Proceedings of the 42nd IEEE
Conference on Decision and Control
Maui, Hawaii USA, December 2003

Velocity Estimation Using Quantized Measurements

Stephen M. Phillips
Department of Electrical Engineering
Arizona State University
Tempe, AZ, 85287-5706, USA
stephen.phillips@asu.edu

Michael S. Branicky
Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH, 44106-7221, USA
msb@r2d2.cwru.edu

Abstract – **This paper compares algorithms for velocity estimation for feedback control using position encoder measurements. Simulations and experiments for position-velocity control of motors using both fixed-time and fixed-displacement algorithms are compared for various operating conditions. While no single approach is optimal for all situations, least-squares fit algorithms are promising in general. Mathematical models are developed to predict the performances of the algorithms. Specifically, upper bounds on absolute and relative errors for each of the algorithms under different constant velocity and constant acceleration conditions are derived. The algorithms are extensively tested in simulation and also on data obtained from an experimental setup consisting of a motor, two encoders with different resolutions, and a real-time feedback control system. The theory and simulations are confirmed by experimental results.**

1. INTRODUCTION

The performance of motion control systems can significantly be increased by using position and velocity feedback. A common, as well as inexpensive way, for deriving the velocity of such systems is to use a discrete position encoder. The encoder is a position detector that generates a pulse when it is displaced by a unit distance and is used as a speed detector either by counting the output pulse trains for a certain time interval (fixed-time or FT) or by measuring the time between a set number of output pulses (fixed-displacement or FD).

Previous research has explored via simulation several algorithms for velocity estimation using position encoders [2]-[3], and primarily concluded that using FT algorithms are better for high-speed conditions, FD algorithms are better for low-speed conditions, and that least-squares fit (LSF) algorithms are, in general, better than exact fit algorithms such as Taylor Series Expansion (TSE) and Backward Difference Expansion (BDE). Also, there were some results presented from experiments using FT [4] and FD [7] approaches to support the simulation results.

The objective of this paper is to develop mathematical models, simulations and experiments to compare algorithms for position-velocity control of motors using position encoders. We begin by extending the work from [2] and [3] through analysis. Section 2 includes the theory of velocity estimation algorithms using discrete position and time data. Section 3 summarizes simulation results using the algorithms explained in Section 2. In Section 4 we determine the best overall algorithm for each velocity condition and compare the performance to analytically derived error bounds. We confirm that the algorithms selected result in errors that are very close to the best attainable for various velocity pro-files. Finally, the algorithms are tested on data obtained from physical experiments in Section 5. The setup consists of a motor, two encoders with different resolutions, and a real-time feedback control system.

2. VELOCITY ESTIMATION ALGORITHMS

Optical discrete position encoders produce two feedback signals in quadrature (90o out of phase). When an edge of either waveform is detected, an encoder "line" or "count" is generated. After obtaining the position and time data using an incremental encoder, we can either count output pulses within a certain time interval (fixed-time or FT) or measure the time required for the encoder to generate a fixed number of edges (fixed-displacement or FD). At the kth sampling instant, we have the time, t_k, and the position traveled, x_k, available as in Fig.1. For velocity we perform the derivative based on this quantized data, a process which inherently tend to magnify errors or noise. One way to mitigate this is to create an estimator, or observer, that manipulates the most recent n position/time data points. Such an observer necessarily has some group delay which can reduce the relative stability of a feedback system [4]. Also, a higher order might improve the performance of the estimator in fitting the gathered data, but may increase measurement errors, making them less suitable for certain applications.

The simplest (first-order) approximations for velocity estimation are the LPP (lines-per-period) and the RT (reciprocal-time) estimators. In the LPP method, an FT approach, the encoder lines detected are counted during a fixed sampling period. The number of lines counted in the k^{th} time interval, Δx_k ($\Delta x_k = x_k - x_{k-1}$), and the duration of each time interval, T, provide the velocity estimate for the k^{th} interval, $\hat{v}_k = \Delta x_k / T$, lines per time unit or $\hat{v}_k = \Delta x_k$ lines per sampling period. In the RT method, a FD approach, the time between a fixed number of encoder counts is measured. $T_k = t_k - t_{k-1}$, the measured time to travel

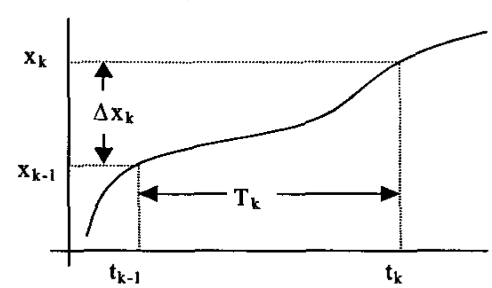

Fig. 1 Position versus time notation for velocity estimation.

this distance, provides the velocity estimate for the k^{th} interval, $\hat{v}_k = 1/T_k$, lines per time unit, or $\Delta x_k / T_k$ lines per unit time. These velocity estimates are the *average* velocities over the sampling intervals, not the true velocity at the samples. The accuracy of these two methods depends on the number of lines per period and the rate of change of the velocity.

A **Taylor Series** expansion (TSE) for the velocity about a fixed value of time t_β with the velocity v_β, is $\sum_{j=0}^{\infty} \frac{1}{j!} v_\beta^{(j)} (t_k - t_\beta)^j$ where $v_\beta^{(j)}$ is j^{th} derivative of v_β

Use this expansion with FD data where $v_\beta \approx 1/T_k$ is the average velocity of period, k, assumed to occur at the center of the period (i.e. $t_k - t_\beta \approx T_k/2$). The derivative data can similarly be approximated. Substituting these approximations into the Nth order TSE yields

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

A similar derivation can be done using FT measurements. Note that when the TSE is truncated after the first term (first-order), $\hat{v}_k = 1/T_k$ is the same as the reciprocal-time estimator and $\hat{v}_k = \Delta x_k$ is the same as the lines-per period estimator. These expressions closely follow those of the data extrapolators or "holds" used in classical digital control.

A **Backwards Difference Estimation** (BDE) approach uses the discrete data points (x_k, t_k), to determine the derivative of the function $t(x_k)$ or $x(t_k)$ by assuming that the actual function can be replaced by an interpolating polynomial that *exactly* fits the data points. The method is based on finite difference approximations. Again, the first-order BDE estimators are identical to the reciprocal-time estimator for FD and to the lines-per period for FT.

A **Least Squares Estimate** (LSE) approach avoids the interpolating functions above which can be very sensitive to measurement errors in the data. This leads us to the least squares fit method. The particular error minimized is the sum of the squares of the deviations (differences) between the estimate and the measured (actual) data points. The velocity is estimated by evaluating the derivative of the polynomial at the most recent data points. Note that the LSF estimators can be implemented as finite impulse response (FIR) digital filters of order M. The FIR filter coefficients for a line fit to two, four or eight points (LSF 1/2, 1/4, 1/8), a quadratic fit to eight points (LSF 2/8), and a cubic fit to eight points (LSF 3/8) are listed in Table 1. Note that LSF 1/2 is the same as the LPP algorithm for FT and the RT algorithm for FD.

3. SIMULATION EXPERIMENTS

The performances and sensitivities of the different estimators were tested and analyzed by simulations in Matlab for systems with position measurement truncation (FT), time measurement truncation (FD), and position-base jitter (FD) which represents encoder imperfections.

	1/2	1/4	LSF 1/8	LSF 2/8	LSF 3/8
h_1	-1.0	-0.30	-0.0833	0.2083333	-0.2777778
h_2	1.0	-0.10	-0.0595	-0.0178571	0.3293651
h_3		0.10	-0.0357	-0.1607143	0.3253968
h_4		0.30	-0.0119	-0.2202381	-0.0119048
h_5			0.0119	-0.1964286	-0.4047619
h_6			0.0357	-0.0892857	-0.5753968
h_7			0.0595	0.1011905	-0.2460317
h_8			0.0833	0.3750000	0.8611111

Table 1. FIR Coefficients for Least-Squares Estimators

Various **velocity profiles** that had wide velocity ranges and changes were used [3]:

1. High-speed: Underdamped velocity profile starts with an initial speed of 15.5 lines/sample and transitions to 103.3 lines/sample with a final time of 150 ms, a damping ratio of 0.2 and a natural frequency of 325.
2. Low-speed: Underdamped velocity profile similar to above except with initial speed 1.5 lines/sample and final speed 10.3 lines/sample.
3. Low-speed trapezoidal: Initial constant speed of 1.5 lines/sample, ramps up to a constant speed of 12.35 lines/sample and ramps down to a speed of 1.75 counts/sample.

The main FT and FD velocity estimators tested on the above three velocity profiles were the first order LPP and RT, the second and third order Taylor TSE2, TSE3 and backward difference BDE2, BDE3 and least squares LSF 1/4, 1/8, 2/8, 3/8, 2/4, 2/6, 2/7, 2/10, 2/12, 3/8, and 3/12.

To make realistic simulations, we performed "imperfect encoder" tests with period 4 and actual increments of [0.95 0.95 0.9 1.2]. Note that these add to 4, as expected.

To evaluate the results of the simulations, we used the Percent Relative Root-Mean Squared Error of the velocity estimators.

For FT: Here we have *position measurement truncation*. At high velocities (>15 counts/sample), a large number of pulses are counted per each timing period, which leads to high relative accuracy. At low velocities (0-15 counts/sample), however, there will only be a few pulses per each time period. Thus, the relative precision will be low.

For FD: In FD measurements, the time between two encoder readings, which gives a fixed displacement, is measured by counting clock cycles. This leads to *time measurement truncation.* In FD simulations, the precision is low at high velocities as the period is relatively short between pulses, but the precision is high at low velocities when the pulses are widely spaced. As noted above, in real encoders, the spacing between the edges may be unequal causing imperfections in the encoder resulting in *position-base jitter.*

The **fixed-time** velocity estimators were simulated using the three velocity profiles with both the perfect and imperfect encoders. The percent relative errors were plotted and bar graphs showing the percent rms relative errors for each of the fixed-time methods were obtained. The sampling period (T) was set to one millisecond (1 ms). The first 18 milliseconds of each profile were ignored due to required number of samples needed to have a velocity estimate.

The low-speed profile results show that the errors are quite large in general. The errors result largely from the poor transient response of most estimation algorithms. LPP, TSE, and BDE estimators have better transient responses. Additionally, LSF estimators result in less error at near-constant speeds. The LSF 2/8 estimator has the least overall error.

The high-speed profile results also show that the LPP, TSE, and BDE estimators have better transient responses and the LSF estimators result in less error at near-constant speeds. However, the overall errors are smaller than those for the low-speed profile. The reason is that position measurement truncation error effects decrease with the increasing speed. LPP, TSE 2, BDE 2 and LSF 2/8 estimators have less overall error than the other estimators.

For the low-speed trapezoidal profile results overall errors are larger than those for the previous two profiles. The reasons are more transitions in the particular profile and the low-velocity range. The LSF 2/8 performs the best among all. There is almost no difference between the plots of the perfect and imperfect encoder simulations. We also completed the case of a period-5 encoder error [0.9 0.85 1.15 0.85 1.25] and the results were similar. We conclude that the encoder errors do not significantly affect the percent RMS relative errors for FT simulations of the speed ranges considered.

Using the above simulation results, we conclude that:

- TSE and BDE estimators have better transient response.
- At near constant speeds, LSF estimators do better than the LPP/TSE/BDE ones.
- At low speeds errors of all methods are quite large, but LSF estimators perform better than the other estimators.
- TSE/BDE estimators amplify position measurement truncation errors.
- FT estimators result in less overall error in the case of high-speed velocities.
- All FT estimators are insensitive to encoder imperfections.
- For low-speed profiles, LSF 2/8 estimator performs the best. For high-speed profiles, TSE 2, BDE 2, and LSF 3/8 estimators almost equally have the least overall error.

We repeated the simulations for the **fixed-displacement** velocity estimators. The t_k data used were assumed to be measured with a perfect 1-MHz clock, which means that all time data were truncated to six decimal places to model the effect of the time measurement truncation in our simulations.

Unlike the FT case, the FD simulations result in significantly larger errors with the imperfect encoder than with the perfect encoder implementation. For the low-speed profile (perfect) case all FD estimators perform well at low speeds with the exception that the LSF estimators do not perform as well as TSE and BDE do at the transients.

The errors are larger for the high-speed profile (perfect) case. After the first high acceleration part of the profile, in which the high-speed range has been entered, the errors are significantly larger. This is the result of time measurement truncation at high speeds.

For the low-speed trapezoidal profile (perfect) it is again confirmed that all FD estimators perform well at low speeds with the exception that the LSF estimators do not perform as well as TSE and BDE do at the transients. The speed is always in the low-speed range (1-15 counts/sample) for this particular profile, but there is larger overall error than for the low-speed profile due to the extra transition.

For the imperfect encoder simulations, we also have the position-base jitter error source besides the time measurement truncation error and all estimators produce much larger errors. The errors for the high-speed profile (imperfect) case are the largest seen so far. The large increase is due to the contributions of the encoder imperfections producing larger errors than the errors caused by the time measurement truncation. LSF estimators are still more promising than the others. For the low-speed trapezoidal profile (imperfect) the higher errors are again due to the number of transitions and the speed-range of the velocity profile.

For all of the **perfect-encoder** simulations, time measurement truncation is the main source of error.

- All FD estimators perform well at constant lower speeds and the errors become larger at higher speeds due to time measurement truncation.
- The RT/TSE/BDE estimators perform better overall than the LSF estimators on the low-speed profiles (due to the poor performance of LSF estimators during the velocity transients).
- The LSF estimators do a better job than RT/TSE/BDE estimators on the high-speed velocity profile because they filter the effect of the time measurement truncation better.
- As for the best performers, TSE 2 is the best for low-speed and low-speed trapezoidal velocity profiles while LSF 1/8 is the best for the high-speed profile.
- LSF 1/4 and 2/8 are the best overall.

For the **imperfect-encoder** simulations, we add the position-base jitter error to the time measurement truncation.

- All imperfect-FD estimators produce much larger errors at higher speeds. This is because the inherent encoder imperfections produce larger errors than the errors caused by the time measurement truncation.
- For all three of the velocity profiles, the LSF estimators are doing a much better job, in general, than the RT/TSE/BDE estimators. They filter the effects of the imperfect measurements better.
- The LSF 2/8 estimator is the best for low-speed and low-speed trapezoidal velocity profiles and the best overall. LSF 1/8 is the best for the high-speed velocity profile and the second best overall.

After characterizing the general observations on the simulations using both perfect and imperfect encoders, we examined how different FD LSF estimators performed on the same profiles. In perfect FD simulations, LSF 2/4 is doing the best job for the low-speed and low-speed trapezoidal velocity profiles. LSF 1/8 estimator is doing the best job for the high-speed velocity profile. If we consider all three velocity profiles, the LSF 2/6, LSF 2/7, and LSF 2/8 estimators are better. For these, the errors are less than 3%.

In imperfect FD simulations (not shown), LSF 2/8 does the best job for the low-speed and low-speed trapezoidal velocity profiles. LSF 1/8 and LSF 2/12 are almost equal for the high-speed velocity profile. Considering all three velocity profiles, LSF 2/8 performs the best.

In comparing the FT and FD estimators the criteria for an algorithm being "better" in a particular case was to be the best or close to the best in that category. The criteria for being "overall better" was that it had the least (or close to the least) overall summed errors over all profiles. It was found that there was no particular estimator that is best for every application.

We now point out the common and distinct properties for both FT and FD.

- The LPP/RT/TSE/BDE estimators (based on finite difference approximations) perform better than the LSF estimators during velocity transients.
- The LSF estimators are more promising in general. LSF 2/8 estimator has a better overall performance for low-to-mid range velocities for both FT and FD. LSF 1/8 has the best overall performance among FD estimators for the high-speed profile and LSF 2/4 estimator has a better overall performance among FT estimators for the high-speed profile.
- Generally, FD estimators are better for low-speed range velocity profiles and FT estimators are better for high-speed range velocity profiles.
- FT estimators are insensitive to encoder imperfections. Conversely, even slight encoder error result in large errors for the FD estimators and this error depends critically on the size and distribution of the imperfections.

The above conclusions showed us that no single estimation method is best for a system that has a large dynamic range of speeds, has large transients, and embodies an imperfect (real) encoder (though LSF 2/8 is a fairly versatile performer). Hence, the method to use may be application-dependent. For instance, the method chosen may depend on the importance that a velocity feedback system follow velocity transients, the maximum allowable error of velocity estimation, the cost of the development and implementation of any particular method, or the actual dynamic range of speed for the application.

4. ANALYTICAL APPROACH

Although simulations are useful in exploring the different velocity estimation methods, we explored analytical tools for evaluating the methods' performance. Specifically, we used floor operations [6] to derive worst-case error bounds for many of the methods in cases of constant velocity and constant acceleration profiles. The results are applicable in determining which algorithms perform best under what conditions.

The floor of a real x is defined as $\lfloor x \rfloor$ = the greatest integer less than or equal to x. Also, $\{x\} = x - \lfloor x \rfloor$ is the fractional part of x. Note that $0 \le \{x\} < 1$. One can easily derive fundamental equations for dealing with floors [5]. Using fundamental formulas for positive x and y, we find $\lfloor x+(-y) \rfloor = \lfloor x \rfloor + \lfloor -y \rfloor$ or $\lfloor x+(-y) \rfloor = \lfloor x \rfloor + \lfloor -y \rfloor + 1$, and the formulas for $\lfloor -y \rfloor$ gives us three possible equalities for $\lfloor x+(-y) \rfloor$. These are $\lfloor x \rfloor - \lfloor y \rfloor - 1$, $\lfloor x \rfloor - \lfloor y \rfloor$, $\lfloor x \rfloor - \lfloor y \rfloor + 1$. Although everything seems to be reasonable, the third choice turns out to be impossible. Hence we find:

$$\lfloor x-y \rfloor = \lfloor x \rfloor - \lfloor y \rfloor \text{ or } \lfloor x-y \rfloor = \lfloor x \rfloor - \lfloor y \rfloor - 1$$

where x and y are *positive* real numbers.

In the following, this equation will be used extensively. In most cases, it will be used multiple times in one equation or derivation. We use $\text{comb}_m\{a_1, a_2, \ldots, a_n\}$, where the a_i are real numbers, to denote the set of real numbers that can be found by adding up to m of the a_i. Thus, $\text{comb}_2\{0,1/2,1\}=\{0,1/2,1,3/2\}$. We also use the notation set{A} to mean that any element from A may be used in the computation at hand. E.g., for a real r, $r\,\text{set}\{a_1, a_2, \ldots, a_n\} = \text{set}\{ra_1, ra_2, \ldots, ra_n\}$.

With the above tools in hand, for **constant velocity**, v_0, the position, $x(t)=x_0+v_0t$, and at $t=kT$, $x(kT)=x_0+kv_0T$.

For the LPP Estimator we use the formula $\hat{v}_k = \frac{\Delta x_k}{T}$ so

$\hat{v}_k = \frac{\lfloor x_0 + kv_0T \rfloor - \lfloor x_0 + (k-1)v_0T \rfloor}{T}$, which is either

$$\frac{\lfloor x_0 + kv_0\mathrm{T} \rfloor - (\lfloor x_0 + kv_0\mathrm{T} \rfloor - \lfloor v_0\mathrm{T} \rfloor)}{\mathrm{T}} \quad \text{or}$$

$$\frac{\lfloor x_0 + kv_0\mathrm{T} \rfloor - (\lfloor x_0 + kv_0\mathrm{T} \rfloor - \lfloor v_0\mathrm{T} \rfloor - 1)}{\mathrm{T}}$$

These are respectively equal to $\frac{\lfloor v_0\mathrm{T} \rfloor}{\mathrm{T}}$ and $\frac{\lfloor v_0\mathrm{T} \rfloor + 1}{\mathrm{T}}$. For FT, we have T = 1 sample, which leads us to $\hat{v}_k = \lfloor v_0 \rfloor$ or $\hat{v}_k = \lfloor v_0 \rfloor + 1$. We can further reduce the estimate velocity to $\hat{v}(k) = v_0 - \{v_0\}$ or $\hat{v}(k) = v_0 - \{v_0\} + 1$. This gives the relative error for one sample as $\frac{-\{v_0\}}{v_0}$ or $\frac{-\{v_0\}+1}{v_0}$ and the upper bound for percent RMS relative error (percent RMSRE) as $100 * \sqrt{\frac{1}{N} N \frac{\max[\{v_0\}^2, (1-\{v_0\})^2]}{v_0^2}}$, which simplifies to $100 * \max[\{v_0\}, 1-\{v_0\}]/v_0$

The **second order** velocity estimate formula for TSE 2 is,

$$\hat{v}_k = \frac{3}{2}x_k - 2x_{k-1} + \frac{1}{2}x_{k-2}$$

Under constant velocity, this simplifies and becomes equal to

$$\hat{v}_k = \frac{3}{2}\lfloor x_0 + v_0k \rfloor - 2\lfloor x_0 + v_0(k-1) \rfloor + \frac{1}{2}\lfloor x_0 + v_0(k-2) \rfloor$$

Using our main formula for floor operations, we now obtain

$$\hat{v}_k = \frac{3}{2}\left(\lfloor x_0 + v_0 k \rfloor - \lfloor x_0 + v_0 k \rfloor + \lfloor v_0 \rfloor + \text{set}\{0,1\} \right)$$

$$-\frac{1}{2}\left(\lfloor x_0 + v_0 k - v_0 \rfloor - \lfloor x_0 + v_0 k - v_0 \rfloor + \lfloor v_0 \rfloor + \text{set}\{0,1\} \right)$$

$$\hat{v}_k = \lfloor v_0 \rfloor + \text{comb}_2\{-1/2, 0, 3/2\} = \lfloor v_0 \rfloor + \text{set}\left\{ -\frac{1}{2}, 0, 1, \frac{3}{2} \right\}$$

Notice that for any k value, the estimate velocity only depends on v_0. This means the number of discrete data elements N for a particular v_0 will be canceled out in the calculation for the maximum Percent RMS Relative Error, which becomes

$$100 * \max\left[\text{abs}\left(\lfloor v_0 \rfloor + \text{set}\left\{ -\frac{1}{2}, 0, 1\frac{3}{2} \right\} - v_0 \right) \right] / v_0$$

Substituting $\lfloor v_0 \rfloor = v_0 - \{v_0\}$, we have

$$100 * \max\left[\text{abs}\left(\text{set}\left\{ -\frac{1}{2}, 0, 1\frac{3}{2} \right\} - \{v_0\} \right) \right] / v_0$$

As the elements of the set are written in ascending order, the element of the set that will generate the maximum percent RMS relative error will be either the first or the last one no matter what the constant velocity, v_0. Hence, we get

$$100 * \max\left[\text{abs}\left(\text{set}\left\{ -\frac{1}{2}, \frac{3}{2} \right\} - \{v_0\} \right) \right] / v_0$$

Since $0 \le \{v_0\} < 1$, this reduces to,

$$100 * \max\left[\{v_0\} + \frac{1}{2}, \frac{3}{2} - \{v_0\} \right] / v_0$$

For **higher-order** TSE/BDE estimators, we proceed as above using $\text{comb}_n\{\}$ (n>=3). The details of these derivations can be found in [9].

For the **LSF 1/4** estimator we have the velocity estimate

$$\hat{v}_k = h_4 x_k + h_3 x_{k-1} + h_2 x_{k-2} + h_1 x_{k-3}$$

which can be simplified as above to

$$\hat{v}_k = 0.3\left(\lfloor x_0 + v_0 k \rfloor - \lfloor x_0 + v_0 k \rfloor + \lfloor 3v_0 \rfloor + \text{set}\{0,1\} \right)$$
$$+ 0.1\left(\lfloor x_0 + v_0 k \rfloor - \lfloor x_0 + v_0 k \rfloor + \lfloor v_0 \rfloor + \text{set}\{0,1\} \right)$$
$$= 0.3\lfloor 3v_0 \rfloor + 0.1\lfloor v_0 \rfloor + \text{comb}_2\{0, 0.1, 0.3\}$$
$$= 0.3\lfloor 3v_0 \rfloor + 0.1\lfloor v_0 \rfloor + \text{set}\{0, 0.1, 0.3, 0.4\}$$

Hence, we find the upper bound on percent RMS relative error for LSF 1/4 as

$$100 * \max[\text{abs}(0.3\lfloor 3v_0 \rfloor + 0.1\lfloor v_0 \rfloor + \text{set}\{0, 0.4\} - v_0)] / v_0$$

Substituting $\lfloor v_0 \rfloor = v_0 - \{v_0\}$ and $\lfloor 3v_0 \rfloor = 3v_0 - \{3v_0\}$,

$$\text{UB} = 100 * \max[\text{abs}(\text{set}\{0, 0.4\} - 0.1\{v_0\} - 0.3\{3v_0\})] / v_0$$
$$= 100 * \max[0.3\{3v_0\} + 0.1\{v_0\}, 0.4 - 0.3\{3v_0\} - 0.1\{v_0\}] / v_0$$

This was the simplest case for the LSF estimators because the LSF 1/M coefficients are symmetrical and hence easy to group together. For higher-order LSF estimators with different order of fit to various number of last data points, the derivations become rather complex. The results are tabulated in [9].

In [9] it is also verified that the actual errors are always under the derived upper bound errors. Taking both the actual and the upper bound error plots into consideration, we can find the estimator best-suited for FT under constant velocity conditions.

The LSF 1/8 estimator guarantees the best (least) upper bounds of percent RMS relative errors across all velocities shown. The LSF 1/4 estimator also has good worst-case performance. LSF 1/M estimators do well for FT under constant velocity conditions because the position data being fit is a (quantized) linear function. We should point out that the running time (the length of the simulation experiments) at the certain constant velocity does not change the results.

We also analyzed the **constant acceleration** case where the velocity is $v(t)=v_0+a_0t$, and the discrete position formula is $x(kT)=x_0+kv_0T+a_0\frac{k^2T^2}{2}$.

For the LPP estimator we follow an approach similar to before and find that UB=

$$= 100\sqrt{\frac{1}{N}\sum_{k=2}^{N}\frac{\max\left[\left(-\{v_0 + a_0 k - a_0/2\} + \text{set}\{-a_0/2, 1 - a_0/2\}\right)^2\right]}{v_k}}$$

This formula gives us an analytical way of determining the maximum allowable Percent RMS Relative Error for LPP estimator at constant acceleration with some initial velocity.

For the **BDE 2/TSE 2** estimators we find,

$$\hat{v}_k = \Delta x_k + \frac{1}{2}\left(\Delta x_k - \Delta x_{k-1} \right) = \frac{3}{2}x_k - 2x_{k-1} + \frac{1}{2}x_{k-2}$$

Notice that for an a_0 value, the estimate velocity does not only depend on v_0, it also depends on how many data samples were used. This implies that the duration of the velocity profile is also important. It is important because it gains more and more speed as the time passes. Thus, the upper bound for Percent RMS Relative Errors for constant acceleration profiles with initial v_0 is

$$100\sqrt{\frac{1}{N}\sum_{k=2}^{N}\frac{\max\left[\left(2\lfloor v_0 + a_0(k-1/2)\rfloor - \frac{1}{2}\lfloor 2v_0 + a_0(2k-2)\rfloor + \text{set}\{-1/2, 1\} - v_k\right)^2\right]}{v_k}}$$

with the element of the set that generates the maximum error at every k^{th} sample. For higher-order TSE/BDE estimators, we proceed similarly. The details of these derivations can be found in [9].

For the **LSF 1/4** estimator we find, after some tedious derivations that $\text{UB} = 100 * \sqrt{\frac{1}{N}\sum_{k=2}^{N}\frac{\max\left[\left(\hat{v}_{1/4} - v_k\right)^2\right]}{v_k}}$.

As the result of our derivations, we found that the LSF 1/8 estimator is the best for FT under constant velocity and FT under low constant acceleration conditions. Additionally, LSF 2/8 estimator performs best for FT under high constant acceleration conditions. Hence, we decided to check on the velocity estimation performance by using only the LSF 1/8, LSF 2/8, and TSE 2 estimators on the velocity profiles we explored in Section 3. Remember that LSF estimators are not good at velocity transients, so we included TSE 2 to add that capability. The performance was close to the "best of all" performance in all cases.

5. PHYSICAL EXPERIMENTS

The experimental setup consists of two encoders, motor with a resolver on board and a Sun Workstation – VME cage system. One encoder generates 10,000 counts per revolution and the other, 360,000 counts per revolution. The VME cage contains several CPU, memory and DAC boards. The internal clock of each CPU is incremented with 100 microsecond resolution. The control loop uses a sampling time of 0.001 sec.

To use the velocity estimation algorithms, position data were taken using both of the encoders on the shaft. The data for the 360K encoder were to be used to obtain actual velocity values and the 10K encoder data for the velocity estimate.

FT and FD algorithms were tested on **constant velocity** profiles. For FT, the position (encoder count) was read whenever a sampling time (T=1ms) passed until a certain number of data points were reached or the whole velocity profile was captured. The accuracy of FT algorithms solely depends on the resolution of the position encoder and the sampling period T. This resolution is not enough at low speeds, making the FT algorithms inadequate for low speed applications [8].

For FD, the velocity estimate was updated at each time a fixed-displacement occurred until a certain number of data points were reached or the whole velocity profile was captured. The accuracy of FD algorithms depends on the counter bit-width and the speed of the motor. High sensor resolution combined with large motor speed causes the time interval between successive fixed encoder ticks to be small. This leads to the sensitivity of the timer resolution, which may contribute to large errors for high-speed estimates [8].

As expected, the errors decrease as the constant velocity value increases. The LSF 1/8 estimator produces the least errors. LSF 1/4 performance is close to it. This is to be expected from our simulation experiments and the upper bound analysis that LSF 1/M estimators perform well at constant velocities.

Comparing the estimates obtained from the 10K-encoder to the actual data obtained from 360K-encoder we find that the LSF 1/8 estimator produces the least errors as expected from our simulation experiments and upper bound analysis. The FD algorithms performed similarly consistent with the simulations as shown in Figure 5.7.

6. CONCLUSIONS

This paper develops simulation software, an analytical approach and an experimental setup for velocity estimation with quantized position measurements. While no algorithm is best-suited for all velocity ranges and changes, using FT algorithms for high-speed, FD for low-speed, and LSF algorithms for general situations proved advantageous. We explored both FT and FD LSF algorithms in some depth. We also used combinations of algorithms that resulted in much lower error than using a single algorithm under different velocity ranges and changes. We compared our results with the 'best possible' case.

Additionally, we investigated the FT and FD methods under constant velocity and constant acceleration conditions with analytical tools, specifically using floor operations. We derived upper bound errors for the algorithms under different velocity conditions. Our analytical results helped us to decide the appropriate algorithms to use under given velocity conditions. We confirmed that the algorithms so-selected resulted in errors that were very close to the best attainable for various velocity profiles. The algorithms were tested on data obtained from a physical experimental setup.

All algorithms investigated in this paper are non-model based algorithms. They do not employ a model for the system to estimate velocity. Other algorithms can be developed using models of the system, and better results may be achieved for velocity estimation.

7. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the National Science Foundation and NASA.

8. REFERENCES

[1] D.M. Auslander and C.J. Kempf, *Mechatronics: Mechanical System Interfacing*, Chapter 10, Upper Saddle River, NJ: Prentice-Hall, 1996.

[2] R.H. Brown and S.C. Schneider, "Velocity Observations from Discrete Position Encoders," Proc. IECON'87, vol. 6, pp. 1111-1118, Cambridge, MA, Nov. 1987.

[3] R.H. Brown, S.C. Schneider, and M.G. Mulligan, "Analysis of Algorithms for Velocity Estimation from Discrete Position versus Time Data," IEEE Transactions on Industrial Electronics, vol. 39, no. 1, pp. 11-19, Feb. 1992.

[4] P.S. Carpenter, R.H. Brown, et al., "On Algorithms for Velocity Estimation Using Discrete Position Encoders", Proc. IECON'95, pp. 844-849, Orlando, FL, Nov. 1995.

[5] D.L. Chow, "Precise Tracking Control for CAM-LEM", Technical Report TR 97-108, CAISR, Case Western Reserve University, Cleveland, OH, Aug. 1997.

[6] R.L. Graham, D.E. Knuth, O. Patashnik, *Concrete Mathematics*, Chap.3, Reading, MA: Addison-Wesley, 1994.

[7] M. Lemkin, P.H.Yang, A.C. Huang, J. Jones, and D.M. Auslander, "Velocity Estimation from Widely Spaced Encoder Pulses," American Control Conference, vol. 1, pp. 998-1002, Seattle, WA, June 1995.

[8] Texas Instruments Digital Signal Processing Group, "Using the Capture Units for Low Speed Velocity Estimation on a TMS320C240," App. Brief SPRA363, July 1997.

[9] B. Bayraktar, Velocity Estimation Using Quantized Measurements: Theory and Experiments, M.S. Thesis, Case Western Reserve University, 1999.