2-1992

Analysis of Algorithms for Velocity Estimation from Discrete Position Versus Time Data

Ronald H. Brown

Susan C. Schneider PhD

Michael G. Mulligan

Analysis of Algorithms for Velocity Estimation from Discrete Position Versus Time Data

Ronald H. Brown, Susan C. Schneider, and Michael G. Mulligan

Abstract—**This paper investigates algorithms for constructing velocity approximations from discrete position versus time data. The study is limited to algorithms suitable to provide velocity information in discrete-time feedback control systems such as microprocessor-based systems with a discrete position encoder. Velocity estimators based on lines per period, reciprocal-time, Taylor series expansions, backward difference expansions, and least-square curve fits are presented. Based on computer simulations, comparisons of relative accuracies of the different algorithms are made. The least-squares velocity estimators filtered the effect of imperfect measurements (''noise'') best, whereas the Taylor series expansions and backward difference equation estimators respond better to velocity transients.**

I. Introduction

With the advent of the microprocessor, many motion control systems are being implemented as discrete-time systems. The performance of motion control systems can often be enhanced by including some type of velocity feedback where, for example, the velocity is estimated from discrete position versus time information provided by an (optical) encoder. In essence, the velocity is estimated by performing an approximate derivative operation on the discrete data. Many designs of discrete-time derivative filters exist today [1], [2]; unfortunately, most of these are unsatisfactory for control applications as the delay inherent to these derivative filters adversely affects stability. Furthermore, it is well known that derivative operators tend to magnify errors. However, several methods have been developed to estimate velocity from discrete data while reducing estimate errors and minimizing the delay [3]–[8].

In this paper, some existing velocity estimator algorithms are reviewed and the mathematical theories of the various algorithms are presented. With the aid of computer simulations, the accuracy of these algorithms are evaluated, including the dynamic range and transient response. In addition, *a new set of algorithms for velocity estimators based on least squares is proposed*. The major contribution of this work is the presentation of the theory behind many existing methods for estimating velocity from discrete position-time data, the development of a new family of algorithms based on least-squares fits, and the error analysis of the various algorithms. The major conclusion of these analyses is that the choice of ''best estimator'' to use is, at best, application dependent.

Manuscript received August 14, 1990; revised, September 28, 1991.

The authors are with the Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI 53233.

IEEE Log Number 9105264.

II. Algorithms for Velocity Estimators

The algorithms investigated in this study fall into two general classes of velocity estimators, defined by how the time and position information are acquired. When position and time information are obtained using an optical encoder system, the user can make a choice regarding whether time information or position information will be required. A rotary optical encoder will typically produce two square waves in quadrature. Each transition of both waves is detected as an encoder line, so the user can choose to count the number of encoder lines in a fixed period of time; or the user can choose to measure the time during which a set number of encoder lines is counted. In either case, only one variable is measured, the other variable is assumed to be constant, which leads to a definition of velocity estimator class. ''Fixed-time'' velocity estimators are those in which the time between successive samples is known (at least approximately), and the distance traveled over this fixed time interval is measured by counting lines. For ''fixed-position'' velocity estimators, the estimation is based on the measurement of time required to travel fixed distances.

Any numerical method that calculates a derivative from discrete position versus time data can be used for a velocity estimator. At the kth sampling instant, the measured position traveled x_k and the time at the sampling instant t_k are generally available. The simplest fixed-time velocity estimator can be called the lines per period (LPP) estimator. In the LPP method, the velocity is estimated by counting the number of encoder lines detected during a sampling period; if $\Delta x_k = x_k - x_{k-1}$, the number of lines counted in the kth time interval, and T is the length of each time interval, then the velocity estimation for the kth interval is $\hat{v}_k = \Delta x_k / T$ lines per time unit or simply $\hat{v}_k = \Delta x_k$ lines per sampling period. A first-order fixed-position estimator is implemented as the reciprocal of the measured time (RT). When the position between successive time measurements is defined to be one line, and $T_k = t_k - t_{k-1}$, the measured time to transverse this distance, the velocity estimate for the kth interval is $\hat{v}_k = 1/T_k$ lines per time unit.

The LPP fixed-time velocity estimator and the RT fixed-position velocity estimator are mathematically equivalent, even through (at first glance) these estimators appear to be different. The estimations of velocity are $\Delta x_k / T_k$ for both methods, with Δx_k constant for the fixed-position velocity estimator and T_k constant for the fixed-time velocity estimator. Assuming accurate data measurements, both of these methods provide a velocity estimate for the current sample that corresponds to the *average* velocity over that sampling interval. During a velocity transient, the "true" velocity at the sampling instant will differ from the average velocity; this velocity difference is equivalent to a time delay that can degrade the performance of a control system and may lead to instability.

In general, adding higher order terms to the derivative approximation should improve the transient response. For velocity estimators, higher order approximations to the derivative are achieved by combining present and "earlier" velocity estimates. Several higher order derivative algorithms for approximating derivatives from discrete data are analyzed in this section. These higher order derivative algorithms, however, tend to magnify measurement errors [9]. To reduce the effect of these errors, a class of derivative algorithms based on least-square fits through data points is proposed. (Because an estimator with minimal delay is sought, standard "windowing" techniques for designing differentiating digital filters are not considered in this paper because of the inherent time delay associated with these filters.)

A. Taylor Series Expansion (TSE) Velocity Estimator

An estimator for estimating the velocity can be developed using a Taylor series expansion of velocity [7]. The velocity at time t_k can be estimated from the velocity at time t_β by the TSE:

$$\begin{aligned}\hat{v}_k &= \hat{v}_\beta + \frac{1}{1!}\frac{d\hat{v}_\beta}{dt}(t_k - t_\beta) + \frac{1}{2!}\frac{d^2\hat{v}_\beta}{dt^2}(t_k - t_\beta)^2 \\ &\quad + \frac{1}{3!}\frac{d^3\hat{v}_\beta}{dt^3}(t_k - t_\beta)^3 + \cdots \\ &= \sum_{j=0}^{\infty}\frac{1}{j!}\hat{v}_\beta^{(j)}(t_k - t_\beta)^j \qquad (1)\end{aligned}$$

where $\hat{v}^{(j)}$ is the jth derivative of v.

If $\hat{v}_\beta$ is estimated as the average velocity during the most recently measured sampling interval and this average velocity is estimated to occur at the center of the sampling interval, then $\hat{v}_\beta \approx \Delta x_k / T_k$ and $(t_k - t_\beta) \approx T_k/2$. If the TSE is truncated after the first term with these estimations, the estimated velocity $\hat{v}_k = \Delta x_k / T_k$ is the same as the reciprocal-time estimator for fixed-position data and is the same as the lines-per-period estimator for fixed-time data. When $\hat{v}_{\beta k}$ is estimated from the kth period as $\Delta x_k / T_k$, and $\hat{v}_{\beta k-1}$ is estimated from the $(k-1)$th period as $\Delta x_{k-1} / T_{k-1}$, the first derivative of the velocity at the middle of the ith period can be approximated by

$$\hat{v}_{\beta i}^{(1)} \approx \frac{\Delta v}{T} \approx \frac{\hat{v}_{\beta i} - \hat{v}_{\beta i-1}}{T_i}. \qquad (2)$$

Higher order derivatives of $\hat{v}_{\beta i}$ are also approximated in a similar fashion by

$$\hat{v}_{\beta i}^{(j)} \approx \frac{\Delta \hat{v}_{\beta i}^{(j-1)}}{T} \approx \frac{\hat{v}_{\beta i}^{(j-1)} - \hat{v}_{\beta i-1}^{(j-1)}}{T_i}. \qquad (3)$$

Using these derivative approximations, the velocity estimator of (1) then becomes

$$\hat{v}_k = \sum_{j=0}^{N}\frac{1}{j!}\hat{v}_\beta^{(j)}\left(\frac{T_k}{2}\right)^j \qquad (4)$$

where the TSE is truncated after the Nth-order derivative term. The TSE estimators for N equal to 0, 1, and 2 are shown in Table I. The fixed-position estimators have been normalized by setting $\Delta x_k = 1$ and the fixed-time estimators have been normalized by setting $T_k = 1$.

The TSE estimator truncated before any derivative term is the same as the reciprocal-time or the lines-per-period estimators. The TSE estimator truncated after the first-order derivative term with these approximations (called the second-order TSE estimator), is the average-plus-acceleration estimator described in [6] (with the scaling parameter set equal to $\frac{1}{2}$).

B. Backward Difference Expansion (BDE) Velocity Estimator [9]

The BDE method for obtaining the derivative of the function $t(x_k)$ or $x(t_k)$ is developed by assuming that the actual function can be replaced by an interpolating polynomial that *exactly* fits the data points. The first and higher order derivatives of the function are obtained in terms of the appropriate finite difference approximations for the derivatives of the approximating polynomial. For fixed-position data, the backward difference equation is obtained by expanding the function t_{k-i} in a Taylor series around t_k and then solving for dt_k / dx,

$$t_{k-i} = t_k + (-i)\frac{dt_k}{dx} + \frac{(-i)^2}{2!}\frac{d^2t_k}{dx^2} + \frac{(-i)^3}{3!}\frac{d^3t_k}{dx^3} + \cdots \qquad (5)$$

The first-order BDE for dt_k / dx is obtained by expanding t_{k-1} using (5) and then truncating the expansion after the first term on the right-hand side, as shown in (6):

$$\frac{dt_k}{dx} = (t_k - t_{k-1}) + \frac{(-1)^2}{2!}\frac{d^2t_k}{dx^2} + \frac{(-1)^3}{3!}\frac{d^3t_k}{dx^3} + \cdots \qquad (6)$$

The first-order BDE is identical to the reciprocal-time estimator (as was the TSE estimator when truncated before the derivative terms). The second-order BDE for dt_k / dx is obtained when both t_{k-1} and t_{k-2} are expanded in the Taylor series of (5), and the resulting two equations are solved for both dt_k / dx and dt_k^2 / dx^2 with third-order derivative terms neglected. This procedure is continued to include terms containing t_{k-3} when the third-order BDE for dt_k / dx is derived. The second- and third-order BDE expansions for dt_k / dx are written both in terms of Δx_k and time interval T_k in Table II. The algorithms for first-, second-, and third-order fixed-time BDE velocity estimators, presented in Table II, are obtained by exchanging t with x and T with Δx. In this table, the fixed-position estimators have been normalized by setting $\Delta x_k = 1$ and the fixed-time estimators have been normalized by setting $T_k = 1$. For

TABLE I
THE FIRST-, SECOND-, AND THIRD-ORDER TAYLOR SERIES EXPANSION ESTIMATORS FOR BOTH POSITION-AND FIXED-TIME ESTIMATORS

Order	Fixed-Position TSE Estimators (in lines/clock cycles)	Fixed-Time TSE Estimators (in lines/sampling period)
1st	$\hat{v}_k = \dfrac{1}{T_k}$	$\hat{v}_k = \Delta x_k$
2nd	$\hat{v}_k = \dfrac{1}{T_k} + \dfrac{1}{2}\left(\dfrac{1}{T_k} - \dfrac{1}{T_{k-1}}\right)$	$\hat{v}_k = \Delta x_k + \frac{1}{2}(\Delta x_k - \Delta x_{k-1})$
3rd	$\hat{v}_k = \dfrac{1}{T_k} + \dfrac{1}{2}\left(\dfrac{1}{T_k} - \dfrac{1}{T_{k-1}}\right) + \dfrac{T_k}{8}\left(\dfrac{\dfrac{1}{T_k} - \dfrac{1}{T_{k-1}}}{T_k} - \dfrac{\dfrac{1}{T_{k-1}} - \dfrac{1}{T_{k-2}}}{T_{k-1}}\right)$	$\hat{v}_k = \Delta x_k + \frac{1}{2}(\Delta x_k - \Delta x_{k-1}) + \frac{1}{8}(\Delta x_k - 2\Delta x_{k-1} + \Delta X_{k-2})$

T_k is the change in time in clock cycles over the k-th position interval, thus $\hat{v}_k$ has units of lines/clock cycle.
Δx_k is the change in position in lines over the k-th sampling period, thus $\hat{v}_k$ has units of lines per sampling period.

TABLE II
THE FIRST, SECOND, AND THIRD-ORDER BACKWARD DIFFERENCE EXPANSION ESTIMATORS FOR BOTH POSITION- AND FIXED-TIME ESTIMATORS.

Order	Fixed-Position BDE Estimators (in-clock cycles line)	Fixed-Time BDE Estimators (in lines/sampling period)
1st	$\dfrac{dt_k}{dx} = T_k$	$\dfrac{dx_k}{dt} = \Delta x_k$
2nd	$\dfrac{dt_k}{dx} = T_k + \frac{1}{2}(T_k - T_{k-1})$	$\dfrac{dx_k}{dt} = \Delta x_k + \frac{1}{2}(\Delta x_k - \Delta x_{k-1})$
3rd	$\dfrac{dt_k}{dx} = T_k + \frac{1}{2}(T_k - T_{k-1}) + \frac{1}{3}(T_k - 2T_{k-1} + T_{k-2})$	$\dfrac{dx_k}{dt} = \Delta x_k + \frac{1}{2}(\Delta x_k - \Delta x_{k-1}) + \frac{1}{3}(\Delta x_k - 2/\Delta x_k - 2\Delta x_{k-1} + \Delta x_{k-2})$

T_k is the change in time in clock cycles over the kth position interval, thus $dt_k/dx = 1/\hat{v}_k$ has units of clock cycles per line.
Δx_k is the change in position in lines over the kth sampling period, thus $dx_k/dt = \hat{v}_k$ has units of lines per sampling period.

fixed-time estimators, the second-order BDE is the same as the second-order TSE. The third-order BDE differs from the third-order TSE estimator only in the coefficients of the third-order terms.

Although not referred to as such, the methods implemented in [8] are fixed-time BDE velocity estimators. Note that Nth-order BDE's can be implemented as finite impulse response digital filters of length equal to $(N + 1)$ if the t_k's or x_k's are used, and of length N if the time or position intervals (T_k's or Δx_k's) are used.

C. Least-Squares Fit (LSF) Velocity Estimator

An alternative algorithm to the "exact polynomial fit to data" approach used for the BDE method is to perform a least-squares fit to the measured data; such an approach has been utilized as a means of predicting the "next time" an encoder pulse will occur [11]. This least-squares fit calculation can also be used to implement a velocity estimator.

In the least-squares fit technique, an Nth-order polynomial can be fit through the M most recent data points in a least-squares sense provided that $M > N + 1$. The velocity estimate is made by evaluating the derivative of the polynomial at the most recent data point. For fixed-position data, an approximating polynomial of the form

$$\hat{t}_k = c_0 + c_1 x_k + c_2 x_k^2 + \cdots + c_N x_k^N \tag{7}$$

is assumed, where N is the order of the polynomial fit. From this equation, a velocity estimate, calculated as the derivative $d\hat{t}_k/dx$ evaluated at the most recent sample, can be obtained as

$$\frac{d\hat{t}_k}{dx} = c_1 + 2c_2 x_k + \cdots + N c_N x_k^{N-1}. \tag{8}$$

The coefficients, $c_0 \cdots c_N$, are chosen to minimize the total squared error between the M most recent data points, $t_k \cdots t_{k-M+1}$, and the M most recent estimates $\hat{t}_k \cdots \hat{t}_{k-M+1}$.

Equation (7) can be expanded to include the M most recent estimates as

$$\hat{\mathbf{t}} = \mathbf{Ac} \tag{9}$$

where $\hat{\mathbf{t}}$ is an M vector containing the M most recent estimates, $\mathbf{c}$ is an $(N + 1)$-vector containing the polynomial coefficients, and $\mathbf{A}$ is an $M \times (N + 1)$-matrix with the ith row containing the 0th through Nth powers of x_i. With the assumption that measurements are obtained on consecutive encoder lines, and recognizing that subtracting an offset from the position data does not affect the value of the derivative, **A** can be written as

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1^2 & 1^3 & \cdot & \cdot & 1^N \\ 1 & 2 & 4 & 8 & \cdot & \cdot & 2^N \\ 1 & 3 & 9 & 27 & \cdot & \cdot & 3^N \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & M & M^2 & M^3 & \cdot & \cdot & M^N \end{bmatrix} \tag{10}$$

where the offset x_i's, for $i = k - M + 1$ to k are equal to 1 to M.

Using standard LSF techniques [12], the coefficient vector c that minimizes the total squared error between the measurements and the estimates is

$$\mathbf{c} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{t} = \mathbf{A}^{\dagger}\mathbf{t}. \tag{11}$$

The derivative of the approximating polynomial $\hat{t}_m$, with respect to position x, evaluated at the most recently acquired data point, with substitution for the coefficients c_i from (11), can be written as

$$\frac{d\hat{t}_M}{dx} = \dot{\mathbf{q}}^T\mathbf{A}^t\mathbf{t} = \dot{\mathbf{h}}^T\mathbf{t} \tag{12}$$

where $\dot{\mathbf{q}}^T = [0\ 1\ 2M\ 3M^2\ \cdots\ (N-1)M^{N-2}NM^{N-1}]$ and $\dot{\mathbf{h}}^T = \dot{\mathbf{q}}^T\mathbf{A}^{\dagger}$.

Thus the velocity estimate, $\hat{v}_k$ is the reciprocal of the derivative of the approximating polynomial $d\hat{t}_M/dx$, which is equivalent to $d\hat{t}_k/dx$ and is obtained as a linear combination of the kth and previous $M - 1$ samples of time. The finite-impulse response (FIR) filter coefficients for a line fit to four points (LSF 1/4), a parabola fit to eight points (LSF 2/8), and a cubic fit to eight points (LSF 3/8) are listed in Table III. Note that the LSF estimators can be implemented as finite impulse response digital filters of order M.

III. Simulation Results

In order to examine and compare the performance of the velocity estimation schemes described above, several trial velocity profiles were used to "test" the estimators. Although the profiles do not completely test these algorithms, the results from two of these tests are sufficient to show that there is no one "best" way to estimate velocity in all situations. These velocity profiles were used to obtain the discrete data points for both time- and fixed-position systems needed for the velocity estimators. The two velocity profiles, shown in Fig. 1, were (A) a low-speed underdamped velocity profile with an initial speed of 1500 lines per second followed by a high acceleration with overshoot, settling at a final speed of 10 300 lines per second, and (B) a high-speed underdamped velocity profile with an initial speed of 15 500 lines per second, a high acceleration with overshoot, settling at a final speed of 103 300 lines per second. A third velocity profile (a low-speed trapezoidal velocity profile) was also included in the simulations to show the effect of the algorithms for a common motor velocity trajectory and is also shown in Fig. 1.

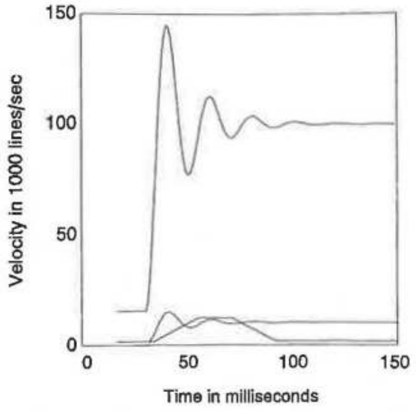


Fig. 1. Trial velocity profiles used in the system simulations.

TABLE III
The FIR Filter Coefficients for the Line Fit to Four (LSF 1/4), the Quadratic Fit to Eight (LSF 2/8), and the Cubic Fit to Eight (LSF 3/8) Least Squares Fit Estimators

	LSF 1/4	LSF 2/8	LSF 3/8
$\dot{h}_1$	−0.3000000	0.2083333	−0.2777778
$\dot{h}_2$	−0.1000000	−0.0178571	0.3293651
$\dot{h}_3$	0.1000000	−0.1607143	0.3253968
$\dot{h}_4$	0.3000000	−0.2202381	−0.0119048
$\dot{h}_5$		−0.1964286	−0.4047619
$\dot{h}_6$		−0.0822857	−0.5753968
$\dot{h}_7$		0.1011905	−0.2460317
$\dot{h}_8$		0.3750000	0.8611111

The most significant error source for all fixed-time estimators is "position measurement truncation." Position measurement truncation errors occur because position can only be measured as an integer number of lines, thus causing the truncation of the position measurement. Additional position measurement errors can occur due to imperfections in the encoder, i.e., the encoder lines are not equally spaced in position, but typically, the deviations will not exceed a position increment in magnitude. Methods have been proposed to eliminate position measurement truncation errors by also measuring the time between the first and last lines in a sampling interval [3].

Two error sources are significant for fixed-position estimators. "Position-base jitter" is caused by imperfections in the position base, that is, the change in position between sampling instants varies. This error source, caused by unequal positional spacings in the encoder, is quite prevalent in real systems. The other significant source of errors is "time measurement truncation." Typically, the time required to transverse the position base distance is measured by counting clock cycles. Only an integer number of clock cycles can be counted, leading to a time measurement error of up to ± 1 clock period. (In addition, a drawback of all fixed-position methods occurs at high speed. The amount of processing time to calculate the velocity is fixed, where the information to be processed is generated at a rate proportional to speed. The top measurable speed of the system is limited to less than $1/T_p$ lines per second, where T_p is the processing time. The top speed can be increased by measuring the time between K encoder lines, where K is a positive integer. This has the effect of reducing the resolution of the encoder and will increase the top speed by a factor of K, but this is detrimental at slow speeds as the number of estimations has been reduced by K. Typically, the velocity changes slowly compared to the time between adjacent encoder lines, allowing the value of K to be dynamically adjusted resulting in an estimator with a larger dynamic range [4], [5].)

The fixed-time and fixed-position velocity estimators tested were:

1) First-order algorithms, i.e., lines-per-period estimator (LPP) for fixed-time and reciprocal-time estimator (RT) for fixed-position
2) Taylor-series expansion estimator truncated after two and three terms (TSE 2 and TSE 3)
3) Second- and third-order backward-difference expansion estimators (BDE 2 and BDE 3)
4) Least-squares fit estimators for
   a) A line fit to 4 points, (LSF 1/4)
   b) A quadratic fit to 8 points, (LSF 2/8) and
   c) A cubic fit to 8 points, (LSF 3/8)

For the purpose of comparison, the velocity estimators were first tested with the time-position sequences developed for a "perfect encoder," where a "perfect encoder" is defined to be an encoder with no errors in position, i.e., the spacing between all adjacent encoder lines Δx was set identically to one. Then the estimators were tested with an imperfect encoder. Typically, a rotary optical encoder produces two square waves in quadrature. Each transition of each wave is detected as an encoder line. When the duty cycle of the encoder output signals is not exactly symmetrical, and/or the alignment of the two signals is not in perfect quadrature, the decoded lines do not correspond to equal position increments. This "imperfect encoder" effect was simulated by varying the Δx_k values cyclically with a period of four lines. This type of encoder error is consistent with what might be obtained from a real encoder. The following Δx_ks were used to simulate an imperfect encoder: $\Delta x_{4i+1} = 0.95$, $\Delta x_{4i+2} = 0.95$, $\Delta x_{4i+3} = 0.90$, and $\Delta x_{4i} = 1.2$ for all i. The results presented here are typical of all the simulations and similar errors occurred in simulations with other combinations of encoder errors. It should also be noted that the error at the transients was quite sensitive to variation in the time and position of the transient.

A. Analysis of Errors in Fixed-Time Velocity Estimators

The fixed-time velocity estimators were simulated using the velocity profiles in Fig. 1 with both the perfect and imperfect encoder implementations. The sampling period was set at one millisecond (1 ms). The percent relative errors, e_k's, where,

$$e_k = 100\frac{\hat{v}_k - v_k}{v_k} \tag{13}$$

where v_k is the true velocity at the kth sample and $\hat{v}_k$ is the velocity estimate, are plotted versus time in Fig. 2 for the perfect encoder. The plots for all encoders have been offset from each other for clarity, and the velocity profile has been overlaid for comparison. Figs. 3 and 4 contain the root mean squared (rms) relative errors of the estimations generated from the three test velocity profiles using the various methods for the perfect and imperfect encoders, respectively. The fixed-time estimators are insensitive to the type of encoder error considered; the rms values of the relative errors for the simulations with the imperfect encoder are virtually identical to those obtained during simulations with the perfect encoder. The errors in the fixed-time estimations are due to either position measurement truncation or poor transient response.

At high speeds, when the number of lines counted in a period is large, the effect of the position measurement trunca-

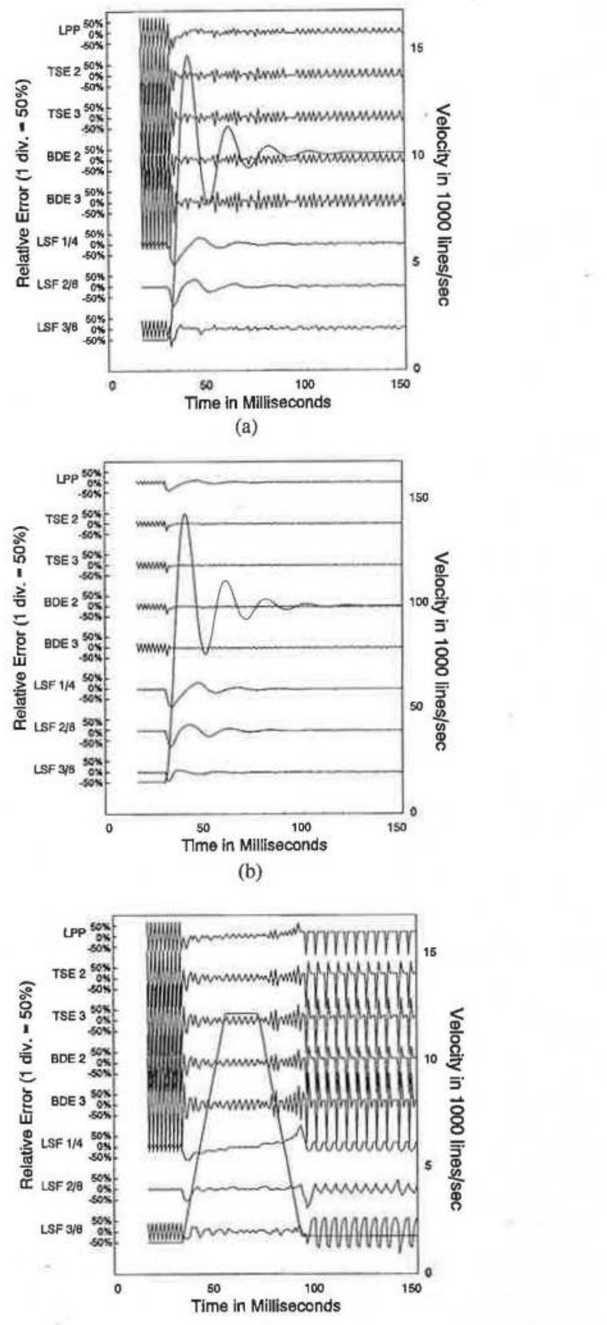


Fig. 2. The fixed-time estimator relative error with the *perfect* encoder. (a) With the low-speed underdamped velocity profile. (b) With the high-speed underdamped velocity profile. (c) With the trapezoidal velocity profile.

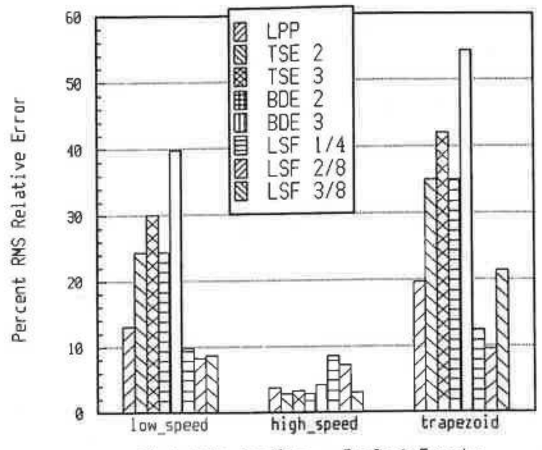


Fig. 3. The percent rms relative error for the fixed-time algorithms with the *perfect* encoder.

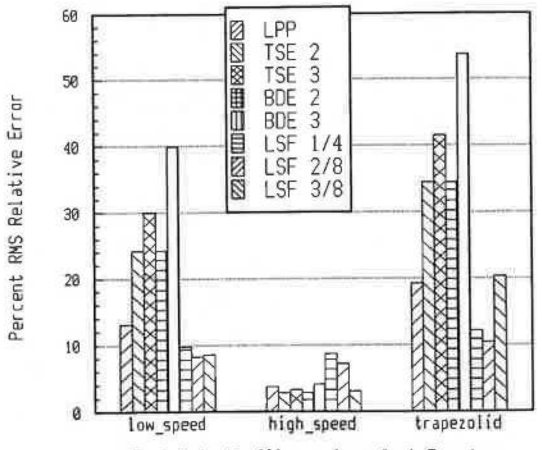


Fig. 4. The percent rms relative error for the fixed-time algorithms with an *imperfect* encoder.

tion is small. But at low speeds, the relative error can become quite large. As can be seen in Fig. 2(b), the relative error of the LPP method is small as the velocity reaches approximately 100 lines per sampling period. When the velocity is low, as in the left part of Fig. 2(a), where the velocity is 1.5 lines per sampling period, the velocity errors of many of the methods are quite large. At intermediate speeds, 10.3 lines per sampling period on the right side of Fig. 2(a) and 15.5 lines per sampling period on the left side of Fig. 2(b), some of the methods are generating significant errors. As expected, the estimations produced by the TSE and BDE methods amplified the position measurement errors of the LPP method but had better transient responses. Note that the TSE 3 is slightly better than the BDE 3. This is due to the attenuation of the third-order term of the TSE 3 compared to the BDE 3. The LSF methods produced estimations with poorer transient responses, but the errors produced at near constant speeds below 20 lines per sampling period were significantly smaller than the errors generated with the LPP, TSE, and BDE methods.

B. Analysis of Errors in Fixed-Position Velocity Estimators

The fixed-position velocity estimator algorithms were also simulated using the velocity profiles in Fig. 1. The t_k test data obtained from these profiles included the effect of the time measurement error inherent in the fixed-position methods. The t_k test data used were the ideal time truncated to six decimal places to represent time being measured with a 1-MHz clock. The relative errors produced by the estimators with the perfect encoder are shown in Fig. 5. Fig. 6 illustrates the same errors when an imperfect encoder is used. the rms relative errors for the simulations are shown in Fig. 7 and Fig. 8 for the perfect and imperfect encoders, respectively.

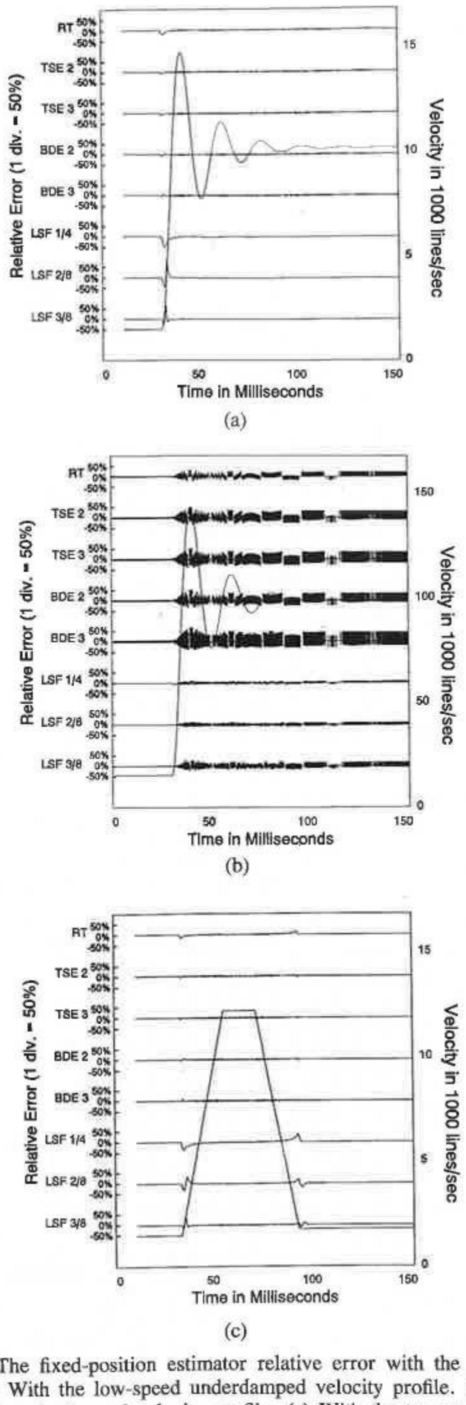
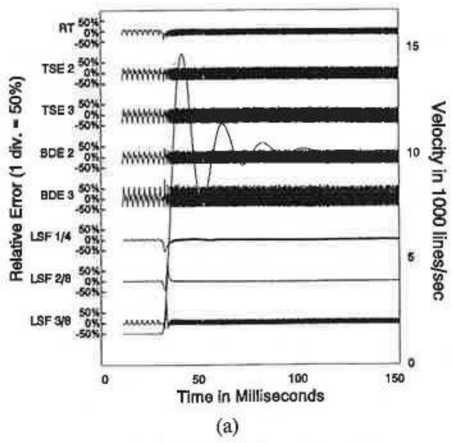
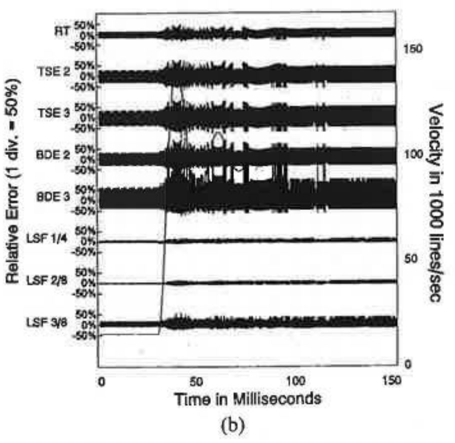


Fig. 5. The fixed-position estimator relative error with the *perfect* encoder. (a) With the low-speed underdamped velocity profile. (b) With the high-speed underdamped velocity profile. (c) With the trapezoidal velocity profile.

Perfect Encoder Simulation: For all the simulations with the perfect encoder, the fixed-position estimators do well at





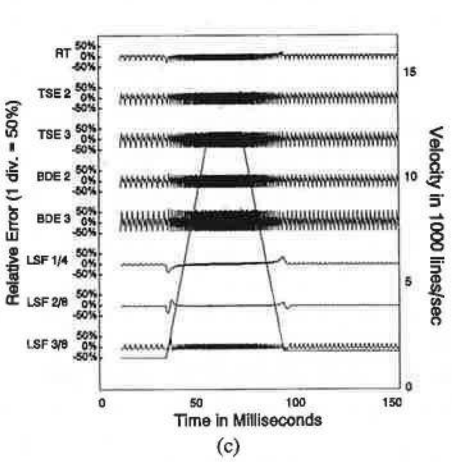


Fig. 6. The fixed-position estimator relative error with an *imperfect* encoder. (a) With the low-speed underdamped velocity profile. (b) With the high-speed underdamped velocity profile. (c) With the trapezoidal velocity profile.

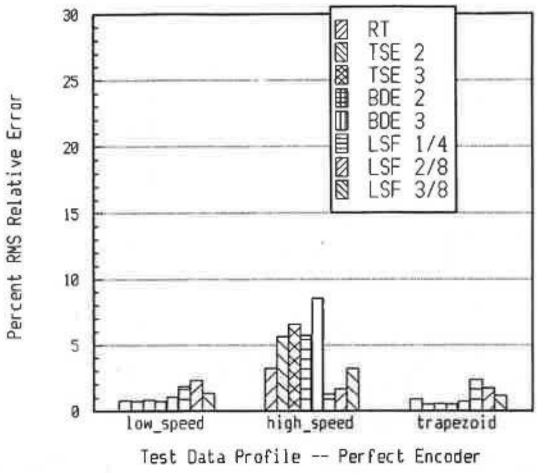


Fig. 7. The percent rms relative error for the fixed-position algorithms with the *perfect* encoder.

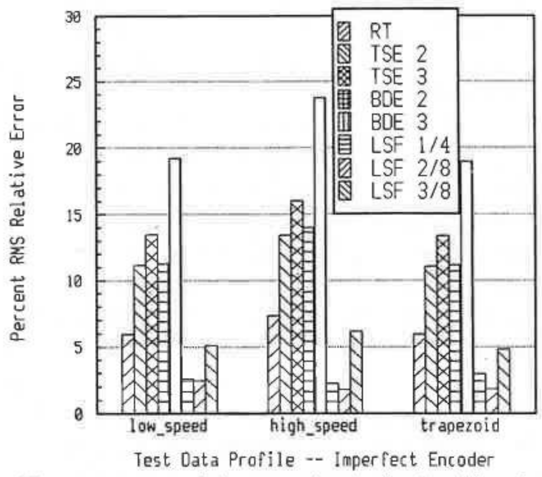


Fig. 8. The percent rms relative error for the fixed-position algorithms with an *imperfect* encoder.

the near constant lower speeds. The estimator based on finite difference approximations of dt/dx, RT, TSE 2, TSE 3, BDE 2, and BDE 3 are most accurate at the lower speeds, with no errors exceeding 5% except at the discontinuities in the acceleration. At higher speeds the errors become large due to time measurement truncation. For example, in the right hand portion of Fig. 5(b), where the time between sampling instants is approximately 9.7 μs, the measured time for any given interval is either 9 or 10 μs. Increasing the order of the estimators from 1 to 2 to 3 does increase the ability to follow the velocity during transients but at the expense of greater error at high speeds.

The LSF estimators have large errors during the velocity transients, but the effect of the time measurement truncation is less (or filtered) compared to the RT, TSE, and BDE methods. At near constant speeds, the LSF 2/8 estimator has the lowest error, whereas the LSF 3/8 estimator is more accurate at the transients.

Imperfect Encoder Simulations: The first observation to make based on a comparison of the error data presented in Figs. 5 and 6, for the perfect and imperfect encoders, respectively, is that the errors of all the fixed-position estimators are significantly larger with imperfect encoders than with perfect encoders; i.e., inherent encoder errors overshadow time measurement errors when determining total fixed-position estimator errors. However, the same *general* features noted above in the discussion of the perfect encoder simulations also appear in the imperfect encoder simulations for the fixed-position estimators.

The most striking result to be noted in the imperfect encoder simulations is that the LSF 2/8 estimator does the best job of all the fixed-position velocity estimators. Except

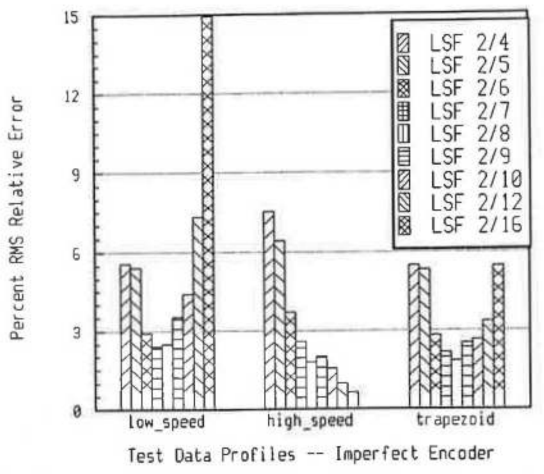


Fig. 9. Percent rms relative errors for the fixed-position quadratic LSF velocity estimators fit to 4, 5, 6, 7, 8, 9, 10, 12, and 16 points.

at the discontinuous accelerations, the relative error in the LSF 2/8 estimator never exceeds 5% and is the least affected by encoder variations.

The LSF 2/8 estimator, as well as acting as a differentiator is also functioning as a low-pass filter for the encoder error spectrum, apparently even better than the LSF 3/8 estimator. Because an overdetermined system is needed to effect the LSF fit, it might be expected that the LSF 2/4 estimator would also act as an effective differentiator/error filter. To test this hypothesis, the results of imperfect encoder simulations for the velocity profiles were compared for the quadratic LSF estimators fit to 4, 5, 6, 7, 8, 9, 10, 12, and 16 points, these results are shown in Fig. 9. The LSF 2/4 estimator represents "too exact" a fit to the data and does not provide much filtering effect. The LSF 2/8 works well because the length of the estimator is a multiple of the encoder error period. Increasing the number of data points in the estimator to 12, and then 16 enhances the filtering, particularly at (near) constant speeds. This enhanced filtering behavior is, of course, at the expense of computation speed (more points take more time to process) and larger errors and longer delays in response to a discontinuity in the acceleration (there is more "old" data in the estimator). The LSF 2/8 estimator is probably the best "all-around" estimator in this series of simulations. The relative error away from the discontinuities in acceleration is less then 3%. At the discontinuities, the error is comparable to or better than the other LSF estimators.

IV. Commentary on Algorithm Implementation

The simulations in the previous section demonstrated that no one algorithm is superior to the others in all situations. The choice of the best estimator algorithm is application dependent and should be coupled with the choice of encoder.

Fixed-time algorithms are generally natural to implement in a discrete time control system, and are best used in high-speed applications. At high speeds, any of the fixed-time algorithms are good velocity estimators at steady speeds, with the LSF algorithms providing the lowest error. If, in a velocity feedback system, the ability of the velocity to follow velocity transients is important, then one of the "exact" fit estimators, such as the BDE3 or TSE 3 may be the proper choice but only when speed is at least on the order of 100 encoder lines per sampling period. In very high speed applications, "processing" time becomes important. In such cases, the simplest algorithm, the LPP, may be the best.

When accuracy at low speeds is critical, the fixed-position LSF algorithms are the best choice. As the fixed-position algorithms are particularly sensitive to imperfections in encoders, the corresponding TSE and BDE algorithms should never be used in low-speed applications, as these "exact fit" algorithms magnify the inherent encoder errors. On the other hand, the LSF fixed-position algorithms filter the effects of the encoder imperfections and significantly extend the low-speed range, and provide good transient response. The accuracy of the fixed-position LSF algorithms begins to degrade at "medium" speed.

Other application specific factors, such as 1) the maximum acceptable error of velocity estimation, 2) the cost of the development and implementation of any particular algorithm, and 3) the actual dynamic range of speed for the application, must also be considered when making a choice of algorithm. An application that will encompass a significant range of speeds from very low to even moderately high, may need to dynamically switch between algorithm types.

V. Summary

In this paper, many velocity estimator algorithms using discrete position versus time measurements have been brought together with the theory behind the algorithms explained, and a new set of algorithms based on least squares has been proposed. Computer simulations have been performed to evaluate and compare these algorithms with simulated ideal and real data as both time- and fixed-position estimators. The simulations show that no one estimator algorithm is best for a system that has a large dynamic range of speeds, has large transients, and uses an imperfect (real) encoder. At low speeds, fixed-position estimators worked best where fixed-time estimators worked best at high speeds. The LSF estimators filtered the effect of imperfect measurements where the TSE and BDE estimators responded better to velocity transients.

The work included in this paper has been intended to serve as a unification of the work of previously presented velocity estimators as well as to specifically outline the theoretical basis of these estimators. In addition, a new group of velocity estimators, the LSF group, has been proposed, developed and evaluated. These is still room for improvement in velocity estimators. Perhaps refined algorithms can be found using narrow band differentiators with constant phase ($+90°$) or using frequency domain designs where both magnitude and phase are specified.

References

[1] L. R. Rabiner and K. Steiglitz, "The design of wide-band recursive and nonrecursive digital differentiators," *IEEE Trans. Audio Electroacoust.*, vol. AU-18, no. 2, pp. 204–209, June 1970.

[2] S. Spriet and J. Bens, "Optimal design and comparision of wide-band

digital on-line differentiators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, no. 1, pp. 46-52, Feb. 1979.

[3] T. Ohmae, T. Matsuda, K. Kamiyama, and M. Tachikawa, "A microprocessor-controlled high-accuracy wide-range speed regulator for motor drives," *IEEE Trans. Industrial Electron.*, vol. IE-29, no. 3, pp. 207-211, Aug. 1982.

[4] R. Bonert, "Digital tachometer with fast dynamic response implemented by a microprocessor," *IEEE Trans. Industry Applications*, vol. IA-19, no. 6, pp. 1052-1056, Nov. 1983.

[5] M. F. Rahman and A. N. Poo, "Optical encoder interfaces for digital speed and position control and systems considerations," in *Proc. Conf. Applied Motion Contr.*, June 1985, pp. 176-180.

[6] J. Tal, "Velocity decoding in digital control systems," in *Proc. Ninth Annual Symp. Incremental Motion Contr. Syst. Devices*, June 1980, pp. 195-203.

[7] R. H. Brown and S. C. Schneider, "Velocity observations from discrete position encoders," in *Proc. IECON'87, Thirteenth Annu. IEEE Industrial Electronics Society Conf.*, Boston, MA, Nov., 1987, pp. 1111-1118.

[8] K. Saito, K. Kamiyama, T. Ohmae, and T. Matsuda, "A microprocessor-controlled speed regulator with instantaneous speed estimation for motor drives," *IEEE Trans. Industrial Electron.*, vol. IE-35, no. 1, pp. 95-99, Feb. 1988.

[9] C. F. Gerald, *Applied Numerical Analysis*. New York: Addison-Wesley, 1970.

[10] R. W. Hamming, *Digital Filters*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[11] D. G. Taylor and B. C. Kuo, "A velocity prediction algorithm for systems with discretized positional feedback," in *Proc. Fourteenth Annual Symp. Incremental Motion Control Systems and Devices*, June 1985, pp. 191-201.

[12] William L. Brogan, *Modern Control Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1983.