operate at a fixed sample rate appropriate to the maximum frequency of interest expected in the data. An example of fixed-rate sampling, where  $t_n$  represents the time of the  $n$ th sample and  $y_n = g(t_n)$  the corresponding value, is shown in Fig. 1. Here, the sample rate is  $1/h$ , where  $h$  is the fixed sampling interval or step size.

While fixed-rate sampling methods enable simplicity of design, it is observed that in many applications, such as some biomedical and environment monitoring systems, the frequency is often low for extended periods [3], [4]. During such periods, when  $y$  remains constant or varies slowly, many readings are taken having the same or similar value, producing a great deal of redundant information. Reducing the sampling rate would solve

Manuscript received January 9, 2002; revised March 28, 2003. This work was supported by the Scottish Higher Education Funding Council RDG 130 and by the U.K. Engineering and Physical Sciences Research Council. The associate editor coordinating the review of this paper and approving it for publication was Prof. Abdelhak M. Zoubir.

G. B. Irvine, L. Wang, and D. R. S. Cumming are with the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, U.K. (e-mail g.irvine@elec.gla.ac.uk, l.wang@elec.gla.ac.uk, d.cumming@elec.gla.ac.uk).

P. Dickman is with the Department of Computing Science, University of Glasgow, Glasgow G12 8RZ, U.K. (e-mail pd@dcs.gla.ac.uk).

Digital Object Identifier 10.1109/TSP.2003.819004

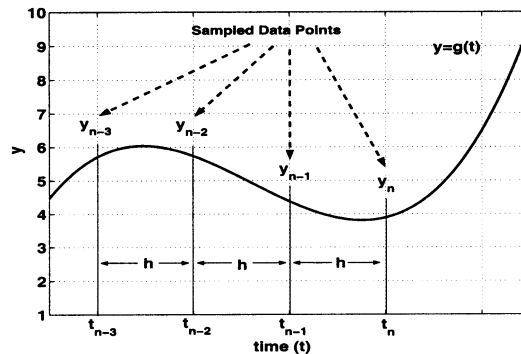


Fig. 1. Fixed-rate sampling where all readings are equispaced with respect to  $t$ , and the sampling rate is  $1/h$ .

this problem, but, if  $y$  begins to change rapidly, readings of interest may be missed. Variable-rate sampling (VRS) attempts to overcome these problems by altering  $h$  depending on the rate of change of values being observed in the data at a given time.

In this paper, we demonstrate a VRS controller for a battery-operated, limited-bandwidth sensor microsystem with wireless transmission to a base station. In the systems we are studying, the maximum sampling frequency is typically  $< 1$  Hz, and the controller takes advantage of instances when  $h$  is large to optimize the use of DPM. The data received at the base station may be time stamped and rebuilt using conventional techniques, and, because the receiver need not be a low-power device, powerful computing capability may be employed. Data transmission delay is not considered significant in these systems as it is negligible compared with the average sampling interval.

## II. BACKGROUND

Some VRS controllers alter  $h$  depending on the rate of change of  $y$ . One such method is the level-crossing technique [5], [6], where the time taken for  $y$  to reach fixed thresholds is measured. The alternative method we demonstrate here relies on the ability to predict, within a user-determined accuracy, the next value in a series of samples.

If, at time  $t_n$ , the value  $y_{n+1}^p$  is predicted and subsequently, at time  $t_{n+1}$ , compared with the true value  $y_{n+1}$ , the prediction error is a function of  $h$  and  $f(t, y)$ . This follows since the prediction is of the form

$$y_{n+1}^p = y_n + hf(t, y) \quad (1)$$

where  $y_{n+1}^p$  is the predicted value of the next reading, and  $f(t, y)$  is an estimated slope between  $(t_n, y_n)$  and  $(t_{n+1}, y_{n+1})$ , as shown in Fig. 2.

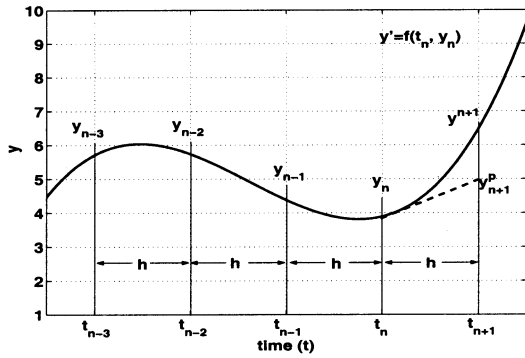


Fig. 2.  $y_{n+1}^p$  predicted by multiplying  $h$  by the estimated slope between  $y_n$  and  $y_{n+1}$ .

Since this is the case, the prediction is an example of an initial value ordinary differential equation (ODE), and, depending on the error in the predicted value, the step size may be altered to attempt to maintain prediction accuracy within a given tolerance ( $tol$ ). Various numerical methods exist for the calculation of solutions of ODEs, and these methods may be separated into two groups: single step and multistep.

#### A. Single Step ODE Methods

To calculate  $f(t, y)$ , single-step ODE solvers use first derivative information either at point  $(t_n, y_n)$ , as with the explicit Euler method [7], or at multiple points within a single step, e.g., in the range  $t_n$  to  $t_{n+1}$ , as is the case with Runge–Kutta (RK) methods [8], [9].

These methods vary in accuracy. For example, the explicit Euler method is empirically less accurate than the widely used fourth-order Runge–Kutta method (RK4). RK4 uses estimated slopes at  $t_n$ ,  $t_n + (1/2)h$  and  $t_n + h$ , requiring numerous calculations to predict  $y_{n+1}$ , as follows [7]:

$$y_{n+1}^{rk4} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2)$$

where the various slope calculations are

$$k_1 = hf(t_n, y_n) \quad (3a)$$

$$k_2 = hf\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \quad (3b)$$

$$k_3 = hf\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right) \quad (3c)$$

$$k_4 = hf(t_n + h, y_n + k_3) \quad (3d)$$

and (2) is the RK4 method for calculating (1).

In a DSP system,  $f$  is unknown and must be derived from previous points, adding to the computational complexity of the algorithm. In a low-power system, it is desirable to minimize the overhead incurred by the VRS system to save processing time and lower hardware or software requirements. We have, therefore, used multistep methods that require less computation to achieve a prediction of similar accuracy.

#### B. Multistep ODE Methods

Multistep predictor methods utilize past values of  $f_n$  to construct a polynomial approximation of the derivative function of  $y$  and extrapolate the polynomial into the next step so that the

prediction again becomes (1). This provides a more accurate estimate of the slope between  $(t_n, y_n)$  and  $(t_{n+1}, y_{n+1})$  than the simplest single-point methods [7].

Two such methods are the Adams–Bashforth and Adams–Moulton predictor and corrector [10], [11] which provide a simple, computationally efficient technique of similar accuracy to RK methods [7]. They require a number of equispaced previous data values of known time interval, but once these values are known, the algorithm for predicting the next  $y$  value is straightforward.

Only a few  $(2m + 1)$  past values need to be stored, and since past values are used more than once and computation per step is low, these methods lend themselves to low-power data sampling. Polynomial predictors are typically used in FIR filtering systems [12]–[15].

### III. VARIABLE SAMPLE RATE CONTROL

As demonstrated in [16], it is possible to use evenly spaced samples to generate a differential equation that represents a low bandwidth signal. The sampling system presented here employs Adams methods that are modified to use  $y$  ordinates rather than derivative values for predicting  $y_{n+1}$  together with Newton–Gregory interpolation techniques [17] to vary the sample rate.

#### A. Modified Adams Methods

The Adams–Bashforth method can be of order 2 or higher and is typically based on a Newton–Gregory (N–G) backward differencing or Lagrange interpolating polynomial fitted at base point  $t_n$  [18]. In this case, a fourth-order method is used as it is accurate enough for the purposes of this application, and the N–G polynomial integration [17] is shown in the following:

$$y_{n+1} = y_n + \int_0^1 \left( f_n + \frac{s}{1!} f'_n + \frac{s^2 + s}{2!} f''_n + \frac{s^3 + 3s^2 + 2s}{3!} f'''_n \right) h ds. \quad (4)$$

Here,  $s$  represents the fraction of  $h$  for which an extrapolation is being performed.

Using (N–G) backward differencing approximations [17] to calculate the derivatives gives

$$f'_n \approx \frac{f_n - f_{n-1}}{h}, f''_n \approx \frac{f_n - 2f_{n-1} + f_{n-2}}{h^2} \quad (5)$$

etc., and substituting (5) into the integrated form of (4) generates the standard fourth-order Adams–Bashforth (AB4) predictor formula [7]

$$y_{n+1}^{ab} = y_n + h \left( \frac{55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}}{24} \right) \quad (6)$$

where  $y_{n+1}^{ab}$  is the result of the AB4 method of calculating (1).

Adams–Moulton is a corrector method that utilizes the value predicted by AB4, and, with ODEs, improves on this estimate giving a more accurate estimation of  $y_{n+1}$ . Again, it is based on a N–G or Lagrange polynomial, but, since the predicted value of  $y_{n+1}$  is now known, the polynomial is fitted at base point  $t_{n+1}$ , and the derivative approximations are

$$f'_{n+1} \approx \frac{f_{n+1} - f_n}{h}, f''_{n+1} \approx \frac{f_{n+1} - 2f_n + f_{n-1}}{h^2} \quad (7)$$

etc.

Substituting these formulae into the differencing polynomial, we have the standard fourth-order Adams–Moulton (AM4) corrector formula [7]

$$y_{n+1}^{am} = y_n + h \left( \frac{9f_{n+1}^{ab} + 19f_n - 5f_{n-1} + f_{n-2}}{24} \right) \quad (8)$$

where  $y_{n+1}^{am}$  is the result of the AM4 method of calculating (1). From (6)

$$f_{n+1}^{ab} = \frac{y_{n+1}^{ab} - y_n}{h}$$

giving

$$f_{n+1}^{ab} = \frac{55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}}{24}. \quad (9)$$

Substituting (9) into (8) now gives

$$y_{n+1}^{abm} = y_n + h \left( \frac{317f_n - 217f_{n-1} + 119f_{n-2} - 27f_{n-3}}{192} \right) \quad (10)$$

where  $y_{n+1}^{abm}$  is a combined Adams–Bashforth–Moulton method using a single equation rather than the conventional two.

To convert the above calculation to a DSP process where  $f$  is unknown, we use the secant method [17] to replace the derivative values with known  $y$  ordinates as follows:

$$f_n \approx \frac{y_n - y_{n-1}}{h}, f_{n-1} \approx \frac{y_{n-1} - y_{n-2}}{h} \quad (11)$$

etc.

Substituting these values into (10),  $h$  is eliminated to produce a novel, Adams-type formula containing only known  $y$  values such that

$$y_{n+1}^{mam} = \frac{509y_n - 534y_{n-1} + 336y_{n-2} - 146y_{n-3} + 27y_{n-4}}{192} \quad (12)$$

where  $y_{n+1}^{mam}$  is a prediction using the modified combined Adams method.

It is also possible to substitute the secant approximations in (11) into (6), resulting in a method that uses only a predictor, as follows:

$$y_{n+1}^{mab} = \frac{79y_n - 114y_{n-1} + 96y_{n-2} - 46y_{n-3} + 9y_{n-4}}{24}. \quad (13)$$

The modified Adams methods are akin to those in [19] and [20] to solve differential algebraic equations. However, unlike these methods, we calculate  $y_{n+1}^p$  using past actual  $y$  values rather than storing their derivative approximations, these approximations having been incorporated into (12) and (13).

Other similar prediction algorithms can be implemented. For example, it is possible to fit a standard interpolation polynomial to past  $y$  values and extrapolate this to  $y_{n+1}$ . The fourth-order N–G backward differencing polynomial is as follows [17]:

$$y_{n+sh} = y_n + \frac{s}{1!} \nabla^1 y_n + \frac{s^2 + s}{2!} \nabla^2 y_n + \frac{s^3 + 3s^2 + 2s}{3!} \nabla^3 y_n + \frac{s^4 + 4s^3 + 11s^2 + 6s}{4!} \nabla^4 y_n \quad (14)$$

where  $s$  is again the fraction of  $h$  to which the polynomial is being extrapolated, and  $\nabla^1 = y_n - y_{n-1}$ ,  $\nabla^2 = y_n - 2y_{n-1} +$

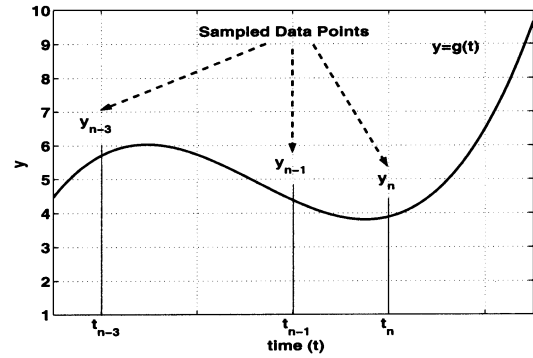


Fig. 3. Variable-rate data sampling where readings are not equispaced with respect to time.

TABLE I  
DOUBLING THE STEP SIZE

Replace	$y_{n-8}$	...	$y_{n-4}$	$y_{n-3}$	$y_{n-2}$	$y_{n-1}$	$y_n$
with			$y_{n-8}$	$y_{n-6}$	$y_{n-4}$	$y_{n-2}$	$y_n$

$y_{n-2}$  etc. from N–G backward differencing. Setting  $s = 1$  yields the following equation:

$$y_{n+1}^{ng} = 5y_n - 10y_{n-1} + 10y_{n-2} - 5y_{n-3} + y_{n-4} \quad (15)$$

where  $y_{n+1}^{ng}$  is the result of extrapolating the N–G interpolation polynomial.

This formula, which is the same as that obtained by extrapolating a Lagrange interpolation polynomial to  $y_{n+1}$  [17] of the same order, is computationally simpler than the modified Adams equations, and the respective merits of the two methods are discussed in Section V.

### B. Varying the Sampling Rate

In a system with a constant sampling rate, the data points are equispaced with respect to time. In a system with a variable sampling rate, this is not the case, as the time between samples is decided by an algorithm. An example of a variable sampling rate is shown in Fig. 3.

Many methods have been examined for step-size alteration for ODEs [21]. In this application, it is desirable that the method employed can be implemented as simply as possible, and we either halve or double the step size, depending on whether the most recent sample was outside or inside a given tolerance  $tol$ , compared with the Adams prediction. That is,  $h$  is doubled if the prediction is sufficiently accurate and halved if the prediction is inaccurate.

1) *Step Size Doubling*: The system requires that nine successive equispaced data values are stored in a first-in-first-out (fifo) structure. A calculation for  $y_{n+1}^p$  is performed using (12). At the next sample time,  $y_{n+1}$  becomes  $y_n$  and so on, and the previous value of  $y_{n-8}$  is discarded from the fifo. If the new value for  $y_n$  is within  $tol/2$  for the predicted value, the sample step  $h$  is doubled.

To ensure that the fifo can be refilled in the minimum number of samples, every second value is discarded and the next value down moved up. This is illustrated in Table I. Once the process of doubling  $h$  is complete, only four more samples must be taken

TABLE II  
 HALVING THE STEP SIZE

Replace	$y_{n-8}$	$y_{n-7}$	..	$y_{n-3}$	$y_{n-2}$	$y_{n-1}$	$y_n$
with	$y_{n-4}$	$y_{n-\frac{7}{2}}$	...	$y_{n-\frac{3}{2}}$	$y_{n-1}$	$y_{n-\frac{1}{2}}$	$y_n$

before  $h$  can be doubled again. However,  $h$  can be halved on the next sample if required because only five readings are required.

2) *Step Size Halving*: To halve  $h$ , the interpolated values of  $y_{n-1/2}$ ,  $y_{n-3/2}$ ,  $y_{n-5/2}$ , and  $y_{n-7/2}$  must be calculated to refill the fifo with values at the new sample rate. Again, this can be achieved using the N-G fourth-order backward differencing polynomial in (14).

Setting  $s$  to  $-1/2$ ,  $-3/2$ ,  $-5/2$  and  $-7/2$  yields the following equations:

$$y_{n-1/2} = \frac{1}{128}(35y_n + 140y_{n-1} - 70y_{n-2} + 28y_{n-3} - 5y_{n-4}) \quad (16)$$

$$y_{n-3/2} = \frac{1}{128}(-5y_n + 60y_{n-1} + 90y_{n-2} - 20y_{n-3} + 3y_{n-4}) \quad (17)$$

$$y_{n-5/2} = \frac{1}{128}(3y_n - 20y_{n-1} + 90y_{n-2} + 60y_{n-3} - 5y_{n-4}) \quad (18)$$

$$y_{n-7/2} = \frac{1}{128}(-5y_n + 28y_{n-1} - 70y_{n-2} + 140y_{n-3} + 35y_{n-4}). \quad (19)$$

The values of  $y$  are then substituted as shown in Table II, giving nine equispaced readings at half the previous step size. Note that the step size can be halved at any time, unlike the process for doubling  $h$ , which is conditional on the fifo being full.

Alternative methods for halving  $h$  are possible. For example, calculating only  $y_{n-1/2}$  and  $y_{n-3/2}$  will enable ABM sampling to resume. However, a further four readings are required before step size doubling may be performed. Another method is to read subsequent values at the halved sample rate until the fifo is filled with equispaced data with respect to  $t$ , thereby enabling a simpler system to be implemented. With this method, a further four samples must be taken before ABM prediction can resume and another four samples taken before the step size may be doubled.

The minimum possible step size is determined by the rate at which data can be read from a given device. The algorithm can operate without a maximum step size. However, if the data values are constant for a sustained period, the step size can rapidly increase to the point that an important change in the value trend can be missed. To eliminate this possibility, the algorithm is modified to have both minimum ( $h_{\min}$ ) and maximum ( $h_{\max}$ ) step sizes.

Together, the ABM formula and the step size changing method form the VRS system.

#### IV. IMPLEMENTATION

The system is designed to operate at a maximum sample rate that is at least twice the Nyquist rate [22], [23] appropriate to the maximum frequency of interest in the data, thereby avoiding the possibility of being prone to undesired aliasing effects leading to unacceptable data loss [24].

To obtain a behavioral analysis of the VRS system, the modified Adams and N-G polynomial prediction methods and the step size changing algorithm are implemented in C. The data stream

is emulated using input from a disk file, each input data value corresponding to a point at the maximum sample rate. When the sample rate is halved ( $h = 2$ ), every second data point in the file is neglected. The VRS system is implemented as follows:

```

Set tol
Set h_max
Set h_min
Set h = h_min
Read first nine y values and store in y_{n-8}
to y_n
Until data sampling ends
  Calculate y_{n+1}^p
  Read y_{n+1}
  Set y_{n-8} = y_{n-7}, y_{n-7} = y_{n-6} \dots y_{n-1} = y_n, y_n =
  y_{n+1}
  If |y_{n+1}^p - y_{n+1}| \ge tol then
    If h > h_min then halve h
  else if |y_{n+1}^p - y_{n+1}| < tol/2 then
    If h < h_max then
      If four readings since last doubling
        then double h
      else do not change h
Loop
    
```

The system has also been designed in prototype form in VHDL that is targeted to an FPGA (Xilinx) to assess circuit functionality and to a CMOS process from Austria Mikro Systeme (AMS) to obtain estimates of power consumption using a synthesis tool (Synopsis). Ultimately, the intended implementation is a system-on-a-chip (SoC) with a high-speed VHDL design to maximize the bandwidth of the system, but, for many applications, a slower software implementation is adequate.

In the systems we are studying, the data is rebuilt to the original number of samples after transmission to a base station running a cubic spline interpolation algorithm.

## V. RESULTS

### A. Prediction Methods

To compare the accuracy of the Adams and N-G polynomial prediction methods described in Section III-A, data from the test data sets shown in Fig. 4 is passed through the algorithms at various fixed step sizes, i.e., the step size changing part of the VRS system is disabled. These data sets represent a sine wave (the fundamental), a sine wave with added noise, a sine wave with second harmonic introduced, a sine wave with second harmonic introduced with added noise, and, finally, with second and third harmonics and added noise. All the waveforms are originally sampled at 0.0139-rad intervals at the fundamental. The noise is in the range  $-0.072$  to  $0.072$  (i.e.,  $\pm 7.2\%$  of the fundamental signal's peak value). The results showing average errors in prediction in terms of  $y$  amplitude are shown in Table III.

The N-G polynomial performs best out of the three algorithms on the simplest data sets, i.e., those with no harmonic or noise content. However, as the waveforms become more complex, the algorithm performs less well, becoming unstable at higher step sizes, as can be seen from the rapid decrease in prediction accuracy.

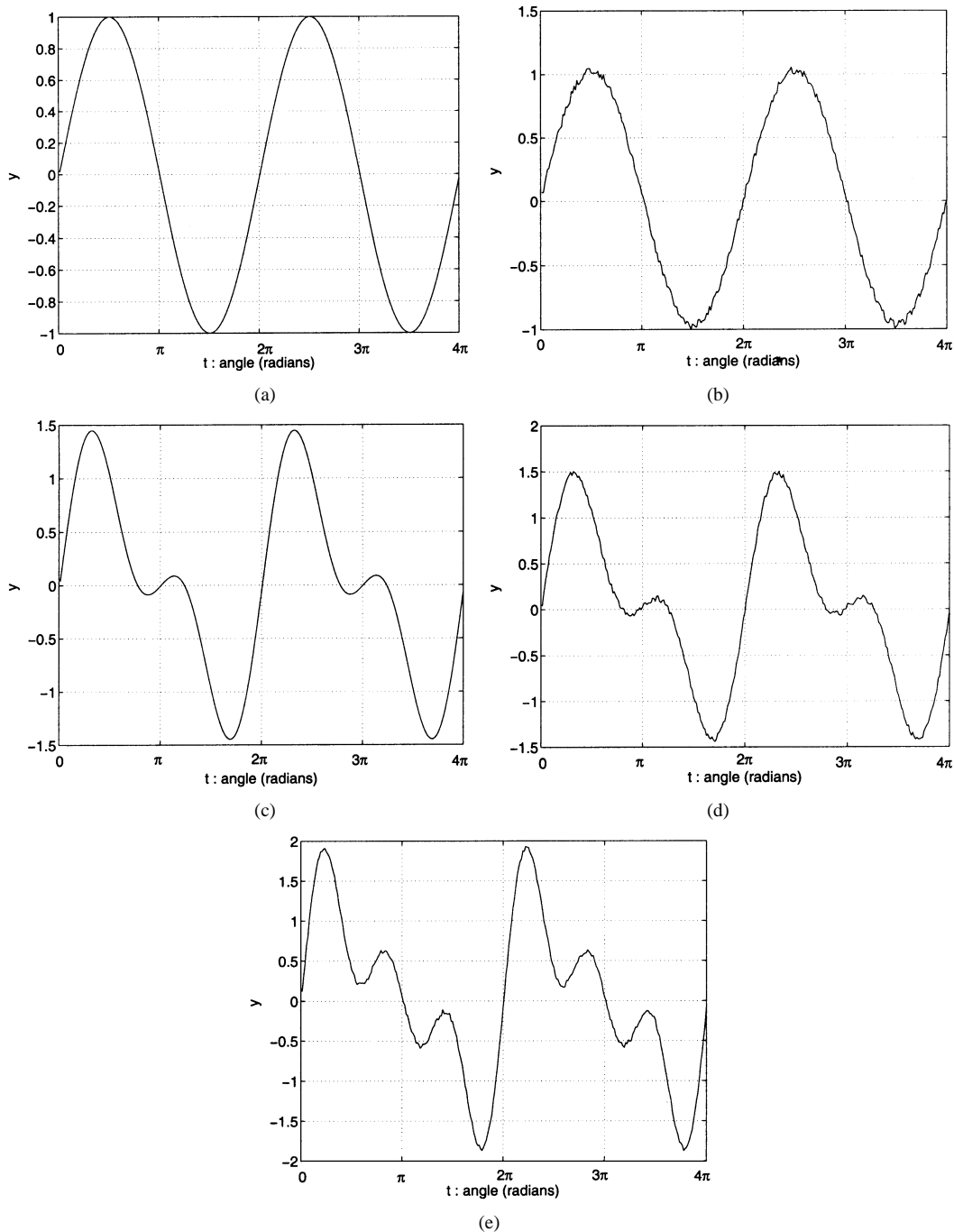


Fig. 4. Various data sets for prediction algorithm performance comparison with samples taken at 0.0139 radian intervals at the fundamental frequency. Noise is added with a random number generator. (a) Test data set 1:  $y = \sin(t)$ . (b) Test data set 2:  $y = \sin(t) + \text{noise}$ . (c) Test data set 3:  $y = \sin(t) + 0.67(\sin(2t))$ . (d) Test data set 4:  $y = \sin(t) + 0.67(\sin(2t)) + \text{noise}$ . (e) Test data set 5:  $y = \sin(t) + 0.67(\sin(2t)) + 0.67(\sin(3t)) + \text{noise}$ .

Similarly, the Adams predictor method outperforms the predictor-corrector at lower step sizes over the simple data sets. This may be attributed to the error introduced into the methods by using the secant method, with the predictor-corrector having that error introduced twice. However, the predictor performs less well over more complex waveforms, also becoming unstable.

The predictor-corrector shows the greatest stability of all the algorithms over more complex data sets. Over the sets with noise present, as may well be the case in a sensor system, the ABM algorithm consistently produces the most accurate predictions.

### B. VRS Sampling

To assess the performance of the VRS system using the ABM predictor, a data set containing actual sensor data is used, as shown in Fig. 5.

This data represents 48 h of equispaced temperature readings taken at 2.5-min intervals, the total range of the measuring instrument being  $-26$  to  $+25$  °C, and the resolution 0.2 °C. In this data set, the temperature varies between approximately 2.5 and 7 °C in a very similar way over the two 24-h periods, hence the apparent periodicity.

TABLE III  
AVERAGE PREDICTION ERROR WITH AN ADAMS PREDICTOR (MAB),  
PREDICTOR-CORRECTOR (MAM), AND A N-G INTERPOLATING  
POLYNOMIAL (N-G)

Data set	$h$	MAB	MAM	N-G
1	1	0.000606	0.000833	0.000000
	2	0.002423	0.003332	0.000003
	4	0.009648	0.013278	0.000102
	8	0.037508	0.052375	0.003194
	16	0.118421	0.196300	0.095825
2	1	0.115286	0.0069085	0.248936
	2	0.114157	0.0068422	0.246560
	4	0.114759	0.069490	0.247139
	8	0.117517	0.082758	0.243152
	16	0.155537	0.204696	0.254795
3	1	0.001672	0.002300	0.000002
	2	0.006669	0.009176	0.000068
	4	0.026001	0.036106	0.002138
	8	0.082667	0.132837	0.063800
	16	0.390779	0.436323	1.473082
4	1	0.112807	0.067534	0.243527
	2	0.117125	0.070504	0.252236
	4	0.117699	0.076545	0.249271
	8	0.142629	0.148726	0.264517
	16	0.408103	0.446665	1.489782
5	1	0.115195	0.069036	0.248741
	2	0.114273	0.070616	0.245043
	4	0.128696	0.104622	0.254566
	8	0.154262	0.276271	0.470468
	16	2.771274	1.783413	6.897779

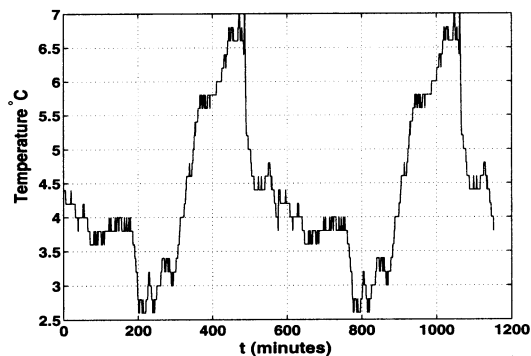


Fig. 5. Test data set for VRS algorithm representing 48 h of temperature readings taken at 2.5-min intervals.

The minimum and maximum step sizes and the tolerance are set to values consistent with the desired results (in this case,

TABLE IV  
STEP SIZE CHANGING

$t$	Unsampled	$y_{n+1}^p$	$y_{n+1}$	Next $h$
796	2.6	2.826041	2.6	4
797	2.8			
798	2.8			
799	2.8			
800	2.8	2.476041	2.8	4
801	2.8			
802	3.0			
803	3.0			
804	3.2	3.158334	3.2	8
805	3.2			
806	3.2			
807	3.0			
808	3.0			
809	3.0			
810	3.0			
811	3.0			
812	2.8	3.422917	2.8	4
813	2.8			
814	2.8			
815	2.6			
816	2.6	2.013151	2.6	2
817	2.8			
818	2.8	2.794210	2.8	2

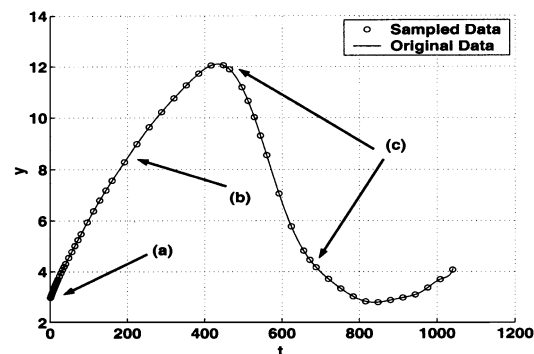


Fig. 6. Data sets with no high-frequency component result in increased step sizes. (a) Algorithm commences using a high sample rate, but (b) takes fewer samples when the trend is predictable. (c) Sample rate is increased as the data becomes less predictable.

1, 32, and 0.4, respectively), and the test data set is passed through the algorithm. The step-size changes as expected, and the results for data stream positions ( $t$ ) 796 to 818 are shown in Table IV.

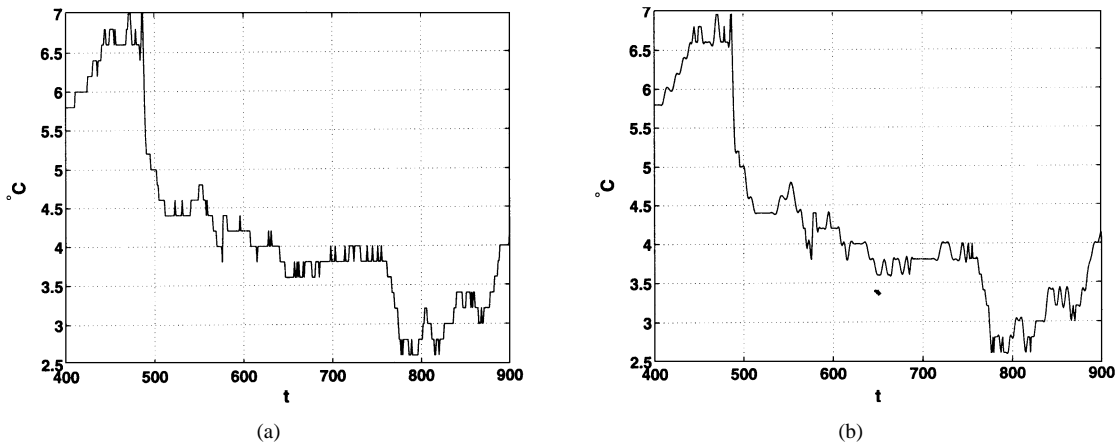


Fig. 7. When  $tol$  is low, there is good correlation between the original and rebuilt data sets. This data sampled with  $tol = 0.4$ ,  $h_{max} = 32$ , and  $h_{min} = 1$ . (a) Original data. (b) Reconstructed data.

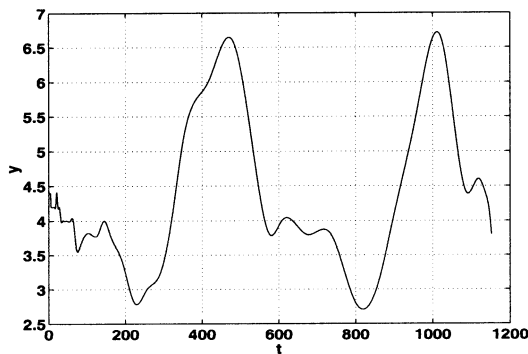


Fig. 8. Higher values of  $tol$  and  $h_{max}$  result in lower correlation but fewer readings being taken. The system also acts as a lowpass filter. This data was sampled with  $tol = 1.6$ ,  $h_{max} = 64$ , and  $h_{min} = 1$ .

This section of data illustrates the technique quite clearly. At position 804 in the simulated data stream, the predicted value was within  $tol/2$  of the actual value measured, resulting in  $h$  doubling for the next step. At first sight, it might be expected that at position 818  $h$  should again be doubled; however, this cannot happen as only two readings have been taken since the previous doubling. At position 812, the error in the prediction was greater than  $tol$ , resulting in  $h$  being halved. Over the entire data set, the maximum step size obtained using the above  $tol$ ,  $h_{max}$ , and  $h_{min}$  is 8. The reduction in the number of readings taken over the whole data set is 60%.

Data sets with no high-frequency component result in increased step sizes, thereby reducing the number of readings taken. For example, the data set shown in Fig. 6 with a  $tol$  of 0.2,  $h_{max}$  of 32, and  $h_{min}$  of 1 resulted in a maximum step size of 32 and an overall reading reduction of over 80%.

Plotting the rebuilt data, as shown in Fig. 7, illustrates that the integrity of the data is maintained, where the lowpass filtering effect that is inherent in the VRS algorithm is largely restricted to the quantization noise [24] introduced by the analog-to-digital converter.

Increasing  $tol$  has the expected result of reducing the number of readings taken but lowering the correlation between the original and rebuilt data sets due to the increased level of filtering. The result of setting  $tol$  to 1.6 and the maximum  $h$  to 64 can be seen in the rebuilt data shown in Fig. 8. Clearly, this level of

TABLE V  
CORRELATION COEFFICIENTS FOR  $tol$  WITH  $h_{max}$  FIXED AT  $\infty$

$tol$	$r$	Samples	Sample Ratio	Highest $h$
0.4	0.9982	516	1 : 2.23	8
0.6	0.9918	219	1 : 5.26	16
0.8	0.9918	159	1 : 7.25	32
1.0	0.9867	105	1 : 10.97	32
1.2	0.9860	86	1 : 13.40	32
1.4	0.9847	74	1 : 15.57	64
1.6	0.9505	54	1 : 21.33	64
1.8	0.9497	52	1 : 22.15	64
2.0	0.9497	52	1 : 22.15	64
2.2	0.9552	58	1 : 19.86	64
2.4	0.9552	58	1 : 19.86	64
2.6	0.9475	55	1 : 20.95	64
2.8	0.9484	51	1 : 22.59	64
3.0	0.9484	51	1 : 22.59	64
3.2	0.9484	51	1 : 22.59	64
3.4	0.9061	44	1 : 26.18	128

data loss would be acceptable only in instruments where a level of undersampling is tolerable because high frequencies are not of interest to the user.

Since we are examining analytic data, it is appropriate to assess the similarity between the original and rebuilt data sets in terms of the linear correlation coefficient ( $r$ ) [25]. An  $r$  value of 1 represents complete positive correlation,  $-1$  complete negative correlation, and 0 no correlation  $-1 \leq r \leq 1$ .

By setting  $h_{max}$  to  $\infty$  and passing the data through the algorithm with various  $tol$  values, the effect on  $r$  is obtained. Values for various tolerances are shown in Table V. A  $tol$  value of 3.4 results in  $h$  doubling at the maximum possible rate, and increasing  $tol$  further would have no effect on this data set.

TABLE VI  
CORRELATION COEFFICIENTS FOR  $h$  WITH  $tol$  FIXED AT 1.4 (2.7% OF THE TOTAL RANGE OF  $-26\text{ }^{\circ}\text{C}$  TO  $+25\text{ }^{\circ}\text{C}$ )

$h_{max}$	$r$	Samples	Sample Ratio	Highest $h$
2	0.9985	585	1 : 1.97	2
4	0.9971	302	1 : 3.82	4
8	0.9954	163	1 : 7.07	8
16	0.9919	96	1 : 12.00	16
32	0.9877	76	1 : 15.16	32
64	0.9847	74	1 : 15.57	64
128	0.9847	74	1 : 15.57	64

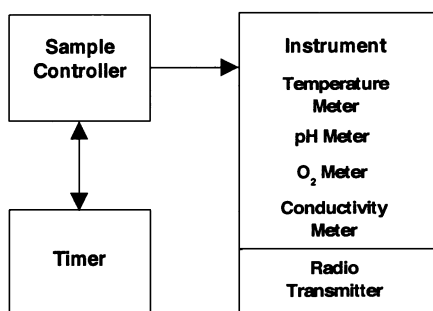


Fig. 9. Example system from [4].

Similarly, fixing  $tol$  to an arbitrary figure and passing the data through the algorithm with various  $h_{max}$  values, the effect of  $h_{max}$  on  $r$  is obtained. Values for various maximum step sizes at a fixed  $tol$  of 1.4 (or 2.7% of the instrument's total range of  $-26\text{ }^{\circ}\text{C}$  to  $+25\text{ }^{\circ}\text{C}$ ) are shown in Table VI. At this tolerance, it can be seen that  $h_{max}$  ceases to have an effect on the data sampling at a value of 64.

### C. Energy Saving

Consider the system shown in Fig. 9, which is comprised of sensors with associated electronics, a radio transmitter, a timer, and the VRS sample controller. The power consumption data for the components in this typical microsystem [4], as shown in Table VII, are obtained by targeting the design, through synthesis, onto a  $0.6\text{-}\mu\text{m}$  CMOS process. In this system, the timer interacts with the VRS controller to obtain a sample interval for the next reading and powers off all the remaining components, including the controller, during this interval. The timer is active continuously and therefore represents the 0.1-mW constant power overhead of the system. The other components represent the 29.48-mW variable power overhead. The samples are transmitted in triplicate, and the total transmission time for the four measured parameters is 192 ms. Since the sampling intervals with VRS enabled are relatively long, the energy saved during power-down periods more than outweighs that consumed during the powering up process.

The efficiency/energy saving is calculated as

$$\frac{\text{energy consumption of system with VRS}}{\text{energy consumption of system only}} \times 100\%. \quad (20)$$

TABLE VII  
POWER CONSUMPTION OF COMPONENTS

Component	Power mW
Timer	0.10
Temperature Meter	0.36
pH Meter	0.24
O <sub>2</sub> Meter	1.36
Conductivity Meter	0.22
Radio Transmitter	24.00
Sample Controller	3.30

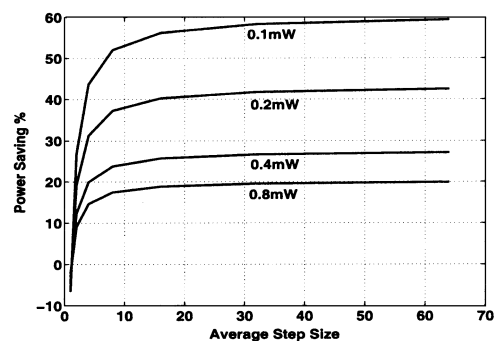


Fig. 10. Energy savings. The power labels represent the constant power overhead for each curve on the graph. The higher this overhead, the lower the power savings gained with VRS. Note that there is potential for an energy loss if the average step size is close to 1.

As can be seen in Fig. 10, the higher the constant power overhead of the system, the lower the overall power savings gained by the variable-rate sampling algorithm. Power savings rapidly reach a plateau when the average step size is relatively low, and it is clear that even with modest increases in the average sample step and a large constant power overhead, power savings of 15% can be readily achieved.

## VI. CONCLUSIONS

We have developed a new real-time sample rate controller based on modified Adams ODE numerical methods. The effectiveness of the VRS technique is dependent on the high-frequency content of the data being sampled. Modified Adams formulae of higher or lower order than those used here are easily implemented to alter prediction accuracy as required. An advantage of the system is that real data is stored or transmitted, as opposed to signal function coefficients [26].

The simplicity of the algorithms enable them to be implemented with a small hardware or software overhead, making them attractive for small footprint systems requiring low-power consumption, low memory usage, and minimum hardware size.

## ACKNOWLEDGMENT

The authors wish to thank the reviewers for their valuable comments and suggestions.



## REFERENCES

- [1] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 299–316, June 2000.
- [2] M. B. Srivastava, A. P. Chandrakasan, and R. W. Brodersen, "Predictive system shutdown and other architectural techniques for energy efficient programmable computation," *IEEE Trans. VLSI Syst.*, vol. 4, pp. 42–55, Mar. 1996.
- [3] A. Banerjee, A. S. Dhar, and S. Banerjee, "FPGA realization of a CORDIC based FFT processor for biomedical signal processing," *Microprocessors Microsyst.*, vol. 25, no. 3, pp. 131–142, May 2001.
- [4] L. Wang *et al.*, "An integrated sensor microsystem for industrial and biomedical applications," in *Proc. 19th IEEE Instrumentation Measurement Technol. Conf.*, vol. 2, May 2002, pp. 1717–1720.
- [5] J. W. Mark and T. D. Todd, "A nonuniform sampling approach to data compression," *IEEE Trans. Commun.*, vol. COM-29, pp. 24–32, Jan. 1981.
- [6] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan, "A level-crossing sampling scheme for A/D conversion," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 335–339, Apr. 1996.
- [7] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*, 6th ed. Reading, MA: Addison-Wesley, 1999.
- [8] C. Runge, "Ueber die numerische auflösung von differentialgleichungen," *Math. Annalen*, vol. 46, pp. 167–178, 1895.
- [9] W. Kutta, "Beitrag zur näherungsweise integration totaler differentialgleichungen," *Zeitschrift Mathematik Physik*, vol. 46, pp. 435–453, 1901.
- [10] F. Bashforth and J. C. Adams, *An Attempt to Test the Theories of Capillary Action ... With an Explanation of the Method of Integration Employed ...*. Cambridge, U.K.: Cambridge Univ. Press, 1883.
- [11] F. R. Moulton, *New Methods in Exterior Ballistics*. Chicago, IL: Univ. Chicago Press, 1926.
- [12] S. J. Ovaska, "Newton-type predictors—A signal processing oriented viewpoint," *Signal Process.*, vol. 25, no. 2, pp. 251–257, Nov. 1991.
- [13] O. Vainio and S. J. Ovaska, "Multirate polynomial prediction with unevenly spaced samples," *IEEE Trans. Instrum. Meas.*, vol. 41, pp. 506–509, Aug. 1992.
- [14] T. I. Laakso and S. J. Ovaska, "Prefiltering approach for optimal polynomial prediction," *IEEE Trans. Signal Processing*, vol. 44, pp. 701–705, Mar. 1996.
- [15] P. T. Harju, "Polynomial prediction using incomplete data," *IEEE Trans. Signal Processing*, vol. 45, pp. 768–770, Mar. 1997.
- [16] G. Kishi and N. Maeda, "Sampling theorems for signals given by linear differential equations with constant coefficients," *Electron. Commun. Jpn.*, vol. 53-A, no. 4, pp. 29–36, 1970.
- [17] J. D. Hoffman, *Numerical Methods for Engineers and Scientists*. New York: McGraw-Hill, 1992.
- [18] L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*. San Francisco, CA: Freeman, 1975.
- [19] R. K. Brayton, F. G. Gustavson, and G. D. Hachtel, "A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas," *Proc. IEEE*, vol. 60, pp. 98–108, Jan. 1972.
- [20] W. M. G. van Bokhoven, "Linear implicit differentiation formulas of variable step and order," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 109–115, Feb. 1975.
- [21] F. T. Krogh, "Algorithms for changing the step size," *SIAM J. Numer. Anal.*, vol. 10, no. 5, pp. 949–965, Oct. 1973.
- [22] H. Nyquist, "Certain factors affecting telegraph speed," *Bell Syst. Tech. J.*, vol. 3, p. 324, Apr. 1924.
- [23] —, "Certain topics in telegraph transmission theory," *Trans. AIEE*, vol. 47, pp. 617–644, 1928.
- [24] B. Mulgrew, P. Grant, and J. Thomson, *Digital Signal Processing Concepts and Applications*. London, U.K.: Palgrave, 1999.
- [25] J. S. Bendat and A. G. Piersol, *Engineering Applications of Correlation and Spectral Analysis*. New York: Wiley, 1980.
- [26] Y. Tanaka and H. Kimura, "Low-bit-rate speech coding using a two-dimensional transform of residual signals and waveform interpolation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, 1994, pp. I-173–I-176.



**Geoff B. Irvine** received the M.Sc. degree in Software Technology from Napier University, Edinburgh, U.K., in 1999. He is currently pursuing the Ph.D. degree with the Microsystem Technology Group, Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, U.K.

He has been a software development engineer in industry on a diverse range of projects. His research interests lie in the areas of digital signal processing and low-power microsystems.



**Lei Wang** received the B.S. degree in information and control engineering and the Ph.D. degree in biomedical engineering from Xi'an Jiaotong University, Xi'an, China, in 1995 and 2000, respectively.

After graduation, he was an academic visitor with the Department of Mechanical Engineering, University of Dundee, Dundee, U.K. In 2001, he joined the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, U.K., as a Research Associate. His research interests focus on physiological measurement, digital signal

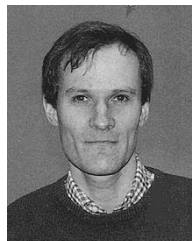
processing, and integrated circuit design.



**Peter Dickman** (M'93) received the B.A. degree in mathematics in 1982, the Diploma in computing science in 1984, and the Ph.D. degree in computer science in 1992, all from the University of Cambridge, Cambridge, U.K.

Prior to his doctoral studies, he produced software for embedded systems in industry. Between 1991 and 1993, he held an ERCIM Fellowship at INRIA Rocquencourt, Paris, France; INESC, Lisbon, Portugal; and GMD, Sankt Austen, Bonn, Germany; and was a visiting academic at the University of Copenhagen, Copenhagen, Denmark. In 1993, he joined the Computing Science Department, University of Glasgow, where he is currently a senior lecturer and acting vice-dean (postgraduates) for the Faculty of Information and Mathematical Sciences. His research interests are in distributed and embedded systems, with an emphasis on resource management, software architectures, and run-time software evolution.

Dr. Dickman is a member of ACM.



**David R. S. Cumming** (M'94) received the B.Eng. degree from the University of Glasgow, Glasgow, U.K., in 1989 and the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 1993.

He has worked variously on mesoscopic device physics, RF characterization of novel devices, fabrication of diffractive optics for optical and sub-millimeter wave applications, and microelectronic design. He is presently a senior lecturer and EPSRC Advanced Research Fellow in Electronics and Electrical Engineering at the University of Glasgow, Glasgow, U.K. where he leads the Microsystem Technology Group.